

# SIEMENS

## SIMATIC

### S7 Automate programmable S7-1200


#### Manuel système


Avant-propos	
Présentation du produit	1
Logiciel de programmation STEP 7	2
Installation	3
Concepts concernant les automates programmables	4
Configuration des appareils	5
Concepts de programmation	6
Instructions de base	7
Instructions avancées	8
Instructions technologiques	9
Communication	10
Serveur Web	11
Processeur de communication	12
Communication TeleService (courrier électronique SMTP)	13
Outils en ligne et de diagnostic	14
Caractéristiques techniques	A
Calcul d'un bilan de consommation	B
Numéros de référence	C


## Mentions légales

### Signalétique d'avertissement

Ce manuel donne des consignes que vous devez respecter pour votre propre sécurité et pour éviter des dommages matériels. Les avertissements servant à votre sécurité personnelle sont accompagnés d'un triangle de danger, les avertissements concernant uniquement des dommages matériels sont dépourvus de ce triangle. Les avertissements sont représentés ci-après par ordre décroissant de niveau de risque.

 <b>DANGER</b>
signifie que la non-application des mesures de sécurité appropriées <b>entraîne</b> la mort ou des blessures graves.

 <b>ATTENTION</b>
signifie que la non-application des mesures de sécurité appropriées <b>peut entraîner</b> la mort ou des blessures graves.

 <b>PRUDENCE</b>
accompagné d'un triangle de danger, signifie que la non-application des mesures de sécurité appropriées peut entraîner des blessures légères.

<b>PRUDENCE</b>
non accompagné d'un triangle de danger, signifie que la non-application des mesures de sécurité appropriées peut entraîner un dommage matériel.

<b>IMPORTANT</b>
signifie que le non-respect de l'avertissement correspondant peut entraîner l'apparition d'un événement ou d'un état indésirable.


En présence de plusieurs niveaux de risque, c'est toujours l'avertissement correspondant au niveau le plus élevé qui est reproduit. Si un avertissement avec triangle de danger prévient des risques de dommages corporels, le même avertissement peut aussi contenir un avis de mise en garde contre des dommages matériels.

### Personnes qualifiées

L'appareil/le système décrit dans cette documentation ne doit être manipulé que par du **personnel qualifié** pour chaque tâche spécifique. La documentation relative à cette tâche doit être observée, en particulier les consignes de sécurité et avertissements. Les personnes qualifiées sont, en raison de leur formation et de leur expérience, en mesure de reconnaître les risques liés au maniement de ce produit / système et de les éviter.

### Utilisation des produits Siemens conforme à leur destination

Tenez compte des points suivants:

 <b>ATTENTION</b>
Les produits Siemens ne doivent être utilisés que pour les cas d'application prévus dans le catalogue et dans la documentation technique correspondante. S'ils sont utilisés en liaison avec des produits et composants d'autres marques, ceux-ci doivent être recommandés ou agréés par Siemens. Le fonctionnement correct et sûr des produits suppose un transport, un entreposage, une mise en place, un montage, une mise en service, une utilisation et une maintenance dans les règles de l'art. Il faut respecter les conditions d'environnement admissibles ainsi que les indications dans les documentations afférentes.

### Marques de fabrique

Toutes les désignations repérées par ® sont des marques déposées de Siemens AG. Les autres désignations dans ce document peuvent être des marques dont l'utilisation par des tiers à leurs propres fins peut enfreindre les droits de leurs propriétaires respectifs.

### Exclusion de responsabilité

Nous avons vérifié la conformité du contenu du présent document avec le matériel et le logiciel qui y sont décrits. Ne pouvant toutefois exclure toute divergence, nous ne pouvons pas nous porter garants de la conformité intégrale. Si l'usage de ce manuel devait révéler des erreurs, nous en tiendrons compte et apporterons les corrections nécessaires dès la prochaine édition.

# Avant-propos

## Objet du manuel

La famille S7-1200 est constituée d'automates programmables (API) utilisables dans des applications d'automatisation variées. Sa forme compacte, son faible prix et son important jeu d'instructions en font une solution idéale pour la commande d'applications très variées. En outre, les modèles S7-1200 ainsi que l'outil de programmation se basant sur Windows vous offrent la souplesse nécessaire pour résoudre vos problèmes d'automatisation.

Ce manuel qui fournit des informations sur l'installation et la programmation des automates S7-1200 est conçu pour des ingénieurs, des programmeurs, des installateurs et des électriciens ayant une connaissance générale des automates programmables.

## Connaissances fondamentales requises

Pour comprendre ce manuel, vous devez avoir des connaissances générales sur l'automatisation et les automates programmables.

## Domaine de validité

Ce manuel décrit les produits suivants :

- STEP 7 V11 Basic et Professional
- Firmware de la CPU S7-1200 version V3.0

Vous trouverez la liste complète des produits S7-1200 décrits dans ce manuel dans les caractéristiques techniques (Page 747).

## Homologation, marquage CE, C-Tick et autres normes

Reportez-vous aux caractéristiques techniques (Page 747) pour plus d'informations à ce sujet.

## Service et assistance

En complément de notre documentation, nous proposons notre expertise technique sur Internet via le site Web du service client (<http://www.siemens.com/automation/>).

Adressez-vous à votre agence Siemens si certaines de vos questions techniques restent sans réponse, si vous voulez connaître les offres de formation ou si vous désirez commander des produits S7. Comme ce personnel est techniquement formé et a des connaissances très pointues sur vos activités, vos processus et vos industries, ainsi que sur les différents produits Siemens que vous utilisez, il peut apporter les réponses les plus rapides et les plus efficaces possibles à tout problème que vous pourriez rencontrer.

## Documentation et information

Le S7-1200 et STEP 7 proposent une grande variété de documents et autres ressources contenant les informations techniques dont vous avez besoin.

- Le manuel système du S7-1200 fournit des informations spécifiques sur le fonctionnement, la programmation et les caractéristiques de toute la gamme du S7-1200. En plus du manuel système, le livret Easy Book du S7-1200 donne une vue d'ensemble plus générale des possibilités offertes par la gamme S7-1200.

Le manuel système et l'Easy Book sont disponibles en formats électronique (PDF) et manuscrit. Les manuels électroniques peuvent être téléchargés depuis le site web du service client et se trouvent également sur le disque qui accompagne chaque CPU S7-1200.

- Le système d'information en ligne du STEP 7 offre un accès immédiat aux informations conceptuelles et aux instructions spécifiques qui décrivent le fonctionnement et les fonctionnalités du progiciel de programmation et le fonctionnement basique des CPU SIMATIC.
- My Documentation Manager permet d'accéder aux versions électroniques (PDF) de l'ensemble de la documentation SIMATIC, y compris le manuel système, l'Easy Book et le système d'informations du STEP 7. Avec My Documentation Manager, vous pouvez récupérer des sujets depuis divers documents pour créer votre propre manuel client.

Le portail d'entrée du service client (<http://support.automation.siemens.com>) affiche un lien vers My Documentation Manager dans mySupport.

- Le site Web du service client fournit également des podcasts, FAQ et autres documents utiles concernant le S7-1200 et STEP 7. Les podcasts utilisent de courtes vidéos éducatives qui se concentrent sur les caractéristiques ou scénarios spécifiques afin de démontrer les interactions, la commodité et l'efficacité offertes par STEP 7. Visitez les sites Web suivants pour accéder à la collection de podcasts :
  - Page Web du STEP 7 Basic (<http://www.automation.siemens.com/mcms/simatic-controller-software/en/step7/step7-basic/Pages/Default.aspx>)
  - Page Web du STEP 7 Professional (<http://www.automation.siemens.com/mcms/simatic-controller-software/en/step7/step7-professional/Pages/Default.aspx>)
- Vous pouvez également suivre ou rejoindre des discussions sur les produits sur le forum technique Service & Support (<https://www.automation.siemens.com/WW/forum/guests/Conferences.aspx?Language=en&siteid=csius&treeLang=en&groupid=4000002&extranet=standard&viewreg=WW&nodeid0=34612486>). Ces forums vous permettent de dialoguer avec différents experts produits.
  - Forum pour le S7-1200 (<https://www.automation.siemens.com/WW/forum/guests/Conference.aspx?SortField=LastPostDate&SortOrder=Descending&ForumID=258&Language=en&onlyInternet=False>)
  - Forum pour STEP 7 Basic (<https://www.automation.siemens.com/WW/forum/guests/Conference.aspx?SortField=LastPostDate&SortOrder=Descending&ForumID=265&Language=en&onlyInternet=False>)

# Sommaire

	<b>Avant-propos .....</b>	<b>3</b>
<b>1</b>	<b>Présentation du produit.....</b>	<b>19</b>
1.1	Introduction à l'automate S7-1200 .....	19
1.2	Possibilités d'extension de la CPU .....	22
1.3	Modules S7-1200.....	25
1.4	Nouvelles fonctions.....	26
1.5	Tableaux IHM basiques .....	27
<b>2</b>	<b>Logiciel de programmation STEP 7 .....</b>	<b>29</b>
2.1	Besoins du système.....	30
2.2	Différentes vues pour faciliter votre travail.....	31
2.3	Simplicité des outils .....	32
2.3.1	Insertion d'instructions dans votre programme utilisateur .....	32
2.3.2	Accès aux instructions depuis la barre d'outils "Favoris".....	32
2.3.3	Création d'une équation complexe avec une instruction simple.....	33
2.3.4	Ajout d'entrées ou de sorties à une instruction CONT ou LOG.....	35
2.3.5	Instructions extensibles.....	35
2.3.6	Sélection d'une version d'une instruction.....	36
2.3.7	Modification de l'apparence et de la configuration de STEP 7 .....	37
2.3.8	Glisser-déplacer entre les éditeurs .....	37
2.3.9	Changement de l'état de fonctionnement de la CPU.....	38
2.3.10	Modification du type d'appel pour un DB .....	39
2.3.11	Déconnexion temporaire d'appareils d'un réseau.....	40
2.3.12	Désenfichage virtuel d'appareils de la configuration .....	41
<b>3</b>	<b>Installation .....</b>	<b>43</b>
3.1	Conseils pour l'installation d'appareils S7-1200 .....	43
3.2	Bilan de consommation.....	45
3.3	Procédures d'installation et de désinstallation.....	47
3.3.1	Dimensions de montage pour les appareils S7-1200.....	47
3.3.2	Installation et désinstallation de la CPU .....	50
3.3.3	Installation et désinstallation d'un SB, d'un CB ou d'un BB.....	53
3.3.4	Installation et désinstallation d'un SM.....	55
3.3.5	Installation et désinstallation d'un CM ou d'un CP.....	57
3.3.6	Démontage et remontage du bornier de connexion S7-1200.....	58
3.3.7	Installation et désinstallation du câble d'extension .....	59
3.3.8	Adaptateur TS Adapter (TeleService).....	61
3.3.8.1	Connexion de l'adaptateur TeleService .....	61
3.3.8.2	Installation de la carte SIM.....	62
3.3.8.3	Installation de l'ensemble adaptateur TeleService .....	64
3.3.8.4	Montage mural de l'adaptateur TS.....	64

3.4	Conseils de câblage .....	65
<b>4</b>	<b>Concepts concernant les automates programmables .....</b>	<b>71</b>
4.1	Exécution du programme utilisateur .....	71
4.1.1	Etats de fonctionnement de la CPU .....	73
4.1.2	Traitement du cycle à l'état MARCHE .....	77
4.1.3	Blocs d'organisation (OB) .....	78
4.1.4	Priorités d'exécution et mise en file d'attente des événements .....	79
4.1.5	Surveillance du temps de cycle .....	85
4.1.6	Mémoire de la CPU .....	88
4.1.6.1	Mémento système et mémento de cadence .....	90
4.1.7	Mémoire tampon de diagnostic .....	92
4.1.8	Horloge temps réel .....	93
4.1.9	Configuration des sorties lors d'un passage de MARCHE à ARRET .....	93
4.2	Stockage des données, zones de mémoire, E/S et adressage .....	94
4.2.1	Accès aux données du S7-1200 .....	94
4.3	Traitement des valeurs analogiques .....	99
4.4	Types de données .....	100
4.4.1	Types de données Bool, Byte, Word et DWord .....	101
4.4.2	Types de données entiers .....	102
4.4.3	Types de données réels à virgule flottante .....	103
4.4.4	Types de données "date et heure" .....	103
4.4.5	Types de données "caractère et chaîne de caractères" .....	105
4.4.6	Type de données "tableau" .....	107
4.4.7	Type de données "structure de données" .....	108
4.4.8	Type de données API .....	108
4.4.9	Types de données "pointeur" .....	108
4.4.9.1	Type de données Pointer .....	109
4.4.9.2	Type de données Any .....	110
4.4.9.3	Type de données Variant .....	111
4.4.10	Accès à une "tranche" d'un type de données de variable .....	112
4.4.11	Accès à une variable par un type de données ajouté AT .....	113
4.5	Utilisation d'une carte mémoire .....	115
4.5.1	Insertion d'une carte mémoire dans la CPU .....	116
4.5.2	Configuration des paramètres de mise en route de la CPU avant copie du projet dans la carte mémoire .....	118
4.5.3	Utilisation de la carte mémoire en tant que carte "transfert" .....	118
4.5.4	Utilisation de la carte mémoire en tant que carte "programme" .....	120
4.5.5	Mise à jour du firmware .....	123
4.6	Récupération en cas d'oubli du mot de passe .....	126
<b>5</b>	<b>Configuration des appareils .....</b>	<b>127</b>
5.1	Insertion d'une CPU .....	128
5.2	Détection de la configuration d'une CPU non spécifiée .....	129
5.3	Ajout de modules à la configuration .....	130
5.4	Configuration du fonctionnement de la CPU .....	131
5.5	Configuration des paramètres des modules .....	133

5.6	Configuration de la CPU pour la communication.....	135
5.6.1	Création d'une liaison réseau .....	135
5.6.2	Configuration du routage local/partenaire.....	136
5.6.3	Paramètres pour la liaison PROFINET .....	138
5.6.4	Affectation d'adresses IP (Internet Protocol) .....	141
5.6.4.1	Affectation d'adresses IP à des consoles de programmation et des dispositifs réseau.....	141
5.6.4.2	Vérification de l'adresse IP de votre console de programmation .....	143
5.6.4.3	Affectation d'une adresse IP à une CPU en ligne.....	144
5.6.4.4	Configuration d'une adresse IP pour une CPU dans votre projet.....	145
5.6.5	Test du réseau PROFINET.....	149
5.6.6	Localisation de l'adresse Ethernet (MAC) sur la CPU .....	150
5.6.7	Configuration de la synchronisation via le protocole NTP .....	151
5.6.8	Temps de mise en route, affectation de nom et d'adresse pour un appareil PROFINET .....	152
<b>6</b>	<b>Concepts de programmation.....</b>	<b>155</b>
6.1	Principes de conception d'un système d'automatisation .....	155
6.2	Organisation de votre programme utilisateur.....	156
6.3	Utilisation de blocs pour structurer votre programme .....	158
6.3.1	Bloc d'organisation (OB) .....	159
6.3.2	Fonction (FC) .....	160
6.3.3	Bloc fonctionnel (FB).....	161
6.3.4	Bloc de données (DB).....	162
6.4	Comprendre le concept de cohérence des données .....	163
6.5	Langage de programmation.....	165
6.5.1	Schéma à contacts (CONT).....	165
6.5.2	Logigramme (LOG).....	166
6.5.3	SCL .....	166
6.5.4	EN et ENO pour CONT, LOG et SCL.....	174
6.6	Protection.....	175
6.6.1	Protection d'accès pour la CPU .....	175
6.6.2	Protection du savoir-faire .....	176
6.6.3	Protection contre la copie .....	177
6.7	Chargement d'éléments de votre programme dans la CPU .....	179
6.8	Chargement à partir de la CPU.....	179
6.8.1	Copie des éléments du projet .....	179
6.8.2	Utilisation de la fonction de comparaison .....	181
6.9	Débogage et test du programme .....	181
6.9.1	Visualisation et forçage de données dans la CPU.....	181
6.9.2	Tables de visualisation et tables de forçage permanent .....	182
6.9.3	Affichage de l'usage des références croisées .....	182
6.9.4	Structure d'appel permettant de constater la hiérarchie d'appel .....	183
<b>7</b>	<b>Instructions de base.....</b>	<b>185</b>
7.1	Instructions logiques sur bits.....	185
7.1.1	Contacts et bobines .....	185
7.1.2	Instructions Mise à 1 et Mise à 0 .....	188
7.1.3	Instructions Front montant et Front descendant.....	190
7.2	Temporisations.....	193

7.3	Compteurs.....	202
7.4	Comparaison.....	208
7.4.1	Comparaison.....	208
7.4.2	Instructions Valeur dans la plage et Valeur en dehors de la plage.....	209
7.4.3	Instructions Contrôler validité et Contrôler invalidité.....	210
7.5	Fonctions mathématiques.....	211
7.5.1	Instruction Calculer.....	211
7.5.2	Instructions Addition, Soustraction, Multiplication et Division.....	212
7.5.3	Instruction Modulo.....	213
7.5.4	Instruction Négation.....	214
7.5.5	Instructions Incrémenter et Décrémenter.....	215
7.5.6	Instruction Valeur absolue.....	215
7.5.7	Instructions Calculer le minimum et Calculer le maximum.....	216
7.5.8	Instruction Définir une limite.....	217
7.5.9	Instructions mathématiques à virgule flottante.....	218
7.6	Transfert.....	220
7.6.1	Instructions Copier valeur et Copier zone.....	220
7.6.2	Instructions FieldRead et FieldWrite.....	222
7.6.3	Instructions Compléter.....	224
7.6.4	Instruction Permutation.....	226
7.7	Convertir.....	227
7.7.1	Instruction CONV.....	227
7.7.2	Instructions de conversion pour SCL.....	228
7.7.3	Instructions Arrondir et Former un nombre entier.....	231
7.7.4	Instructions Arrondir à l'entier supérieur et Arrondir à l'entier inférieur.....	232
7.7.5	Instructions Mise à l'échelle et Normaliser.....	233
7.8	Gestion du programme.....	236
7.8.1	Présentation des instructions de contrôle SCL.....	236
7.8.2	Instruction IF-THEN.....	237
7.8.3	Instruction CASE.....	238
7.8.4	Instruction FOR.....	239
7.8.5	Instruction WHILE-DO.....	240
7.8.6	Instruction REPEAT-UNTIL.....	241
7.8.7	Instruction CONTINUE.....	242
7.8.8	Instruction EXIT.....	243
7.8.9	Instruction GOTO.....	243
7.8.10	Instruction RETURN.....	244
7.8.11	Instructions Saut et Repère de saut.....	244
7.8.12	Instruction JMP_LIST.....	245
7.8.13	Instruction SWITCH.....	246
7.8.14	Instruction de gestion d'exécution RET.....	248
7.8.15	Instruction Redéclencher le temps de surveillance du cycle.....	249
7.8.16	Instruction Commuter en STOP.....	251
7.8.17	Instructions de traitement d'erreur.....	251
7.9	Opérations logiques sur mots.....	255
7.9.1	Instructions AND, OR et XOR.....	255
7.9.2	Instruction Inverser.....	256
7.9.3	Instructions Encoder et Décoder.....	256
7.9.4	Instructions Sélectionner, Multiplexeur et Démultiplexeur.....	258



7.10	Décalage et rotation.....	261
7.10.1	Instructions de décalage.....	261
7.10.2	Instructions de rotation.....	262
<b>8</b>	<b>Instructions avancées.....</b>	<b>263</b>
8.1	Date et heure.....	263
8.1.1	Instructions sur la date et l'heure.....	263
8.1.2	Régler et lire l'horloge système.....	265
8.1.3	Instruction Compteur d'heures de fonctionnement.....	267
8.1.4	Instruction SET_TIMEZONE.....	269
8.2	Chaînes de caractères et caractères.....	270
8.2.1	Présentation des données chaînes de caractères (type String).....	270
8.2.2	Instruction S_MOVE.....	271
8.2.3	Instructions de conversion de chaîne.....	271
8.2.3.1	Conversions de chaîne en valeur et de valeur en chaîne.....	271
8.2.3.2	Conversions de chaîne en caractères et de caractères en chaîne.....	280
8.2.3.3	Conversions de ASCII en hexadécimal et de hexadécimal en ASCII.....	282
8.2.4	Instructions sur chaîne.....	284
8.2.4.1	LEN.....	284
8.2.4.2	CONCAT.....	285
8.2.4.3	LEFT, RIGHT et MID.....	286
8.2.4.4	DELETE.....	287
8.2.4.5	INSERT.....	288
8.2.4.6	REPLACE.....	289
8.2.4.7	FIND.....	290
8.3	Périphérie décentralisée (PROFINET, PROFIBUS ou AS-i).....	291
8.3.1	Instructions de périphérie décentralisée.....	291
8.3.2	RDREC et WRREC.....	292
8.3.3	RALRM.....	295
8.3.4	Paramètre STATUS pour RDREC, WRREC et RALRM.....	298
8.3.5	DPRD_DAT et DPWR_DAT.....	302
8.3.6	DPNRM_DG.....	304
8.4	Alarmes.....	307
8.4.1	Instructions ATTACH et DETACH.....	307
8.4.2	Alarmes cycliques.....	310
8.4.2.1	SET_CINT (Régler alarme cyclique).....	310
8.4.2.2	QRY_CINT (Interroger alarme cyclique).....	312
8.4.3	Alarmes temporisées.....	314
8.4.4	Alarmes asynchrones.....	316
8.5	Diagnostic (PROFINET ou PROFIBUS).....	317
8.5.1	Opérations de diagnostic.....	317
8.5.2	Événements de diagnostic d'une périphérie décentralisée.....	317
8.5.3	Instruction LED.....	318
8.5.4	Instruction DeviceStates.....	320
8.5.5	Instruction ModuleStates.....	321
8.5.6	Instruction GET_DIAG.....	322
8.6	Impulsion.....	330
8.6.1	Instruction CTRL_PWM.....	330
8.6.2	Fonctionnement des sorties d'impulsions.....	332
8.6.3	Configuration d'une voie d'impulsion pour PWM.....	334

8.7	Consignation de données .....	335
8.7.1	Structure des enregistrements de journaux .....	336
8.7.2	Instructions de gestion des journaux de données .....	337
8.7.2.1	DataLogCreate .....	337
8.7.2.2	DataLogOpen .....	340
8.7.2.3	DataLogClose .....	342
8.7.2.4	DataLogWrite .....	343
8.7.2.5	DataLogNewFile .....	344
8.7.3	Utilisation des journaux de données .....	346
8.7.4	Limites de taille des fichiers journaux .....	348
8.7.5	Exemple de programme pour les journaux de données .....	351
8.8	Gestion des blocs de données .....	355
8.8.1	READ_DBL, WRIT_DBL (Lecture ou écriture dans un DB en mémoire de chargement) .....	355
8.9	Codes d'erreur communs pour les instructions avancées .....	358
<b>9</b>	<b>Instructions technologiques .....</b>	<b>359</b>
9.1	Compteur rapide .....	359
9.1.1	Fonctionnement du compteur rapide .....	361
9.1.2	Configuration du HSC .....	368
9.2	Régulation PID .....	369
9.2.1	Insertion de l'instruction PID et de l'objet technologique .....	371
9.2.2	Instruction PID_Compact .....	374
9.2.3	Paramètre ErrorBits de l'instruction PID_Compact .....	379
9.2.4	Instruction PID_3STEP .....	380
9.2.5	Paramètre ErrorBits de l'instruction PID_3STEP .....	386
9.2.6	Configuration du régulateur PID .....	388
9.2.7	Mise en service du régulateur PID .....	390
9.3	Commande de mouvement .....	391
9.3.1	Configuration de l'axe .....	395
9.3.2	Configuration de la table de commande TO_CommandTable_PTO .....	398
9.3.3	Instructions de commande de mouvement (Motion Control) .....	401
9.3.3.1	Instruction MC_Power .....	401
9.3.3.2	Instruction MC_Reset .....	404
9.3.3.3	Instruction MC_Home .....	405
9.3.3.4	Instruction MC_Halt .....	408
9.3.3.5	Instruction MC_MoveAbsolute .....	410
9.3.3.6	Instruction MC_MoveRelative .....	412
9.3.3.7	Instruction MC_MoveVelocity .....	414
9.3.3.8	Instruction MC_MoveJog .....	417
9.3.3.9	Instruction MC_CommandTable .....	419
9.3.3.10	MC_ChangeDynamic .....	422
9.3.4	Fonctionnement de la commande de mouvement pour le S7-1200 .....	424
9.3.4.1	Sorties de la CPU pour la commande de mouvement .....	424
9.3.4.2	Fins de course matériels et logiciels pour la commande de mouvement .....	426
9.3.4.3	Référencement .....	429
9.3.4.4	Limitation d'à-coup .....	435
9.3.5	Mise en service .....	436
9.3.6	Suivi des commandes actives .....	439
9.3.6.1	Suivi des instructions MC avec paramètre de sortie "Done" .....	439
9.3.6.2	Suivi de l'instruction MC_Velocity .....	443

9.3.6.3	Suivi de l'instruction MC_MoveJog .....	447
<b>10</b>	<b>Communication.....</b>	<b>451</b>
10.1	Nombre de liaisons de communication asynchrones prises en charge.....	452
10.2	PROFINET .....	453
10.2.1	Liaison appareil local/partenaire .....	453
10.2.2	Communication ouverte (Open User Communication) .....	455
10.2.2.1	ID de liaison pour les instructions PROFINET .....	455
10.2.2.2	Protocoles .....	458
10.2.2.3	Mode ad hoc .....	459
10.2.2.4	TCP et ISO sur TCP .....	460
10.2.2.5	UDP.....	475
10.2.2.6	T_CONFIG .....	480
10.2.2.7	Paramètres communs des instructions.....	487
10.2.3	Communication avec une console de programmation .....	489
10.2.3.1	Etablissement de la liaison de communication matérielle .....	489
10.2.3.2	Configuration des appareils .....	490
10.2.3.3	Affectation d'adresses IP (Internet Protocol) .....	491
10.2.3.4	Test de votre réseau PROFINET .....	491
10.2.4	Communication IHM vers automate .....	491
10.2.4.1	Configuration des liaisons réseau logiques entre deux appareils .....	493
10.2.5	Communication API-API .....	493
10.2.5.1	Configuration des liaisons réseau logiques entre deux appareils .....	494
10.2.5.2	Configuration du routage local/partenaire entre deux appareils.....	494
10.2.5.3	Configuration des paramètres d'émission et de réception.....	495
10.2.6	Configuration d'une CPU et d'un périphérique PROFINET IO .....	497
10.2.7	Diagnostic .....	501
10.2.8	Instructions de périphérie décentralisée .....	501
10.2.9	Instructions de diagnostic .....	502
10.2.10	Événements de diagnostic pour la périphérie décentralisée .....	502
10.3	PROFIBUS.....	502
10.3.1	Modules de communication pour PROFIBUS .....	503
10.3.1.1	Connexion à PROFIBUS .....	503
10.3.1.2	Services de communication des CM PROFIBUS .....	504
10.3.1.3	Autres propriétés des CM PROFIBUS.....	506
10.3.1.4	Exemples de configuration pour PROFIBUS.....	507
10.3.2	Configuration d'un maître et d'un esclave DP.....	508
10.3.2.1	Ajout du module CM 1243-5 (maître DP) et d'un esclave DP .....	508
10.3.2.2	Configuration des liaisons réseau logiques entre deux appareils PROFIBUS .....	509
10.3.2.3	Affectation d'adresses PROFIBUS au module CM 1243-5 et à l'esclave DP.....	509
10.3.3	Instructions de périphérie décentralisée .....	511
10.3.4	Instructions de diagnostic .....	511
10.3.5	Événements de diagnostic pour périphérie décentralisée .....	511
10.4	Interface AS-i .....	511
10.4.1	Configuration d'un maître et d'un esclave AS-i.....	512
10.4.1.1	Ajout du module maître AS-i CM 1243-2 et d'un esclave AS-i .....	512
10.4.1.2	Configuration des liaisons réseau logiques entre deux appareils AS-i.....	513
10.4.1.3	Configuration des propriétés du maître AS-i CM 1243-2.....	513
10.4.1.4	Affectation d'une adresse AS-i à un esclave AS-i .....	514
10.4.2	Echange de données entre le programme utilisateur et les esclaves AS-i .....	515
10.4.2.1	Configuration STEP 7 de base .....	515

10.4.2.2	Configuration d'esclaves avec STEP 7 .....	516
10.4.3	Instructions de périphérie décentralisée .....	518
10.4.4	Utilisation d'outils en ligne AS-i .....	518
10.5	Communication S7 .....	520
10.5.1	Instructions GET et PUT .....	520
10.5.2	Création d'une liaison S7 .....	524
10.5.3	Configuration du routage local/partenaire entre deux appareils .....	524
10.5.4	Paramétrage de la liaison GET/PUT .....	525
10.5.4.1	Paramètres de la liaison .....	525
10.5.4.2	Configuration d'une liaison S7 CPU à CPU .....	528
<b>11</b>	<b>Serveur Web .....</b>	<b>533</b>
11.1	Activation du serveur Web .....	534
11.2	Pages Web standard .....	535
11.2.1	Accès aux pages Web standard depuis le PC .....	535
11.2.2	Disposition des pages Web standard .....	537
11.2.3	Introduction .....	539
11.2.4	Page de démarrage .....	540
11.2.5	Identification .....	541
11.2.6	Mémoire tampon de diagnostic .....	541
11.2.7	Informations sur les modules .....	542
11.2.8	Communication .....	545
11.2.9	Etat des variables .....	546
11.2.10	Journaux .....	548
11.2.11	Mise à jour du firmware .....	551
11.3	Pages Web personnalisées .....	553
11.3.1	Création de pages HTML .....	554
11.3.2	Commandes AWP prises en charge par le serveur Web du S7-1200 .....	555
11.3.2.1	Lecture de variables .....	556
11.3.2.2	Ecriture de variables .....	557
11.3.2.3	Lecture de variables spéciales .....	559
11.3.2.4	Ecriture de variables spéciales .....	561
11.3.2.5	Utilisation d'un alias pour une référence de variable .....	562
11.3.2.6	Définition de types Enum .....	563
11.3.2.7	Référencement de variables CPU avec un type Enum .....	564
11.3.2.8	Création de fragments .....	565
11.3.2.9	Importation de fragments .....	566
11.3.2.10	Combinaison de définitions .....	567
11.3.2.11	Gestion des noms de variables contenant des caractères spéciaux .....	567
11.3.3	Configuration de l'utilisation de pages Web personnalisées .....	569
11.3.4	Programmation de l'instruction WWW pour les pages Web personnalisées .....	571
11.3.5	Chargement des blocs de programme dans la CPU .....	573
11.3.6	Accès aux pages Web personnalisées depuis le PC .....	573
11.3.7	Contraintes spécifiques aux pages Web personnalisées .....	573
11.3.8	Exemple de page Web personnalisée .....	574
11.3.8.1	Page Web pour le contrôle-commande d'une éolienne .....	574
11.3.8.2	Lecture et affichage des données de l'automate .....	577
11.3.8.3	Utilisation d'un type Enum .....	577
11.3.8.4	Ecriture d'une entrée utilisateur dans l'automate .....	578
11.3.8.5	Ecriture d'une variable spéciale .....	579
11.3.8.6	Référence : liste HTML de la page Web de contrôle d'éolienne à distance .....	580

11.3.8.7	Configuration de l'exemple de page Web dans STEP 7.....	584
11.3.9	Configuration de pages Web personnalisées dans plusieurs langues.....	586
11.3.9.1	Création de la structure de dossiers.....	586
11.3.9.2	Programmation du changement de langue.....	587
11.3.9.3	Configuration de STEP 7 pour qu'il utilise une structure de page multilingue.....	589
11.3.10	Commande avancée de pages Web personnalisées.....	590
11.4	Contraintes.....	593
11.4.1	Fonctionnalités limitées lorsque JavaScript est désactivé.....	594
11.4.2	Fonctions restreintes lorsque les cookies ne sont pas autorisés.....	595
11.4.3	Importation du certificat de sécurité Siemens.....	596
11.4.4	Importation de journaux en format CSV dans des versions non anglaises/américaines de Microsoft Excel.....	597
<b>12</b>	<b>Processeur de communication.....</b>	<b>599</b>
12.1	Utilisation des interfaces de communication série.....	599
12.2	Polarisation et terminaison d'un connecteur de réseau RS485.....	600
12.3	Communication point à point (PtP).....	602
12.3.1	Instructions point à point.....	603
12.3.1.1	Paramètres communs pour les instructions point à point.....	603
12.3.1.2	Instruction PORT_CFG.....	605
12.3.1.3	Instruction SEND_CFG.....	607
12.3.1.4	Instruction RCV_CFG.....	609
12.3.1.5	Instruction SEND_PTP.....	613
12.3.1.6	Instruction RCV_PTP.....	617
12.3.1.7	Instruction RCV_RST.....	618
12.3.1.8	Instruction SGN_GET.....	619
12.3.1.9	Instruction SGN_SET.....	621
12.3.2	Configuration des ports de communication.....	622
12.3.2.1	Gestion du contrôle de flux.....	624
12.3.3	Configuration des paramètres d'émission et de réception.....	625
12.3.3.1	Configuration des paramètres d'émission.....	626
12.3.3.2	Configuration des paramètres de réception.....	627
12.3.4	Programmation de la communication point à point.....	635
12.3.4.1	Architecture d'interrogation.....	636
12.3.5	Exemple : Communication point à point.....	637
12.3.5.1	Configuration du module de communication.....	638
12.3.5.2	Configuration de RS422 et RS485.....	641
12.3.5.3	Programmation du programme STEP 7.....	644
12.3.5.4	Configuration de l'émulateur de terminal.....	645
12.3.5.5	Exécution de l'exemple.....	646
12.4	Communication USS (interface série universelle).....	646
12.4.1	Conditions requises pour l'utilisation du protocole USS.....	648
12.4.2	Instruction USS_DRV.....	651
12.4.3	Instruction USS_PORT.....	654
12.4.4	Instruction USS_RPM.....	655
12.4.5	Instruction USS_WPM.....	656
12.4.6	Codes d'état USS.....	658
12.4.7	Informations générales sur la configuration d'un entraînement.....	660
12.5	Communication Modbus.....	663
12.5.1	Présentation de la communication Modbus RTU et TCP.....	663

12.5.2	Modbus TCP .....	666
12.5.2.1	MB_CLIENT (Modbus TCP).....	666
12.5.2.2	MB_SERVER (Modbus TCP).....	673
12.5.2.3	Exemple pour MB_SERVER : Liaisons TCP multiples .....	678
12.5.2.4	Exemple 1 pour MB_CLIENT : Plusieurs demandes avec une liaison TCP commune .....	679
12.5.2.5	Exemple 2 pour MB_CLIENT : Plusieurs demandes avec des liaisons TCP différentes .....	680
12.5.2.6	Exemple 3 pour MB_CLIENT : Demande d'écriture dans la mémoire image des sorties .....	681
12.5.2.7	Exemple 4 pour MB_CLIENT : Coordination de plusieurs demandes .....	682
12.5.3	Modbus RTU .....	683
12.5.3.1	MB_COMM_LOAD .....	684
12.5.3.2	MB_MASTER .....	687
12.5.3.3	MB_SLAVE .....	693
12.5.3.4	Exemple de programme maître Modbus RTU .....	699
12.5.3.5	Exemple de programme esclave Modbus RTU .....	701
12.6	Telecontrol et TeleService avec le CP 1242-7.....	702
12.6.1	Connexion à un réseau GSM.....	702
12.6.2	Application du CP 1242-7 .....	704
12.6.3	Autres propriétés du CP.....	705
12.6.4	Accessoires .....	706
12.6.5	Exemples de configuration pour Telecontrol.....	707
<b>13</b>	<b>Communication TeleService (courrier électronique SMTP) .....</b>	<b>713</b>
13.1	Instruction TM_Mail (Transférer e-mail).....	713
<b>14</b>	<b>Outils en ligne et de diagnostic.....</b>	<b>721</b>
14.1	DEL d'état.....	721
14.2	Passage en ligne et connexion à une CPU .....	723
14.3	Affectation d'un nom à un périphérique PROFINET IO en ligne .....	724
14.4	Réglage de l'adresse IP et de l'heure .....	726
14.5	Restauration des réglages d'usine.....	726
14.6	Panneau de commande CPU de la CPU en ligne .....	727
14.7	Surveillance du temps de cycle et de l'utilisation de la mémoire.....	728
14.8	Affichage des événements de diagnostic dans la CPU .....	728
14.9	Comparaison de CPU hors ligne et en ligne.....	729
14.10	Visualisation et forçage de valeurs dans la CPU.....	730
14.10.1	Passage en ligne pour visualiser les valeurs dans la CPU.....	731
14.10.2	Affichage de l'état dans l'éditeur de programme.....	731
14.10.3	Acquisition des valeurs en ligne d'un DB pour redéfinir les valeurs initiales .....	732
14.10.4	Utilisation d'une table de visualisation pour visualiser et forcer des valeurs dans la CPU.....	733
14.10.4.1	Utilisation d'un déclenchement lors de la visualisation ou du forçage de variables API .....	735
14.10.4.2	Déblocage des sorties à l'état ARRÊT.....	736
14.10.5	Forçage permanent de valeurs dans la CPU.....	736
14.10.5.1	Utilisation de la table de forçage permanent .....	736
14.10.5.2	Fonctionnement de la fonction de forçage permanent .....	737
14.11	Chargement dans la CPU à l'état MARCHE.....	739
14.11.1	Conditions requises pour un chargement dans la CPU à l'état MARCHE.....	740
14.11.2	Modification du programme à l'état "Marche" .....	741

14.11.3	Chargement de blocs sélectionnés.....	742
14.11.4	Chargement d'un bloc individuel sélectionné alors qu'un autre bloc présente une erreur de compilation.....	743
14.11.5	Réaction du système en cas d'échec de l'opération de chargement.....	744
14.11.6	Chargement du programme dans la CPU à l'état MARCHE .....	745
<b>A</b>	<b>Caractéristiques techniques.....</b>	<b>747</b>
A.1	Caractéristiques techniques d'ordre général .....	747
A.2	CPU 1211C.....	754
A.2.1	Caractéristiques et fonctions générales.....	754
A.2.2	Temporisations, compteurs et blocs de code pris en charge par la CPU 1211C.....	756
A.2.3	Entrées et sorties TOR .....	758
A.2.4	Entrées analogiques .....	760
A.2.4.1	Réponse indicielle des entrées analogiques intégrées de la CPU.....	760
A.2.4.2	Temps d'échantillonnage pour les ports analogiques intégrés de la CPU .....	761
A.2.5	Schémas de câblage de la CPU 1211C .....	761
A.3	CPU 1212C.....	764
A.3.1	Caractéristiques et fonctions générales.....	764
A.3.2	Temporisations, compteurs et blocs de code pris en charge par la CPU 1212C.....	766
A.3.3	Entrées et sorties TOR .....	768
A.3.4	Entrées analogiques .....	769
A.3.4.1	Réponse indicielle des entrées analogiques intégrées de la CPU.....	770
A.3.4.2	Temps d'échantillonnage pour les ports analogiques intégrés de la CPU .....	770
A.3.5	Schémas de câblage de la CPU 1212C .....	771
A.4	CPU 1214C.....	774
A.4.1	Caractéristiques et fonctions générales.....	774
A.4.2	Temporisations, compteurs et blocs de code pris en charge par la CPU 1214C.....	775
A.4.3	Entrées et sorties TOR .....	778
A.4.4	Entrées analogiques .....	779
A.4.4.1	Réponse indicielle des entrées analogiques intégrées de la CPU.....	780
A.4.4.2	Temps d'échantillonnage pour les ports analogiques intégrés de la CPU .....	780
A.4.5	Schémas de câblage de la CPU 1214C .....	781
A.5	CPU 1215C.....	784
A.5.1	Caractéristiques et fonctions générales.....	784
A.5.2	Temporisations, compteurs et blocs de code pris en charge par la CPU 1215C.....	786
A.5.3	Entrées et sorties TOR .....	788
A.5.4	Entrées et sorties analogiques.....	790
A.5.4.1	Caractéristiques des entrées analogiques.....	790
A.5.4.2	Réponse indicielle des entrées analogiques intégrées de la CPU.....	790
A.5.4.3	Temps d'échantillonnage pour les ports analogiques intégrés de la CPU .....	791
A.5.4.4	Caractéristiques des sorties analogiques .....	791
A.5.5	Schémas de câblage de la CPU 1215C .....	792
A.6	Modules d'entrées-sorties TOR (SM) .....	796
A.6.1	Caractéristiques des modules d'entrées TOR SM 1221.....	796
A.6.2	Caractéristiques des modules 8 sorties TOR SM 1222.....	798
A.6.3	Caractéristiques des modules 16 sorties TOR SM 1222.....	799
A.6.4	Caractéristiques des modules d'entrées/sorties TOR SM 1223 VDC .....	804
A.6.5	Caractéristiques du module d'entrées/sorties TOR SM 1223 VAC .....	809
A.7	Modules d'entrées-sorties analogiques (SM).....	812

A.7.1	Caractéristiques des modules d'entrées analogiques SM 1231 .....	812
A.7.2	Caractéristiques des modules de sorties analogiques SM 1232 .....	816
A.7.3	Caractéristiques du module d'entrées/sorties analogiques SM 1234 .....	818
A.7.4	Réponse indicielle des entrées analogiques.....	821
A.7.5	Temps d'échantillonnage et temps d'actualisation pour les entrées analogiques .....	821
A.7.6	Plages de mesure des entrées analogiques pour la tension .....	822
A.7.7	Plages de mesure des entrées analogiques pour le courant.....	822
A.7.8	Plages de mesures de sortie (AQ) pour tension et courant (SB et SM) .....	823
A.8	Modules d'entrées-sorties (SM) Thermocouple et RTD.....	824
A.8.1	SM 1231 Thermocouple.....	824
A.8.1.1	Principe de fonctionnement des thermocouples .....	827
A.8.1.2	Tableaux de sélection pour les SM 1231 Thermocouple.....	828
A.8.2	SM 1231 RTD .....	830
A.8.2.1	Tableaux de sélection pour les SM 1231 RTD .....	833
A.9	Signal Boards (SB) TOR.....	836
A.9.1	Caractéristiques des SB 1221 entrées TOR, 200 kHz .....	836
A.9.2	Caractéristiques des SB 1222 sorties TOR, 200 kHz.....	838
A.9.3	Caractéristiques des SB 1223 entrées/sorties TOR, 200 kHz.....	841
A.9.4	Caractéristiques du SB 1223 2 entrées 24 VDC / 2 sorties 24 VDC .....	844
A.10	Signal Boards (SB) analogiques.....	846
A.10.1	Caractéristiques du SB 1231 1 sortie analogique.....	846
A.10.2	Caractéristiques du SB 1232 1 sortie analogique.....	849
A.10.3	Plages de mesure pour les entrées et sorties analogiques .....	851
A.10.3.1	Réponse indicielle des entrées analogiques.....	851
A.10.3.2	Temps d'échantillonnage et temps d'actualisation pour les entrées analogiques .....	851
A.10.3.3	Plages de mesure des entrées analogiques pour la tension .....	852
A.10.3.4	Plages de mesure des entrées analogiques pour le courant.....	852
A.10.3.5	Plages de mesures de sortie (AQ) pour tension et courant (SB et SM) .....	853
A.10.4	SB Thermocouple .....	854
A.10.4.1	Caractéristiques du SB 1231 1 entrée analogique Thermocouple .....	854
A.10.4.2	Principe de fonctionnement des thermocouples .....	855
A.10.5	Signal Boards (SB) RTD .....	858
A.10.5.1	Caractéristiques du SB 1231 1 entrée analogique RTD .....	858
A.10.5.2	Tableaux de sélection pour le SB 1231 RTD.....	861
A.11	Battery Board BB 1297 .....	863
A.12	Interfaces de communication .....	864
A.12.1	PROFIBUS.....	864
A.12.1.1	CM 1242-5.....	864
A.12.1.2	CM 1243-5.....	866
A.12.2	GPRS .....	868
A.12.2.1	CP 1242-7 .....	868
A.12.3	CM 1243-2 maître AS-i .....	871
A.12.3.1	Données techniques pour l'interface AS-i master CM 1243-2.....	871
A.12.3.2	Connexions électriques de l'interface AS-i master CM 1243-2 .....	872
A.12.4	RS232, RS422 et RS485 .....	873
A.12.4.1	Caractéristiques RS485 CB 1241 .....	873
A.12.4.2	Caractéristiques du module CM 1241 RS232.....	875
A.12.4.3	Caractéristiques du module CM 1241 RS422/485 .....	877
A.13	TeleService (TS Adapter et TS Adapter modulaire) .....	878



---

A.14	Cartes mémoire SIMATIC .....	878
A.15	Simulateurs d'entrées .....	879
A.16	Câble d'extension d'E/S .....	880
A.17	Produits connexes .....	881
A.17.1	Module d'alimentation PM 1207 .....	881
A.17.2	Module commutateur compact CSM 1277 .....	881
<b>B</b>	<b>Calcul d'un bilan de consommation .....</b>	<b>883</b>
<b>C</b>	<b>Numéros de référence .....</b>	<b>887</b>
C.1	Modules CPU .....	887
C.2	Modules d'entrées-sorties (SM), Signal Boards (SB) et Battery Boards (BB) .....	887
C.3	Communication .....	888
C.4	Autres modules .....	889
C.5	Cartes mémoire.....	890
C.6	Appareils IHM Basic.....	890
C.7	Pièces détachées et autres matériels.....	890
C.8	Logiciel de programmation.....	891
C.9	Documentation .....	891
	<b>Index.....</b>	<b>893</b>



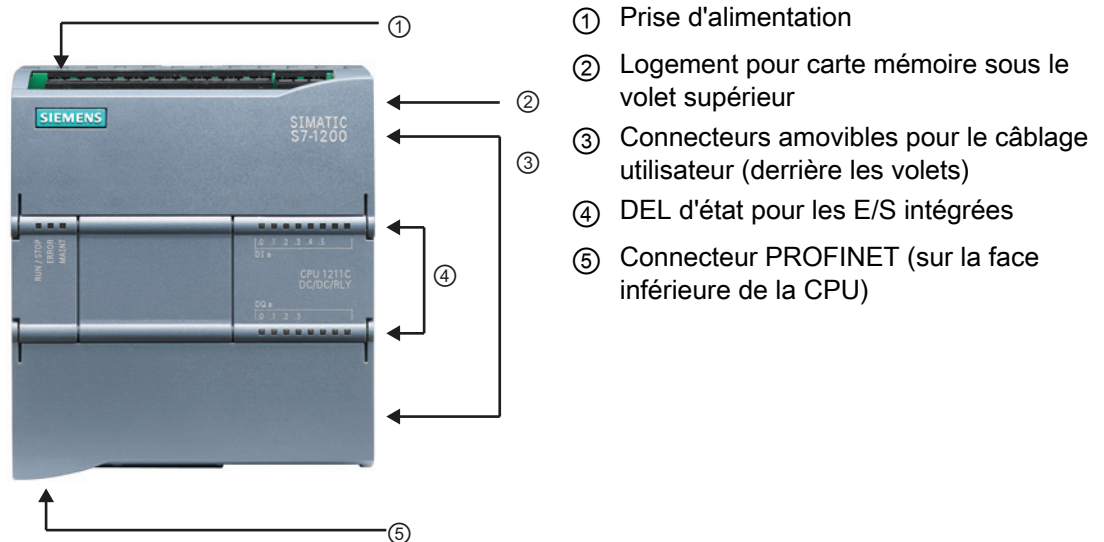
## Présentation du produit

### 1.1 Introduction à l'automate S7-1200

Le contrôleur S7-1200 offre la souplesse et la puissance nécessaires pour commander une large gamme d'appareils afin de répondre à vos besoins en matière d'automatisation. Sa forme compacte, sa configuration souple et son important jeu d'instructions en font une solution idéale pour la commande d'applications très variées.

La CPU combine un microprocesseur, une alimentation intégrée, des circuits d'entrée et de sortie, un PROFINET intégré, des E/S rapides de commande de mouvement, ainsi que des entrées analogiques intégrées dans un boîtier compact en vue de créer un contrôleur puissant. Une fois que vous avez chargé votre programme, la CPU contient la logique nécessaire au contrôle et à la commande des appareils dans votre application. La CPU surveille les entrées et modifie les sorties conformément à la logique de votre programme utilisateur, qui peut contenir des instructions booléennes, des instructions de comptage, des instructions de temporisation, des instructions mathématiques complexes ainsi que des commandes pour communiquer avec d'autres appareils intelligents.

La CPU fournit un port PROFINET permettant de communiquer par le biais d'un réseau PROFINET. Des modules supplémentaires sont disponibles pour communiquer via les réseaux PROFIBUS, GPRS, RS485 ou RS232.



1.1 Introduction à l'automate S7-1200

Plusieurs fonctions de sécurité vous aident à protéger l'accès à la CPU et au programme de commande :

- Chaque CPU fournit une protection par mot de passe (Page 175) qui vous permet de configurer l'accès aux fonctions CPU.
- Vous pouvez utiliser la protection du savoir-faire (Page 176) (protection "know-how") pour masquer le code d'un bloc spécifique.
- Vous pouvez utiliser la protection contre la copie (Page 177) pour lier votre programme à une carte mémoire ou une CPU spécifique.

Tableau 1- 1 Comparaison des modèles de CPU

Caractéristique		CPU 1211C	CPU 1212C	CPU 1214C	CPU 1215C
Dimensions (mm)		90 x 100 x 75	90 x 100 x 75	110 x 100 x 75	130 x 100 x 75
Mémoire utilisateur	de travail	30 Ko	50 Ko	75 Ko	100 Ko
	de chargement	1 Mo	1 Mo	4 Mo	4 Mo
	rémanente	10 Ko	10 Ko	10 Ko	10 Ko
E/S intégrées locales	TOR	6 entrées/4 sorties	8 entrées/6 sorties	14 entrées/10 sorties	14 entrées/10 sorties
	Analogiques	2 entrées	2 entrées	2 entrées	2 entrées/2 sorties
Taille de la mémoire image	Entrées (I)	1024 octets	1024 octets	1024 octets	1024 octets
	Sorties (Q)	1024 octets	1024 octets	1024 octets	1024 octets
Mémentos (M)		4096 octets	4096 octets	8192 octets	8192 octets
Modules d'entrées-sorties (SM) pour extension		Aucun	2	8	8
Signal Board (SB), Battery Board (BB) ou Communication Board (CB)		1	1	1	1
Module de communication (CM) (extension vers la gauche)		3	3	3	3
Compteurs rapides	Total	3 intégrés, 5 avec SB	4 intégrés, 6 avec SB	6	6
	Monophasé	3 à 100 kHz SB : 2 à 30 kHz	3 à 100 kHz 1 à 30 kHz SB : 2 à 30 kHz	3 à 100 kHz 3 à 30 kHz	3 à 100 kHz 3 à 30 kHz
	Quadrature de phase	3 à 80 kHz SB : 2 à 20 kHz	3 à 80 kHz 1 à 20 kHz SB : 2 à 20 kHz	3 à 80 kHz 3 à 20 kHz	3 à 80 kHz 3 à 20 kHz
Sorties d'impulsions <sup>1</sup>		4	4	4	4
Carte mémoire		Carte mémoire SIMATIC (facultative)			
Durée de conservation de l'horloge temps réel		20 jours, typ. / 12 jours min. à 40 degrés C (supercondensateur sans maintenance)			
PROFINET		1 port de communication Ethernet			2 ports de communication Ethernet

Caractéristique	CPU 1211C	CPU 1212C	CPU 1214C	CPU 1215C
Vitesse d'exécution des instructions mathématiques sur réels	2,3 µs/instruction			
Vitesse d'exécution des instructions booléennes	0,08 µs/instruction			

- <sup>1</sup> Pour les modèles de CPU avec sorties relais, vous devez installer un Signal Board (SB) TOR pour utiliser les sorties d'impulsions.

Chaque CPU fournit des liaisons IHM spécialisées pour la prise en charge de 3 appareils IHM au maximum. Le nombre total de stations IHM dépend des types de pupitres IHM dans votre configuration. Vous pouvez, par exemple, avoir trois SIMATIC Basic Panels au plus connectés à votre CPU ou bien deux SIMATIC Comfort Panels au plus avec un Basic Panel additionnel.

Les différents modèles de CPU disposent de fonctions et éléments divers qui vous aident à créer des solutions efficaces pour vos diverses applications. Reportez-vous aux caractéristiques techniques (Page 747) pour des informations détaillées sur une CPU spécifique.

Tableau 1- 2 Blocs, temporisations et compteurs pris en charge par le S7-1200

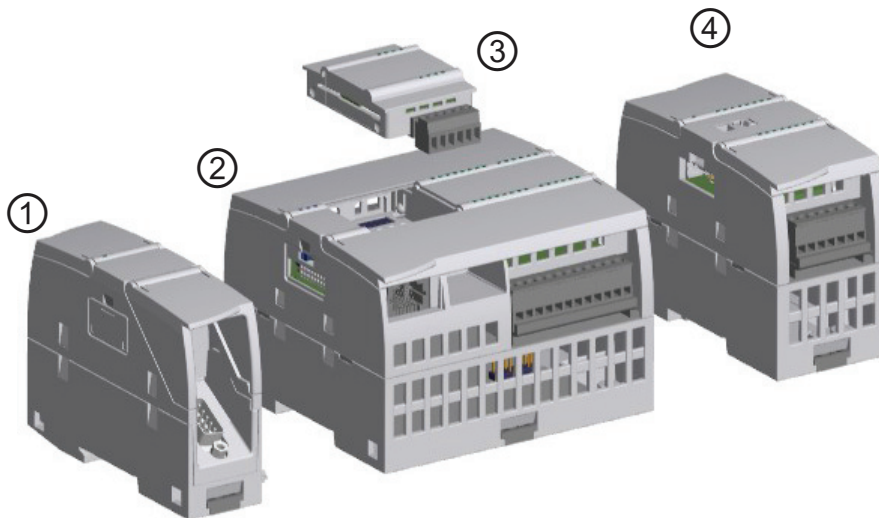
Élément	Description	
Blocs	Type	OB, FB, FC, DB
	Taille	30 Ko (CPU 1211C) 50 Ko (CPU 1212C) 64 Ko (CPU 1214C et CPU 1215C)
	Quantité	Jusqu'à 1024 blocs au total (OB + FB + FC + DB)
	Plage d'adresses pour les FB, FC et DB	1 à 65535 (FB 1 à FB 65535, par exemple)
	Profondeur d'imbrication	16 à partir de l'OB de cycle de programme ou de démarrage ; 4 à partir de l'OB d'alarme temporisée, d'alarme horaire, d'alarme cyclique, d'alarme de processus, d'alarme d'erreur de temps ou d'alarme de diagnostic
	Visualisation	Il est possible de visualiser simultanément l'état de 2 blocs de code.
	OB	Cycle de programme
Mise en route		Plusieurs : OB 100, OB 200 à OB 65535
Alarmes temporisées et alarmes cycliques		4 <sup>1</sup> (1 par événement) : OB 200 à OB 65535
Alarmes de processus (fronts et HSC)		50 (1 par événement) : OB 200 à OB 65535
Alarmes d'erreur de temps		1: OB 80
Alarmes de diagnostic		1: OB 82
Temporisations	Type	CEI
	Quantité	Limité uniquement par la taille de la mémoire
	Stockage	Structure dans un DB, 16 octets par temporisation
Compteurs	Type	CEI
	Quantité	Limité uniquement par la taille de la mémoire

Élément	Description
Stockage	Structure dans un DB, la taille dépend du type de compteur <ul style="list-style-type: none"><li>• SInt, USInt : 3 octets</li><li>• Int, UInt : 6 octets</li><li>• DInt, UDInt : 12 octets</li></ul>

<sup>1</sup> Les alarmes temporisées et cycliques utilisent les mêmes ressources dans la CPU. Vous pouvez avoir uniquement 4 de ces alarmes au total (temporisées et cycliques). Vous ne pouvez pas avoir 4 alarmes temporisées et 4 alarmes cycliques.

## 1.2 Possibilités d'extension de la CPU

La gamme S7-1200 offre divers modules et cartes enfichables pour accroître les capacités de la CPU avec des E/S supplémentaires ou d'autres protocoles de communication. Référez-vous aux caractéristiques techniques (Page 747) pour des informations détaillées sur un module spécifique.



- ① Module de communication (CM), processeur de communication (CP) ou adaptateur TS Adapter
- ② CPU
- ③ Signal Board (SB), Communication Board (CB) ou Battery Board (BB)
- ④ Module d'entrées-sorties (SM)

Tableau 1- 3 Modules d'entrées-sorties et Signal Boards TOR

Type	Entrées uniquement	Sorties uniquement	Entrées/sorties combinées
③ SB TOR	<ul style="list-style-type: none"> <li>• 4 entrées 24VDC, 200 kHz</li> <li>• 4 entrées 5VDC, 200 kHz</li> </ul>	<ul style="list-style-type: none"> <li>• 4 sorties 24VDC, 200 kHz</li> <li>• 4 sorties 5VDC, 200 kHz</li> </ul>	<ul style="list-style-type: none"> <li>• 2 entrées 24VDC / 2 sorties 24VDC</li> <li>• 2 entrées 24VDC / 2 sorties 24VDC, 200 kHz</li> <li>• 2 entrées 5VDC / 2 sorties 5VDC, 200 kHz</li> </ul>
④ SM TOR	• 8 entrées 24VDC	<ul style="list-style-type: none"> <li>• 8 sorties 24VDC</li> <li>• 8 sorties Relais</li> <li>• 8 sorties Relais (inverseur)</li> </ul>	<ul style="list-style-type: none"> <li>• 8 entrées 24VDC / 8 sorties 24VDC</li> <li>• 8 entrées 24VDC / 8 sorties Relais</li> <li>• 8 entrées 120/230VAC / 8 sorties Relais</li> </ul>
	• 16 entrées 24VDC	<ul style="list-style-type: none"> <li>• 16 sorties 24VDC</li> <li>• 16 sorties Relais</li> </ul>	<ul style="list-style-type: none"> <li>• 16 entrées 24VDC / 16 sorties 24VDC</li> <li>• 16 entrées 24VDC / 16 sorties Relais</li> </ul>

Tableau 1- 4 Modules d'entrées-sorties et Signal Boards analogiques

Type	Entrées uniquement	Sorties uniquement	Entrées/sorties combinées
③ SB analogique	<ul style="list-style-type: none"> <li>• 1 entrée analogique 12 bits</li> <li>• 1 RTD 16 bits</li> <li>• 1 Thermocouple 16 bits</li> </ul>	• 1 sortie analogique	-
④ SM analogique	<ul style="list-style-type: none"> <li>• 4 entrées analogiques</li> <li>• 4 entrées analogiques 16 bits</li> <li>• 8 entrées analogiques</li> <li>• Thermocouple : <ul style="list-style-type: none"> <li>- 4 TC 16 bits</li> <li>- 8 TC 16 bits</li> </ul> </li> <li>• RTD : <ul style="list-style-type: none"> <li>- 4 RTD 16 bits</li> <li>- 8 RTD 16 bits</li> </ul> </li> </ul>	<ul style="list-style-type: none"> <li>• 2 sorties analogiques</li> <li>• 4 sorties analogiques</li> </ul>	• 4 entrées analogiques / 2 sorties analogiques

Tableau 1- 5 Interfaces de communication

Module	Type	Description
① Module de communication (CM)	RS232	Duplex intégral
	RS422/485	Duplex intégral (RS422) Semi-duplex (RS485)
	Maître PROFIBUS	DPV1
	Esclave PROFIBUS	DPV1
	Maître AS-i (CM 1243-2)	Interface AC
① Processeur de communication (CP)	Connectivité modem	GPRS
① Communication Board (CB)	RS485	Semi-duplex
① TeleService	Adaptateur TS Adapter IE Basic <sup>1</sup>	Connexion à la CPU
	Adaptateur TS Adapter GSM	GSM/GPRS
	Adaptateur TS Adapter Modem	Modem
	Adaptateur TS Adapter ISDN	RNIS
	Adaptateur TS Adapter RS232	RS232

<sup>1</sup> L'adaptateur TS Adapter permet la connexion de diverses interfaces de communication au port PROFINET de la CPU. Vous installez l'adaptateur TS sur le coté gauche de la CPU et connectez l'adaptateur TS Adapter modulaire (3 au plus) sur l'adaptateur TS.

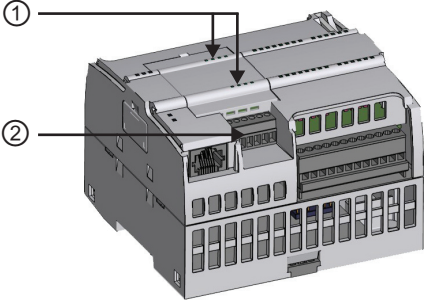
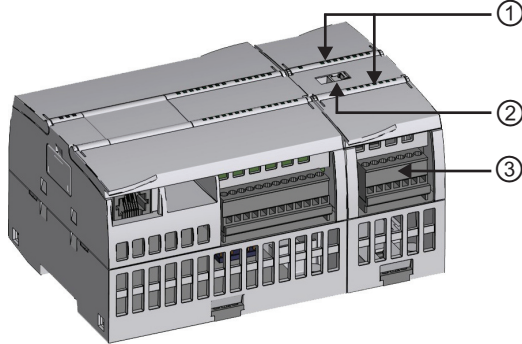
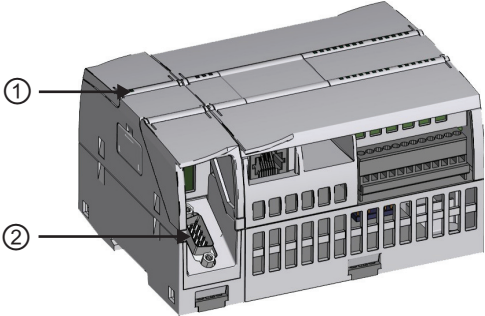
Tableau 1- 6 Autres modules

Module	Description
③ Battery Board	S'enfiche dans l'interface de carte d'extension à l'avant de la CPU. Permet une sauvegarde à long terme de l'horloge temps réel.



## 1.3 Modules S7-1200

Tableau 1- 7 Modules d'extension S7-1200

Type de module	Description		
<p>La CPU prend en charge une carte d'extension enfichable :</p> <ul style="list-style-type: none"> <li>Un Signal Board (SB) fournit des E/S supplémentaires pour votre CPU. Le SB se raccorde à l'avant de la CPU.</li> <li>Un Communication Board (CB) vous permet d'ajouter un autre port de communication à votre CPU.</li> <li>Un Battery Board (BB) permet une sauvegarde à long terme de l'horloge temps réel.</li> </ul>		<p>①</p> <p>②</p>	<p>DEL d'état sur le Signal Board</p> <p>Connecteur amovible pour le câblage utilisateur</p>
<p>Les modules d'entrées-sorties (SM) permettent d'ajouter des fonctionnalités à la CPU. Les SM se raccordent sur le côté droit de la CPU.</p> <ul style="list-style-type: none"> <li>E/S TOR</li> <li>E/S analogiques</li> <li>RTD et Thermocouple</li> </ul>		<p>①</p> <p>②</p> <p>③</p>	<p>DEL d'état</p> <p>Connecteur de bus</p> <p>Connecteur amovible pour le câblage utilisateur</p>
<p>Les modules de communication (CM) et les processeurs de communication (CP) ajoutent des options de communication à la CPU, telles que la connectivité PROFIBUS ou RS232 / RS485 (pour PtP, Modbus ou USS) ou le maître AS-i. Un CP offre la possibilité d'autres types de communication, par exemple la connexion de la CPU par le biais d'un réseau GPRS.</p> <ul style="list-style-type: none"> <li>La CPU accepte jusqu'à 3 CM ou CP.</li> <li>Chaque CM ou CP se raccorde sur le côté gauche de la CPU (ou sur le côté gauche d'un autre CM ou CP).</li> </ul>		<p>①</p> <p>②</p>	<p>DEL d'état</p> <p>Connecteur de communication</p>

## 1.4 Nouvelles fonctions

Cette version comporte les nouvelles fonctions suivantes :

- Une Page standard du serveur Web pour effectuer une mise à jour du firmware de la CPU (Page 551)
- La possibilité d'utiliser trois modules maîtres PROFIBUS DP CM 1243-5 ou trois modules maîtres CM 1243-2 AS-i

---

### Remarque

Pour utiliser trois modules AS-i comme maîtres, vous devez mettre à jour le firmware des modules AS-i.

---

### Nouveaux modules pour le S7-1200

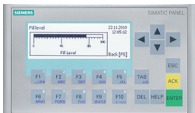
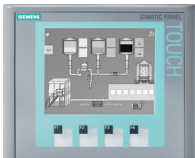
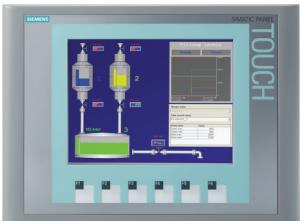
Divers nouveaux modules augmentent la puissance de la CPU S7-1200 et offrent la souplesse nécessaire à vos besoins en automatisation :

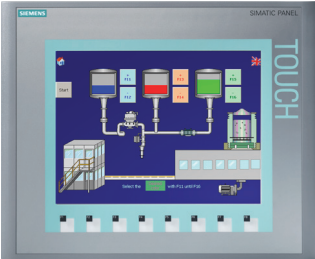
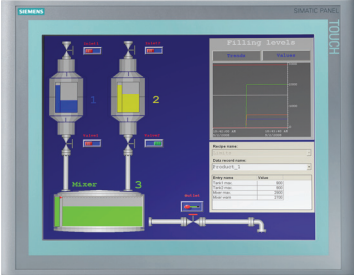
- Nouvelles CPU et CPU améliorées :
  - Les nouvelles CPU 1215C DC/DC/DC, CPU 1215C DC/DC/Relais et CPU 1215C AC/DC/Relais offrent 100 Ko de mémoire de travail, double Ethernet et des sorties analogiques.
  - Les CPU 1211C, CPU 1212C et CPU 1214C nouvelles et améliorées ont un temps de traitement plus rapide, 4 sorties PTO possibles (la CPU 1211C requiert un Signal Board), plus de mémoire rémanente (10 Ko) et un temps de conservation de l'horloge temps réel accru (20 jours).
- Nouveau module d'entrées/sorties : le SM 1231 AI 4 x 16 bits offre une fréquence d'échantillonnage plus élevée et un nombre de bits accru.
- Le nouveau Battery Board (BB 1297) offre une sauvegarde à long terme de l'horloge temps réel. Le BB 1297 s'enfiche dans le logement pour Signal Board de la CPU S7-1200 (firmware 3.0 et versions ultérieures).

Vous devez disposer de STEP 7 V11 SP2 Update 3 ou version ultérieure (Basic ou Professional) pour utiliser ces nouveaux modules et vous devez télécharger le paquet Hardware Support Package (HSP) pour nouveaux modules via la commande **Outils > Support Packages** de STEP 7. Suivez les instructions pour l'ajout de modules au catalogue matériel de STEP 7 (portail TIA) figurant sur le site Service & Support de Siemens (<http://support.automation.siemens.com>).

## 1.5 Tableaux IHM basiques

Comme la visualisation devient un composant standard pour la plupart des modèles de machine, les SIMATIC HMI Basic Panels sont composés d'écrans tactiles pour les tâches basiques de surveillance et de contrôle de l'opérateur. Tous les tableaux ont un indice de protection de IP65 et sont certifiés CE, UL, cULus et NEMA 4x.

Basic HMI Panel	Description	Caractéristiques techniques
 <p>KP 300 Basic PN</p>	<p>Clavier à membrane 3,6" avec 10 touches tactiles librement configurables</p> <ul style="list-style-type: none"> <li>• Monochrome (STN, noir/blanc)</li> <li>• 87 mm x 31 mm (3,6")</li> <li>• Rétroéclairé couleur programmé (blanc, vert, jaune ou rouge)</li> <li>• Résolution : 240 x 80</li> </ul>	<ul style="list-style-type: none"> <li>• 250 variables</li> <li>• 50 écrans de processus</li> <li>• 200 alarmes</li> <li>• 25 courbes</li> <li>• 40 Ko de mémoire des recettes</li> <li>• 5 recettes, 20 enregistrements, 20 entrées</li> </ul>
 <p>KTP 400 Basic PN</p>	<p>Écran tactile 4" avec 4 touches tactiles</p> <ul style="list-style-type: none"> <li>• Mono (STN, nuances de gris)</li> <li>• 76,79 mm x 57,59 mm (3,8") Portrait ou paysage</li> <li>• Résolution : 320 x 240</li> </ul>	<ul style="list-style-type: none"> <li>• 250 variables</li> <li>• 50 écrans de processus</li> <li>• 200 alarmes</li> <li>• 25 courbes</li> <li>• 40 Ko de mémoire des recettes</li> <li>• 5 recettes, 20 enregistrements, 20 entrées</li> </ul>
 <p>KTP 600 Basic PN</p>	<p>Écran tactile 6" avec 6 touches tactiles</p> <ul style="list-style-type: none"> <li>• Couleur (TFT, 256 couleurs) ou Mono (STN, nuances de gris)</li> <li>• 115,2 mm x 86,4 mm (5,7") Portrait ou paysage</li> <li>• Résolution : 320 x 240</li> </ul>	<ul style="list-style-type: none"> <li>• 500 variables</li> <li>• 50 écrans de processus</li> <li>• 200 alarmes</li> <li>• 25 courbes</li> <li>• 40 Ko de mémoire des recettes</li> <li>• 5 recettes, 20 enregistrements, 20 entrées</li> </ul>

Basic HMI Panel	Description	Caractéristiques techniques
	<p>Écran tactile 10" avec 8 touches tactiles</p> <ul style="list-style-type: none"> <li>• Couleur (TFT, 256 couleurs)</li> <li>• 211,2 mm x 158,4 mm (10,4")</li> <li>• Résolution : 640 x 480</li> </ul>	<ul style="list-style-type: none"> <li>• 500 variables</li> <li>• 50 écrans de processus</li> <li>• 200 alarmes</li> <li>• 25 courbes</li> <li>• 40 Ko de mémoire des recettes</li> <li>• 5 recettes, 20 enregistrements, 20 entrées</li> </ul>
<p>KTP 1000 Basic PN</p>	<p>Écran tactile 15"</p> <ul style="list-style-type: none"> <li>• Couleur (TFT, 256 couleurs)</li> <li>• 304,1 mm x 228,1 mm (15,1")</li> <li>• Résolution : 1024 x 768</li> </ul>	<ul style="list-style-type: none"> <li>• 500 variables</li> <li>• 50 écrans de processus</li> <li>• 200 alarmes</li> <li>• 25 courbes</li> <li>• 40 Ko de mémoire des recettes (flash intégrée)</li> <li>• 5 recettes, 20 enregistrements, 20 entrées</li> </ul>
	<p>TP 1500 Basic PN</p>	

## Logiciel de programmation STEP 7

STEP 7 fournit un environnement convivial pour concevoir, éditer et surveiller la logique nécessaire à la commande de votre application, et notamment les outils pour gérer et configurer tous les appareils dans votre projet, tels que des automates et appareils IHM. Pour vous aider à trouver les informations dont vous avez besoin, STEP 7 fournit un système d'aide en ligne complet.

STEP 7 comprend des langages de programmation standard, ce qui s'avère très pratique et efficace pour la mise au point du programme de commande de votre application.

- CONT (schéma à contacts) est un langage de programmation graphique. Sa représentation se base sur des schémas (Page 165) de circuit.
- LOG (logigramme) est un langage de programmation se fondant sur les symboles logiques graphiques utilisés en algèbre (Page 166) booléenne.
- SCL (Structured Control Language) est un langage de programmation littéral évolué.

Lorsque vous créez un bloc de code, vous sélectionnez le langage de programmation à utiliser par ce bloc. Votre programme utilisateur peut utiliser des blocs de code créés dans n'importe lequel des langages de programmation disponibles.

---

### Remarque

STEP 7 est le composant logiciel de programmation et de configuration du portail TIA. Outre STEP 7, le portail TIA comprend le système WinCC permettant de concevoir et d'exécuter la visualisation runtime du process, et il contient l'aide en ligne pour WinCC et STEP 7.

---

## 2.1 Besoins du système

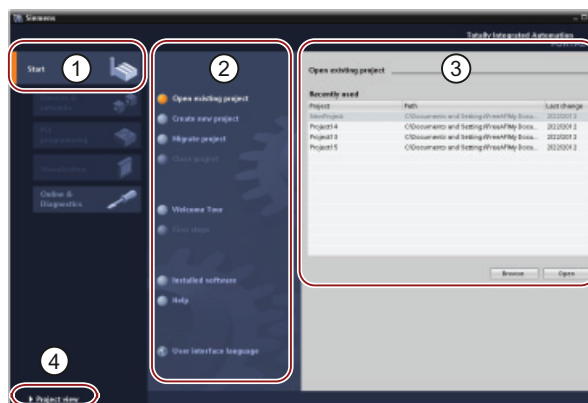
Vous devez ouvrir une session en tant qu'administrateur pour installer le logiciel STEP 7 sur un PC fonctionnant sous le système d'exploitation Windows XP ou Windows 7.

Tableau 2- 1 Besoins du système

Matériel / logiciel	Conditions requises
Type de processeur	Pentium M, 1,6 GHz ou similaire
Mémoire vive	1 Go
Espace disponible sur le disque dur	2 Go sur le lecteur système C:\
Systèmes d'exploitation	<ul style="list-style-type: none"><li>• Windows XP Professionnel SP3</li><li>• Windows 2003 Server R2 StdE SP2</li><li>• Windows 7 Home Premium (STEP 7 Basic uniquement, non pris en charge pour STEP 7 Professional)</li><li>• Windows 7 (Professional, Enterprise, Ultimate)</li><li>• Windows 2008 Server StdE R2</li></ul>
Carte graphique	32 Mo RAM Profondeur de couleur 24 bits
Résolution de l'écran	1024 x 768
Réseau	Ethernet 20 Mbits/s ou plus
Lecteur optique	DVD-ROM

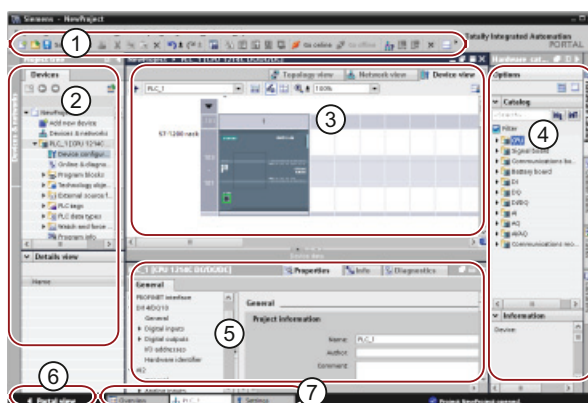
## 2.2 Différentes vues pour faciliter votre travail

STEP 7 offre un environnement convivial pour développer la logique du contrôleur, configurer la visualisation IHM et établir la communication réseau. Pour permettre d'augmenter votre productivité, STEP 7 offre deux vues différentes du projet : un ensemble orienté tâche de portails qui sont organisés selon la fonctionnalité des outils (vue du portail) et une vue orientée projet des éléments dans le projet (vue du projet). Choisissez la vue qui permet un travail le plus efficace possible. Avec un simple clic, vous pouvez faire le va-et-vient entre la vue du portail et la vue du projet.



Vue du portail

- ① Portails des différentes tâches
- ② Tâches du portail sélectionné
- ③ Panneau de sélection de l'action
- ④ Bascule dans la vue du projet



Vue du projet

- ① Menus et barre d'outils
- ② Navigateur du projet
- ③ Zone de travail
- ④ Task Cards
- ⑤ Fenêtre d'inspection
- ⑥ Bascule dans la vue du portail
- ⑦ Barre d'édition

Comme tous ces composants sont regroupés à un endroit, vous pouvez facilement accéder à chaque élément de votre projet. La fenêtre d'inspection montre, par exemple, les propriétés et informations de l'objet que vous avez sélectionné dans la zone de travail. Lorsque vous sélectionnez différents objets, la fenêtre d'inspection affiche les propriétés que vous pouvez configurer. La fenêtre d'inspection contient des onglets vous permettant de voir les informations de diagnostic et autres messages.

En affichant tous les éditeurs ouverts, la barre d'édition vous permet de travailler plus rapidement et efficacement. Pour basculer d'un éditeur ouvert à un autre, il suffit de cliquer sur l'éditeur correspondant. Vous pouvez également aligner deux éditeurs verticalement ou horizontalement. Cette fonction vous permet d'utiliser la fonction glisser-déplacer entre les éditeurs.

## 2.3 Simplicité des outils

### 2.3.1 Insertion d'instructions dans votre programme utilisateur

STEP 7 fournit des Task Cards qui contiennent les instructions pour votre programme. Les opérations sont regroupées selon leur fonction.

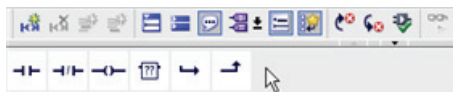


Pour créer votre programme, vous amenez les opérations de la Task Card dans un réseau.

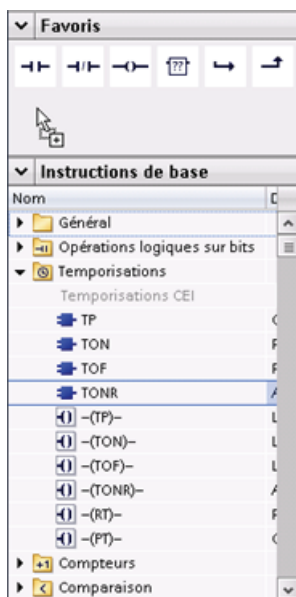


### 2.3.2 Accès aux instructions depuis la barre d'outils "Favoris"

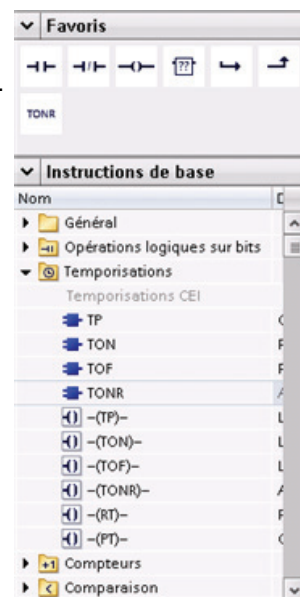
STEP 7 fournit une barre d'outils "Favoris" pour permettre un accès rapide aux instructions que vous utilisez fréquemment. Effectuez un clic simple sur l'icône de l'opération à insérer dans votre réseau !



(Pour les "Favoris" dans l'arborescence d'instructions, double-cliquez sur l'icône)



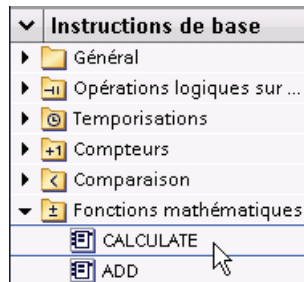
Vous pouvez facilement personnaliser les "Favoris" en ajoutant de nouvelles opérations. Il vous suffit de glisser-déplacer une opération sur vos "Favoris". Un simple clic permet alors d'accéder à l'opération !



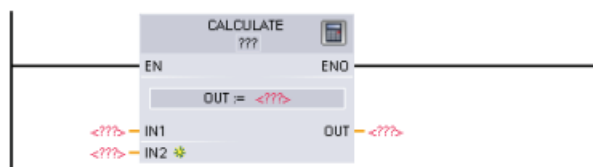


### 2.3.3 Création d'une équation complexe avec une instruction simple

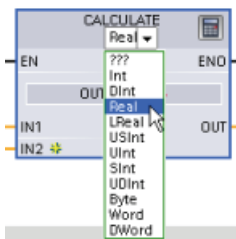
L'instruction Calculer vous permet de créer une fonction mathématique qui fonctionne sur des paramètres à entrées multiples pour produire le résultat, selon l'équation que vous définissez.



Dans l'arborescence d'opération Basic, agrandissez le fichier de fonctions Mathématiques. Double-cliquez sur l'opération Calculer pour insérer l'opération dans votre programme utilisateur.



L'opération Calculer non-configurée fournit deux paramètres d'entrée et un paramètre de sortie.

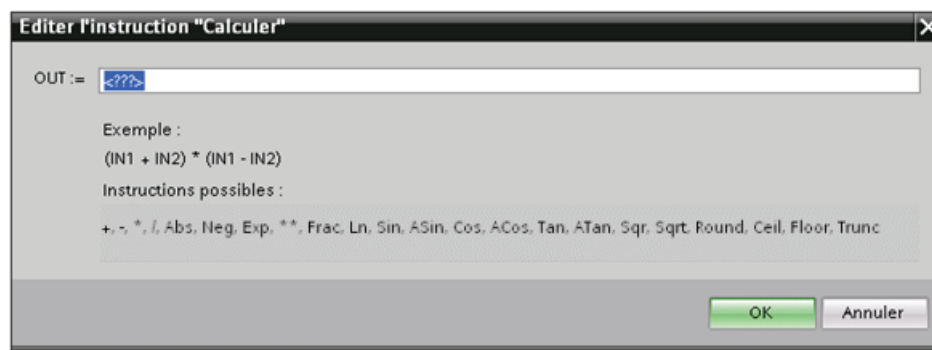


Cliquez sur "???" et sélectionnez les types de données pour les paramètres d'entrée et de sortie. (Les paramètres d'entrée et de sortie doivent tous être du même type de données.)

Pour cet exemple, sélectionnez le type de données "Réal".



Cliquez sur l'icône "Éditer équation" pour saisir l'équation.



Pour cet exemple, saisissez l'équation suivante pour établir une valeur brute analogique. (Les désignations "Entrée" et "Sortie" correspondent aux paramètres de l'opération Calculer.)

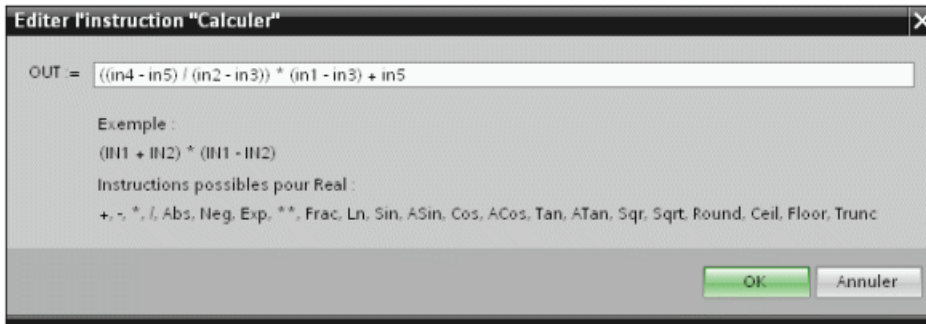
$$\text{Out}_{\text{value}} = ((\text{Out}_{\text{high}} - \text{Out}_{\text{low}}) / (\text{In}_{\text{high}} - \text{In}_{\text{low}})) * (\text{In}_{\text{value}} - \text{In}_{\text{low}}) + \text{Out}_{\text{low}}$$

$$\text{Out} = ((\text{in4} - \text{in5}) / (\text{in2} - \text{in3})) * (\text{in1} - \text{in3}) + \text{in5}$$

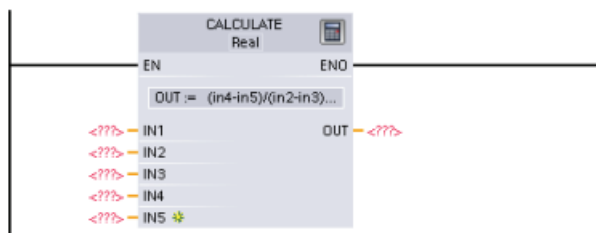
Si :	Out <sub>value</sub>	(Out)	Valeur de sortie graduée
	In <sub>value</sub>	(in1)	Valeur d'entrée analogique
	In <sub>high</sub>	(in2)	Limite supérieure pour la valeur d'entrée graduée
	In <sub>low</sub>	(in3)	Limite inférieure pour la valeur d'entrée graduée
	Out <sub>high</sub>	(in4)	Limite supérieure pour la valeur de sortie graduée
	Out <sub>low</sub>	(in5)	Limite inférieure pour la valeur de sortie graduée

Dans le champ "Éditer Calculer", saisissez l'équation avec les noms de paramètre :

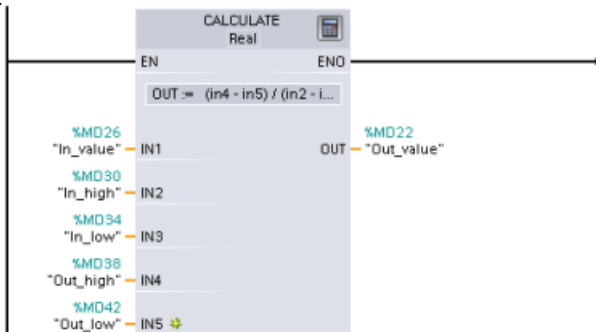
$$\text{OUT} = ((\text{in4} - \text{in5}) / (\text{in2} - \text{in3})) * (\text{in1} - \text{in3}) + \text{in5}$$



Lorsque vous cliquez sur "OK", l'opération Calculer crée les entrées nécessaires à l'opération.



Saisissez les noms de variables pour les valeurs qui correspondent aux paramètres.



### 2.3.4 Ajout d'entrées ou de sorties à une instruction CONT ou LOG

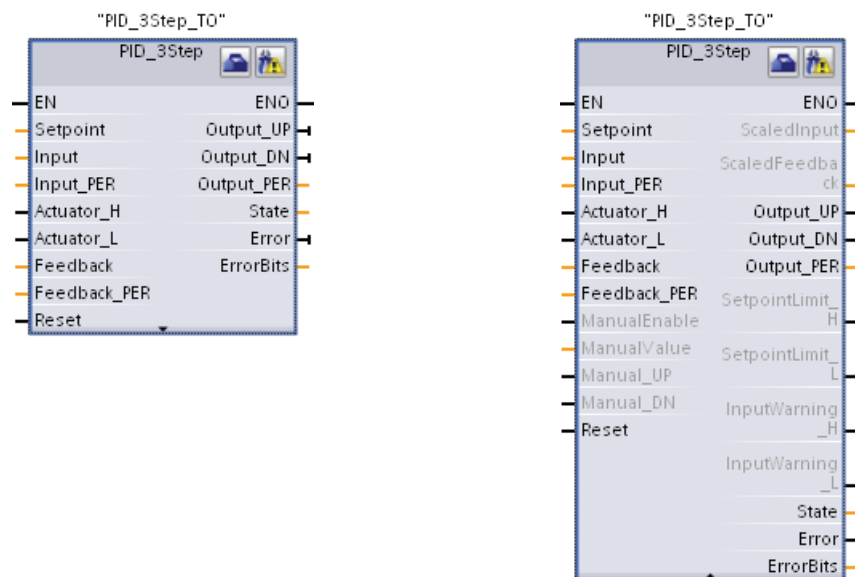


Certaines instructions vous permettent de créer des entrées ou sorties supplémentaires.

- Pour ajouter une entrée ou une sortie, cliquez sur l'icône de création ou cliquez avec le bouton droit de la souris sur la ligne de l'un des paramètres IN ou OUT existants et sélectionnez la commande "Insérer entrée".
- Pour supprimer une entrée ou une sortie, cliquez avec le bouton droit de la souris sur la ligne de l'un des paramètres IN ou OUT existants (lorsqu'il y a plus d'entrées que les deux entrées d'origine) et sélectionnez la commande "Supprimer".

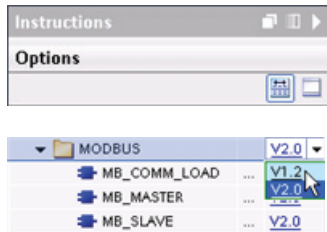
### 2.3.5 Instructions extensibles

Quelques-unes des instructions les plus complexes sont extensibles, n'affichant que les entrées et sorties importantes. Pour afficher toutes les entrées et sorties, cliquez sur la flèche au bas de l'instruction.



### 2.3.6 Sélection d'une version d'une instruction

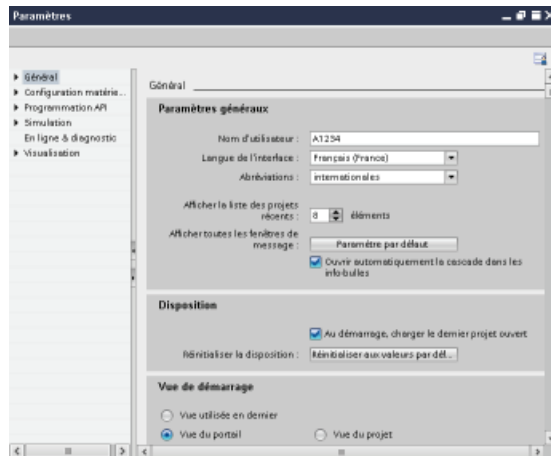
Les cycles de développement et de mise à disposition pour certains jeux d'instructions (tels que Modbus, PID et commande de mouvement) ont créé de multiples versions disponibles de ces instructions. Pour vous aider à garantir la compatibilité et la migration avec des projets plus anciens, STEP 7 vous permet de choisir la version d'une instruction à insérer dans votre programme utilisateur.



Dans la Task Card d'arborescence d'instructions, cliquez sur l'icône pour activer les en-têtes et colonnes de l'arborescence d'instructions.

Pour changer la version de l'instruction, sélectionnez la version appropriée dans la liste déroulante.

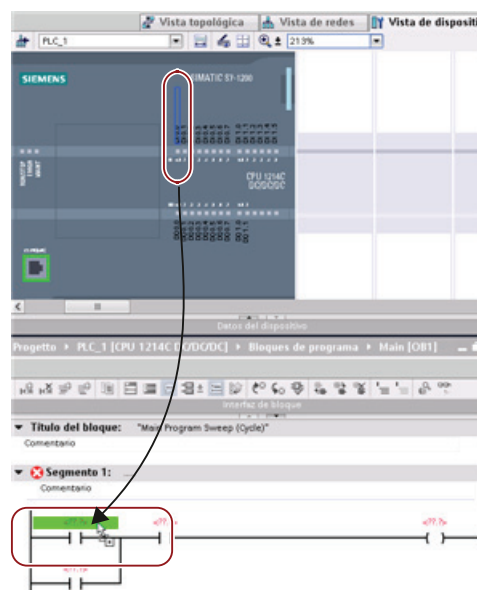
## 2.3.7 Modification de l'apparence et de la configuration de STEP 7



Vous pouvez sélectionner divers paramètres, tels que l'apparence de l'interface, la langue ou le dossier d'enregistrement de votre travail.

Pour modifier ces paramètres, sélectionnez la commande "Paramètres" dans le menu "Outils".

## 2.3.8 Glisser-déplacer entre les éditeurs

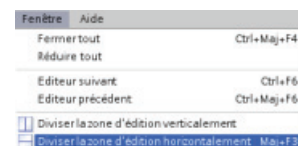


Pour vous aider à réaliser des tâches rapidement et facilement, STEP 7 vous permet de faire glisser des éléments d'un éditeur à un autre. Vous pouvez, par exemple, amener une entrée de la CPU sur l'adresse d'une opération dans votre programme utilisateur.

Vous devez effectuer un agrandissement d'au moins 200 % pour pouvoir sélectionner les entrées ou les sorties de la CPU.

Notez que les noms des variables sont affichés non seulement dans la table des variables de l'API, mais également dans la CPU.

Pour afficher simultanément deux éditeurs, choisissez les commandes de menu "Fractionner éditeur" ou sélectionnez les boutons correspondants dans la barre d'outils.



Pour basculer entre les éditeurs ouverts, cliquez sur les icônes correspondantes dans la barre d'édition.



### 2.3.9 Changement de l'état de fonctionnement de la CPU

Se reporter à

La CPU ne possède pas de commutateur physique pour changer l'état de fonctionnement (ARRET ou MARCHE).

Cliquez sur les boutons "Démarrer CPU" ou "Arrêter CPU" de la barre d'outils pour changer l'état de fonctionnement de la CPU.



Lorsque vous configurez la CPU dans les paramètres de configuration, vous configurez le comportement au démarrage dans les propriétés de la CPU.

Le portail "En ligne & diagnostic" comprend un panneau de commande pour modifier l'état de fonctionnement de la CPU en ligne. Pour pouvoir utiliser le panneau de commande de la CPU, vous devez être connecté en ligne à la CPU. La Task Card "Outils en ligne" affiche un panneau de commande indiquant l'état de fonctionnement de la CPU en ligne. Ce panneau de commande vous permet également de changer l'état de fonctionnement de la CPU en ligne.

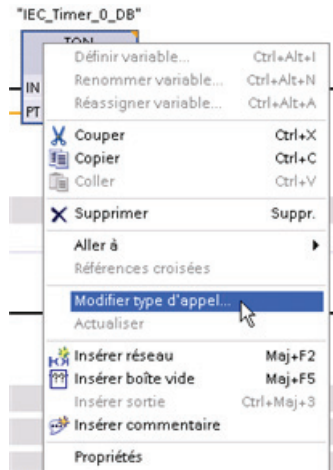


Servez-vous du bouton sur le panneau de commande pour changer l'état de fonctionnement (ARRET ou MARCHE). Le panneau de commande dispose également d'un bouton MRES pour réinitialiser la mémoire.

La couleur de l'indicateur MARCHE/ARRET signale l'état de fonctionnement en cours de la CPU. Le jaune correspond à l'état ARRET, le vert à l'état MARCHE.

Reportez-vous à Etats de fonctionnement de la CPU dans le Manuel système S7-1200 (Page 73) pour configurer l'état de fonctionnement par défaut à la mise sous tension.

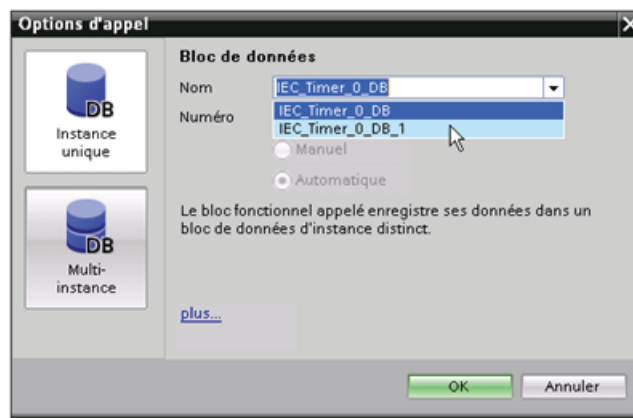
### 2.3.10 Modification du type d'appel pour un DB



STEP 7 permet de créer et de changer aisément l'association d'un DB à une instruction ou à un FB se trouvant dans un FB.

- Vous pouvez permuter l'association entre différents DB.
- Vous pouvez permuter l'association entre un DB mono-instance et un DB multi-instance.
- Vous pouvez créer un DB d'instance (s'il manque ou n'est pas disponible).

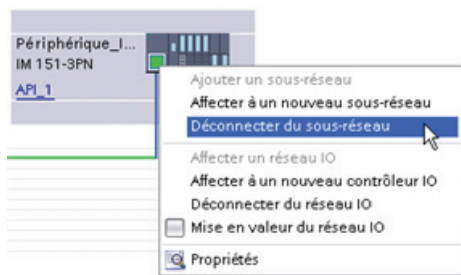
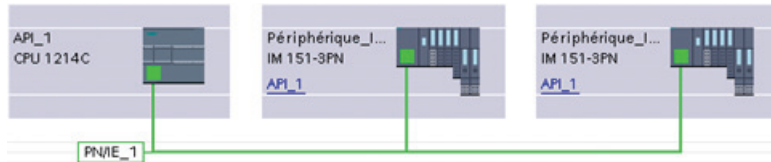
Vous pouvez accéder à la commande "Modifier type d'appel" en cliquant avec le bouton droit de la souris sur l'instruction ou le FB dans l'éditeur de programmes ou en sélectionnant la commande "Appel de bloc" dans le menu "Outils".



La boîte de dialogue "Options d'appel" vous permet de sélectionner un DB mono-instance ou un DB multi-instance. Vous pouvez également sélectionner des DB spécifiques dans une liste déroulante de DB disponibles.

### 2.3.11 Déconnexion temporaire d'appareils d'un réseau

Vous pouvez déconnecter des appareils individuels du sous-réseau. Comme la configuration de l'appareil n'est pas supprimée du projet, vous pouvez aisément restaurer la connexion à l'appareil.



Cliquez avec le bouton droit de la souris sur le port d'interface de l'appareil du réseau et sélectionnez la commande "Déconnecter du sous-réseau" dans le menu contextuel.

STEP 7 reconfigure les liaisons réseau, mais ne supprime pas l'appareil déconnecté du projet. Alors que la connexion réseau est supprimée, les adresses d'interface ne sont pas modifiées.



Vous devez faire passer la CPU à l'état ARRET lorsque vous chargez les nouvelles connexions réseau.

Pour reconnecter l'appareil, il vous suffit de créer une nouvelle connexion réseau au port de cet appareil.



### 2.3.12 Désenfichage virtuel d'appareils de la configuration



STEP 7 propose une zone de stockage pour les modules "retirés". Vous avez la possibilité d'amener un module du châssis afin d'en enregistrer la configuration. Ces modules retirés sont enregistrés dans votre projet, ce qui permettra de les réinsérer ultérieurement sans avoir à redéfinir les paramètres de configuration. L'une des applications de cette fonctionnalité est la maintenance temporaire. Considérez l'hypothèse où vous auriez à attendre avant de pouvoir remplacer un module et où vous décideriez d'utiliser temporairement un module différent comme solution à court terme. Vous auriez la possibilité d'amener le module configuré du châssis dans les "Modules non enfichés", puis d'insérer le module temporaire.



## Installation

### 3.1 Conseils pour l'installation d'appareils S7-1200

Le matériel S7-1200 est conçu pour être facile à installer. Vous pouvez monter l'automate S7-1200 sur un panneau ou sur un profilé support et l'orienter horizontalement ou verticalement. La petite taille du S7-1200 permet une optimisation de l'espace.

 **ATTENTION**

Les S7-1200 SIMATIC sont des automates de type ouvert. Il est nécessaire de les installer dans un boîtier, une armoire ou une salle électrique auxquels seules les personnes autorisées doivent avoir accès.

Le non-respect de ces règles d'installation peut entraîner la mort, des blessures graves et des dommages matériels importants.

Suivez toujours ces prescriptions lorsque vous installez des automates S7-1200.

#### **Ne placez pas les appareils S7-1200 à proximité de sources de chaleur, de haute tension et de bruit électrique**

En règle générale pour la mise en place des appareils de votre système, séparez toujours les appareils générant une tension élevée et un bruit électrique important des appareils de type logique basse tension, tels que le S7-1200.

Lors de l'implantation du S7-1200 dans votre panneau, tenez compte des appareils sources de chaleur et placez les appareils de type électronique dans les zones plus fraîches de votre armoire. En effet, réduire l'exposition aux températures élevées allonge la durée de vie des appareils électroniques.

Tenez également compte de l'acheminement du câblage pour les appareils dans le panneau. Évitez de placer les câbles de signaux et de communication de faible tension dans la même goulotte que le câblage d'alimentation en courant alternatif et le câblage pour courant continu à commutation rapide et haute énergie.



## 3.2 Bilan de consommation

Votre CPU possède une alimentation interne fournissant du courant à la CPU, aux modules d'entrées-sorties, au Signal Board et aux modules de communication ainsi qu'à d'autres équipements consommant du courant 24 V-.

Reportez-vous aux Caractéristiques techniques (Page 747) pour plus d'informations sur le bilan de consommation constitué par le courant 5 V- fourni par votre CPU et par les besoins en courant 5 V- des modules d'entrées-sorties, des Signal Boards et des modules de communication. Reportez-vous à "Calcul d'un bilan de consommation" (Page 883) pour déterminer combien de courant la CPU peut mettre à la disposition de votre configuration.

La CPU dispose également d'une alimentation de capteur 24 V- pouvant fournir du courant 24 V- aux entrées, aux bobines de relais sur les modules d'entrées-sorties ou à d'autres équipements. Si vos besoins en courant 24 V- dépassent ce que fournit l'alimentation de capteur, vous devez ajouter à votre système un module d'alimentation 24 V- externe. Vous trouverez dans les Caractéristiques techniques (Page 747) le bilan de consommation d'alimentation de capteur 24 V- pour votre CPU particulière.

---

### Remarque

Le CM 1243-5 (module maître PROFIBUS) doit être alimenté en courant par l'alimentation capteur 24 V- de la CPU.

---

Si vous avez besoin d'un module d'alimentation 24 V- externe, assurez-vous que ce module n'est pas connecté en parallèle avec l'alimentation de capteur de la CPU. En effet, il est recommandé, pour une meilleure protection contre les bruits électriques, que les communs (M) des différentes alimentations soient connectés.

 <b>ATTENTION</b>
------------------------------------------------------------------------------------------------------

Connecter une alimentation 24 V- externe en parallèle avec l'alimentation de capteur 24 V- peut entraîner un conflit entre les deux alimentations, chacune cherchant à établir son propre niveau de tension de sortie préféré.
--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

Ce conflit peut réduire la durée de vie ou provoquer une défaillance immédiate de l'une ou des deux alimentations, ayant pour effet un fonctionnement imprévisible du système d'automatisation pouvant entraîner la mort, des blessures graves et des dommages matériels importants.
--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

L'alimentation de capteur CC et toute alimentation externe doivent fournir du courant à des points différents.
----------------------------------------------------------------------------------------------------------------

Certains ports d'entrée d'alimentation 24 V- dans le système S7-1200 sont interconnectés, avec un circuit logique commun connectant plusieurs bornes M. Par exemple, les circuits suivants sont interconnectés lorsqu'ils sont signalés comme "non isolés" dans les fiches techniques : l'alimentation 24 V- de la CPU, l'entrée d'alimentation pour la bobine de relais d'un SM ou l'alimentation pour une entrée analogique non isolée. Toutes les bornes M non isolées doivent être connectées au même potentiel de référence externe.



**ATTENTION**

Connecter des bornes M non isolées à des potentiels de référence différents provoque des flux de courant indésirables qui peuvent être à l'origine de dégâts ou d'un fonctionnement imprévisible dans l'automate et tout équipement connecté.

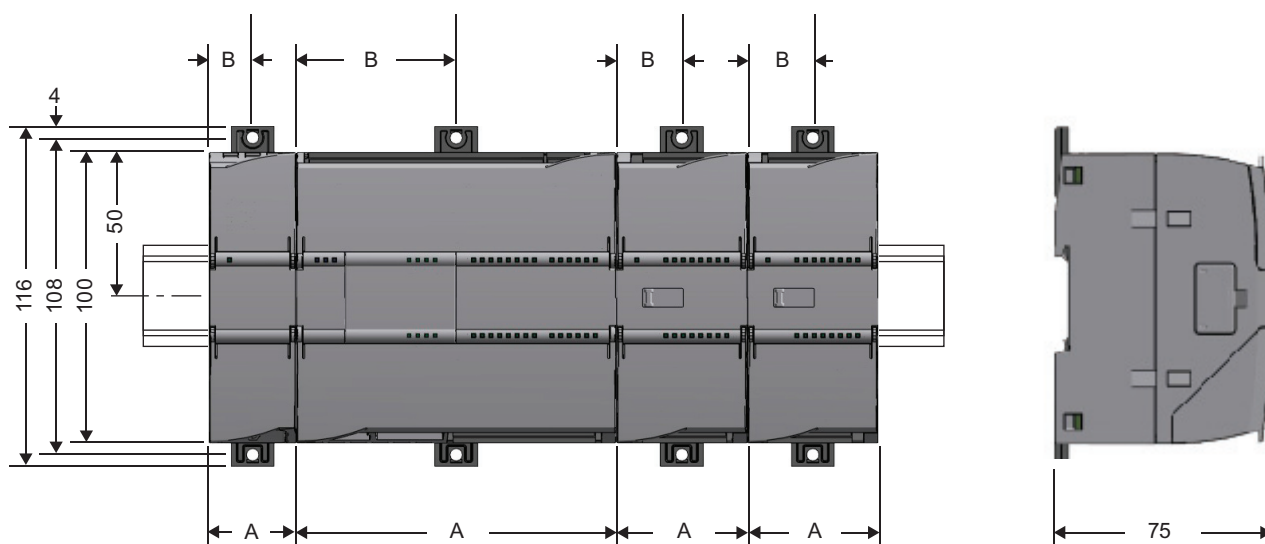
Le non-respect de ces conseils peut être à l'origine de dégâts ou d'un fonctionnement imprévisible pouvant entraîner la mort, des blessures graves et des dommages matériels importants.

Vérifiez toujours que toutes les bornes M non isolées dans un système S7-1200 sont connectées au même potentiel de référence.

### 3.3 Procédures d'installation et de désinstallation

#### 3.3.1 Dimensions de montage pour les appareils S7-1200

CPU 1211C, CPU 1212C, CPU 1214C



CPU 1215C

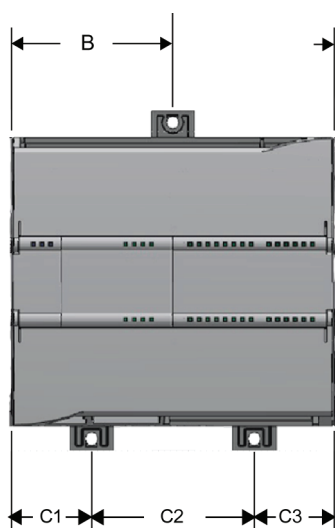


Tableau 3- 1 Dimensions de montage (mm)

Appareils S7-1200		Largeur A (mm)	Largeur B (mm)	Largeur C (mm)
CPU	CPU 1211C et CPU 1212C	90	45	--
	CPU 1214C	110	55	--
	CPU 1215C	130	65 (haut)	Bas : C1 : 32.5 C2 : 65 C3 : 32.5
Modules d'entrées-sorties	TOR, 8 et 16 entrées/sorties Analogiques, 2, 4 et 8 entrées/sorties Thermocouple, 4 et 8 entrées/sorties RTD, 4 entrées/sorties	45	22.5	--
	TOR, 8 sorties Relais (inverseur)	70	22.5	--
	Analogiques, 16 entrées/sorties RTD, 8 entrées/sorties	70	35	--
Interfaces de communication	CM 1241 RS232 et CM 1241 RS422/485 CM 1243-5 maître PROFIBUS et CM 1242-5 esclave PROFIBUS CM 1242-2 maître AS-i CP 1242-7 GPRS	30	15	--
	Adaptateur TS Adapter IE Basic	60 <sup>1</sup>	15	--

<sup>1</sup> Comme vous devez installer un adaptateur TS modulaire avec l'adaptateur TS, la largeur totale ("largeur A") est de 60 mm.

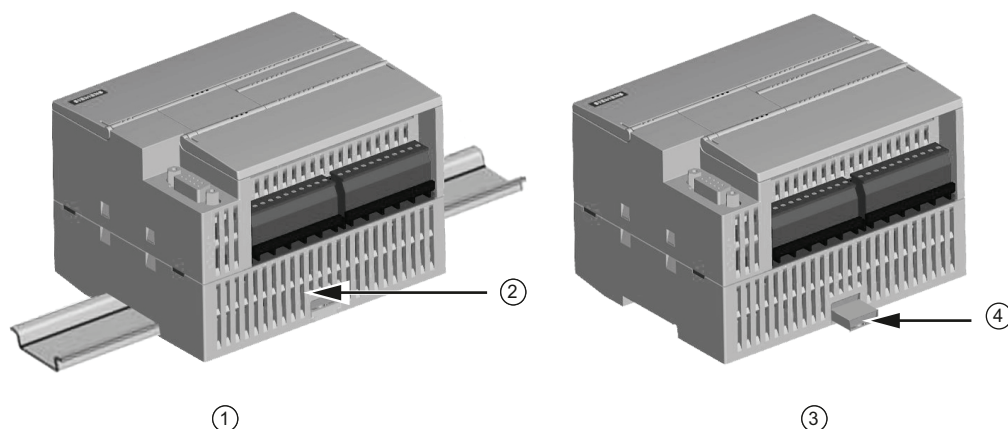
Chaque CPU, SM, CM et CP peut être indifféremment monté sur un profilé support ou encastré dans un panneau. Servez-vous des barrettes de fixation sur le module pour fixer l'appareil sur le profilé support. Ces barrettes s'encliquettent également en position sortie afin de fournir des points de vissage pour monter l'unité directement sur un panneau. Le diamètre intérieur de l'alésage pour les barrettes de fixation de l'appareil est de 4,3 mm.

Une zone thermique de 25 mm doit également être ménagée au-dessus et en dessous de l'unité pour la libre circulation de l'air.



## Installation et désinstallation des appareils S7-1200

Il est très aisé d'installer la CPU sur un profilé support ou sur un panneau. Des barrettes de fixation sont fournies pour fixer l'appareil sur le profilé support. Ces barrettes s'encliquettent également en position sortie afin de fournir un point de vissage pour le montage sur panneau de l'unité.



- |                                            |                                                           |
|--------------------------------------------|-----------------------------------------------------------|
| ① Montage sur profilé support              | ③ Montage sur panneau                                     |
| ② Barrette de fixation en position rentrée | ④ Barrette en position sortie pour le montage sur panneau |

Assurez-vous, avant d'installer ou de démonter tout appareil électrique, que cet appareil a été mis hors tension. Assurez-vous également que tout équipement associé a été mis hors tension.

### ATTENTION

L'installation ou le démontage du S7-1200 ou d'équipements reliés alors qu'ils sont sous tension peut provoquer un choc électrique ou un fonctionnement imprévisible du matériel.

La non-désactivation de l'alimentation complète du S7-1200 et des appareils qui y sont reliés pendant les procédures d'installation ou de désinstallation peut entraîner la mort, des blessures graves ou des dommages matériels importants en raison du choc électrique ou du fonctionnement imprévisible du matériel.

Prenez toujours toutes les mesures de sécurité nécessaires et assurez-vous que l'alimentation de l'automate S7-1200 est coupée avant de tenter d'installer ou de démonter des CPU S7-1200 ou des matériels reliés.

Assurez-vous toujours, lorsque vous remplacez ou installez un appareil S7-1200, que vous utilisez le module correct ou un appareil équivalent.

 **ATTENTION**

Une installation incorrecte d'un module S7-1200 peut entraîner un fonctionnement imprévisible du programme dans le S7-1200.

Le remplacement d'un appareil S7-1200 par un autre modèle, sans respecter l'orientation correcte et l'ordre initial, peut entraîner la mort, des blessures graves ou des dommages matériels importants en raison d'un fonctionnement imprévisible de l'équipement.

Remplacez toujours un appareil S7-1200 par le même modèle et assurez-vous que vous l'orientez et le positionnez correctement.

 **ATTENTION**

Ne déconnectez pas les équipements en cas d'atmosphère inflammable ou combustible.

En effet, la déconnexion d'équipements en présence d'une atmosphère inflammable ou combustible peut provoquer un incendie ou une explosion, pouvant entraîner la mort, des blessures graves et des dommages matériels importants.

**PRUDENCE**

Des décharges électrostatiques peuvent endommager l'appareil ou son emplacement sur la CPU.

Pour éviter tout risque lorsque vous manipulez l'appareil, soyez en contact avec un tapis conducteur mis à la terre ou portez un bracelet spécial avec chaînette.

### 3.3.2 Installation et désinstallation de la CPU

Vous pouvez monter la CPU sur un panneau ou sur un profilé support.

**Remarque**

Reliez les modules de communication éventuels à la CPU et installez l'ensemble en tant qu'unité. Installez les modules d'entrées-sorties séparément une fois la CPU installée.

Tenez compte des points suivants lorsque vous installez les unités sur le profilé support ou sur un panneau :

- En cas de montage sur profilé support, assurez-vous que la barrette de fixation supérieure est en position rentrée et que la barrette inférieure est en position sortie sur la CPU et les CM reliés.
- Une fois les appareils installés sur le profilé support, faites glisser les barrettes de fixation inférieure dans leur position rentrée pour verrouiller les appareils sur le profilé support.
- En cas de montage sur panneau, assurez-vous que les barrettes de fixation sont en position sortie.

Procédez comme suit pour installer la CPU sur un panneau :

1. Positionnez, percez et taraudez les trous de fixation (M4) en vous aidant des dimensions indiquées dans le tableau Dimensions de montage (mm) (Page 47).
2. Vérifiez que la CPU et tout le matériel S7-1200 sont hors tension.
3. Sortez les barrettes de fixation du module. Assurez-vous que les barrettes de fixation en haut et en bas de la CPU sont bien en position sortie.
4. Fixez le module sur le panneau à l'aide d'une vis M4 à tête cylindrique avec rondelle plate élastique. N'utilisez pas de vis à tête plate.

---

**Remarque**

Vous déterminerez le type de vis en fonction du matériau sur lequel vous la fixerez. Vous devez appliquer un couple approprié jusqu'à ce que la rondelle élastique soit aplatie. Veillez à ne pas appliquer un couple excessif aux vis de montage. N'utilisez pas de vis à tête plate.

---

**Remarque**

Si votre système se situe dans un environnement à fortes vibrations ou est disposé verticalement, le montage du S7-1200 sur panneau offrira un niveau de protection plus élevé.

---

3.3 Procédures d'installation et de désinstallation

Tableau 3- 2 Installation de la CPU sur un profilé support

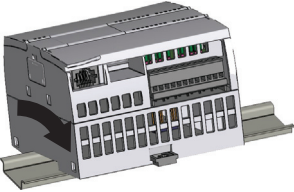
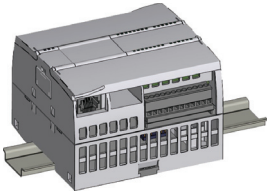
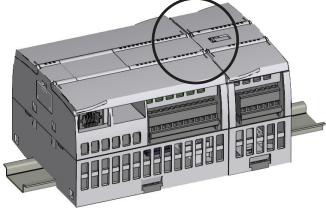
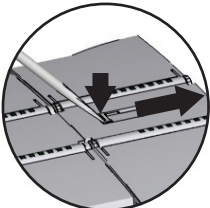
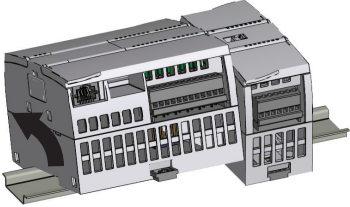
Tâche	Marche à suivre
	<ol style="list-style-type: none"> <li>1. Installez le profilé support. Vissez le profilé support tous les 75 mm sur le panneau de montage.</li> <li>2. Vérifiez que la CPU et tout le matériel S7-1200 sont hors tension.</li> <li>3. Accrochez la CPU sur le haut du profilé support.</li> <li>4. Tirez la barrette de fixation située au bas de la CPU pour que la CPU puisse s'enclencher sur le profilé support.</li> </ol>
	<ol style="list-style-type: none"> <li>5. Faites pivoter la CPU vers le bas sur le profilé support.</li> <li>6. Repoussez les barrettes de fixation pour verrouiller la CPU sur le profilé support.</li> </ol>

Tableau 3- 3 Désinstallation de la CPU d'un profilé support

Tâche	Marche à suivre
	 <ol style="list-style-type: none"> <li>1. Vérifiez que la CPU et tout le matériel S7-1200 sont hors tension.</li> <li>2. Déconnectez les connecteurs d'E/S, les fils et les câbles de la CPU (Page 58).</li> <li>3. Démontez la CPU et tout module de communication relié en tant qu'unité. Tous les modules d'entrées-sorties doivent rester en place.</li> </ol>
	<ol style="list-style-type: none"> <li>4. Si un module SM est connecté à la CPU, rentrez le connecteur de bus :             <ul style="list-style-type: none"> <li>- Placez un tournevis à côté de la languette sur le haut du module d'entrées-sorties.</li> <li>- Appuyez pour dégager le connecteur de la CPU.</li> <li>- Faites glisser la languette complètement à droite.</li> </ul> </li> <li>5. Démontez la CPU :             <ul style="list-style-type: none"> <li>- Tirez la barrette de fixation pour libérer la CPU du profilé support.</li> <li>- Faites pivoter la CPU vers le haut pour la dégager du profilé support et retirez-la du système.</li> </ul> </li> </ol>

### 3.3.3 Installation et désinstallation d'un SB, d'un CB ou d'un BB

Tableau 3- 4 Installation d'un SB, d'un CB ou d'un BB 1297

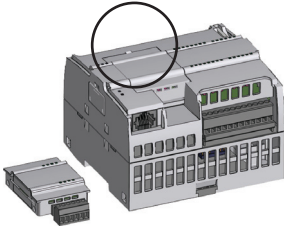
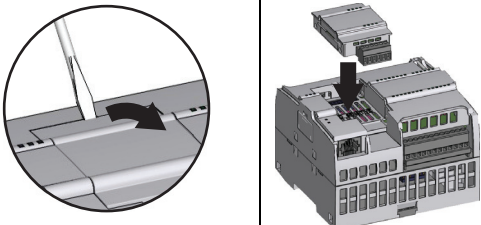
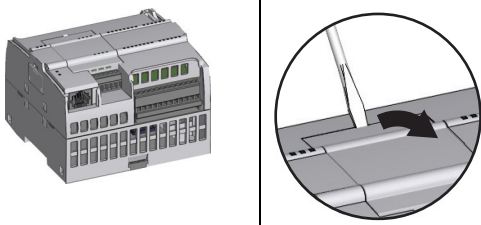
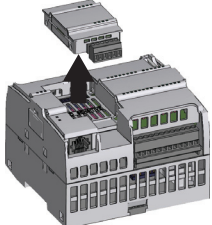
Tâche	Marche à suivre
	<ol style="list-style-type: none"> <li>1. Vérifiez que la CPU et tout le matériel S7-1200 sont hors tension.</li> <li>2. Retirez les caches des borniers supérieur et inférieur de la CPU.</li> <li>3. Placez un tournevis dans l'encoche à l'arrière du cache sur le haut de la CPU.</li> <li>4. Faites doucement levier vers le haut et retirez le cache de la CPU.</li> <li>5. Insérez le module verticalement dans sa position de montage dans la partie supérieure de la CPU.</li> </ol>
	<ol style="list-style-type: none"> <li>6. Appuyez fermement sur le module jusqu'à ce qu'il s'enclenche en position.</li> <li>7. Remplacez les caches des borniers.</li> </ol>

Tableau 3- 5 Désinstallation d'un SB, d'un CB ou d'un BB 1297

Tâche	Marche à suivre
	<ol style="list-style-type: none"> <li>1. Vérifiez que la CPU et tout le matériel S7-1200 sont hors tension.</li> <li>2. Retirez les caches des borniers supérieur et inférieur de la CPU.</li> <li>3. Placez un tournevis dans l'encoche sur le haut du module.</li> <li>4. Faites doucement levier pour dégager le module de la CPU.</li> <li>5. Retirez le module verticalement de sa position de montage dans la partie supérieure de la CPU.</li> <li>6. Remplacez le cache sur la CPU.</li> </ol>
	<ol style="list-style-type: none"> <li>7. Remplacez les caches des borniers.</li> </ol>

### **Installation ou remplacement de la pile dans le Battery Board BB 1297**

Le BB 1297 fonctionne avec une pile de type CR1025. La pile n'est pas fournie avec le BB 1297 et doit être achetée par l'utilisateur. Procédez comme suit pour installer ou remplacer la pile :

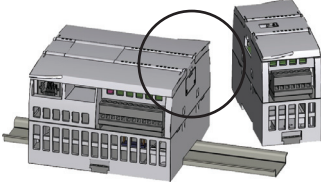
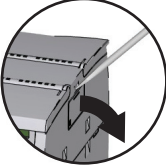
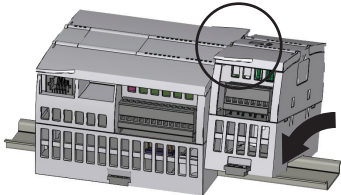
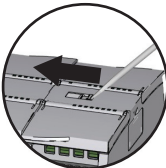
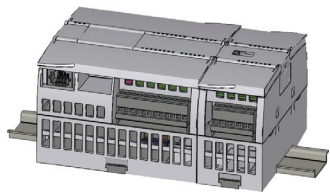
1. Placez une nouvelle pile dans le BB 1297, avec le pôle positif de la pile vers le haut et le pôle négatif à côté de la carte de câblage imprimée.
2. Le BB 1297 est prêt à être installé dans la CPU. Suivez les instructions d'installation données plus haut pour installer le BB 1297.

Procédez comme suit pour remplacer la pile dans le BB 1297 :

1. Retirez le BB 1297 de la CPU en suivant les instructions de démontage données plus haut.
2. Retirez l'ancienne pile avec précaution à l'aide d'un petit tournevis. Extrayez la pile de dessous l'attache.
3. Installez une nouvelle pile de remplacement CR1025 avec le pôle positif de la pile vers le haut et le pôle négatif à côté de la carte de câblage imprimée.
4. Réinstallez le Battery Board BB 1297 en suivant les instructions d'installation données plus haut.

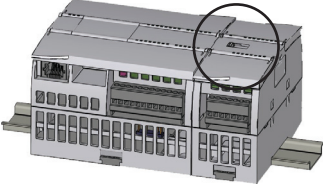
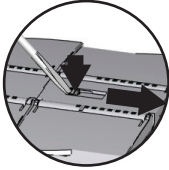
### 3.3.4 Installation et désinstallation d'un SM

Tableau 3- 6 Installation d'un SM

Tâche	Marche à suivre	
		<p>Installez votre module d'entrées-sorties après avoir installé la CPU.</p> <ol style="list-style-type: none"> <li>1. Vérifiez que la CPU et tout le matériel S7-1200 sont hors tension.</li> <li>2. Retirez le cache pour le connecteur du côté droit de la CPU :</li> <li>3. Insérez un tournevis dans l'encoche au-dessus du cache.</li> <li>4. Détachez doucement le côté supérieur du cache puis retirez le cache. Conservez-le en vue de sa réutilisation.</li> </ol>
		<p>Raccordez le SM à la CPU :</p> <ol style="list-style-type: none"> <li>1. Positionnez le SM à côté de la CPU.</li> <li>2. Accrochez le SM sur le haut du profilé support.</li> <li>3. Tirez la barrette de fixation inférieure pour que le SM puisse s'enclencher sur le profilé support.</li> <li>4. Faites pivoter le SM vers le bas à côté de la CPU et repoussez la barrette de fixation inférieure pour verrouiller le SM sur le profilé support.</li> </ol>
	<p>L'insertion du connecteur de bus réalise les connexions mécaniques et électriques pour le SM.</p> <ol style="list-style-type: none"> <li>1. Placez un tournevis à côté de la languette sur le haut du module d'entrées-sorties.</li> <li>2. Faites glisser la languette complètement vers la gauche pour insérer le connecteur de bus dans la CPU.</li> </ol> <p>Procédez de la même façon pour installer un module d'entrées-sorties sur un autre module d'entrées-sorties.</p>	

3.3 Procédures d'installation et de désinstallation

Tableau 3- 7 Désinstallation d'un SM

Tâche	Marche à suivre
	<p>Vous pouvez démonter n'importe quel SM sans démonter la CPU ni d'autres SM en place.</p> <ol style="list-style-type: none"> <li>1. Vérifiez que la CPU et tout le matériel S7-1200 sont hors tension.</li> <li>2. Retirez les connecteurs d'E/S et le câblage du SM (Page 58).</li> <li>3. Retirez le connecteur de bus :             <ul style="list-style-type: none"> <li>- Placez un tournevis à côté de la languette sur le haut du module d'entrées-sorties.</li> <li>- Appuyez pour dégager le connecteur de la CPU.</li> <li>- Faites glisser la languette complètement à droite.</li> </ul> </li> </ol> <p>S'il y a un autre SM à droite, répétez cette procédure pour ce SM.</p>
	<p>Démontez le SM :</p> <ol style="list-style-type: none"> <li>1. Tirez la barrette de fixation inférieure pour libérer le SM du profilé support.</li> <li>2. Faites pivoter le SM vers le haut pour le dégager du profilé support. Retirez le SM du système.</li> <li>3. Si nécessaire, couvrez le connecteur de bus sur la CPU de son cache pour éviter la contamination.</li> </ol> <p>Procédez de la même façon pour démonter un module d'entrées-sorties d'un autre module d'entrées-sorties.</p>



### 3.3.5 Installation et désinstallation d'un CM ou d'un CP

Reliez les modules de communication éventuels à la CPU et installez l'ensemble en tant qu'unité, comme illustré au paragraphe Installation et désinstallation de la CPU (Page 50).

Tableau 3- 8 Installation d'un CM ou d'un CP

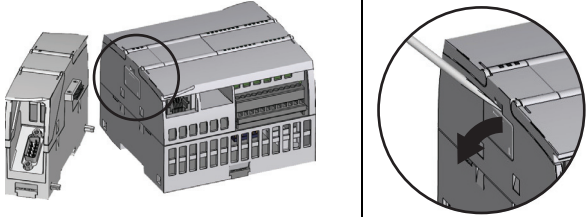
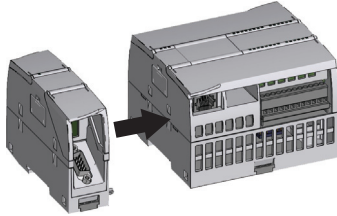
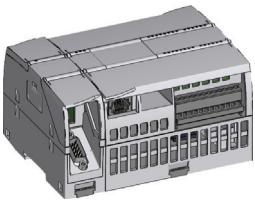
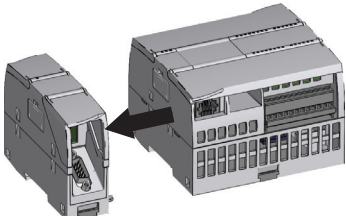
Tâche	Marche à suivre
	<ol style="list-style-type: none"> <li>1. Vérifiez que la CPU et tout le matériel S7-1200 sont hors tension.</li> <li>2. Reliez le CM à la CPU avant d'installer l'ensemble en tant qu'unité sur le profilé support ou le panneau.</li> <li>3. Retirez le cache de bus sur le côté gauche de la CPU : <ul style="list-style-type: none"> <li>– Insérez un tournevis dans l'encoche au-dessus du cache de bus.</li> <li>– Détachez doucement le côté supérieur du cache.</li> </ul> </li> </ol>
	<ol style="list-style-type: none"> <li>4. Retirez le cache de bus. Conservez-le en vue de sa réutilisation.</li> <li>5. Raccordez le CM ou le CP à la CPU : <ul style="list-style-type: none"> <li>– Alignez le connecteur de bus et les montants du CM avec les trous de la CPU.</li> <li>– Appuyez fermement les unités l'une contre l'autre jusqu'à ce que les montants s'enclenchent.</li> </ul> </li> <li>6. Installez la CPU et le CP sur un profilé support ou dans un panneau.</li> </ol>

Tableau 3- 9 Désinstallation d'un CM ou d'un CP

Tâche	Marche à suivre
	<p>Démontez la CPU et le CM du profilé support ou du panneau en tant qu'unité.</p> <ol style="list-style-type: none"> <li>1. Vérifiez que la CPU et tout le matériel S7-1200 sont hors tension.</li> <li>2. Retirez les connecteurs d'E/S et tous les fils et câbles de la CPU et des CM.</li> <li>3. En cas de montage sur profilé support, tirez les barrettes de fixation inférieures sur la CPU et les CM en position sortie.</li> <li>4. Démontez la CPU et les CM du profilé support ou du panneau.</li> <li>5. Maintenez fermement la CPU et le CM et tirez pour les séparer.</li> </ol>
	

**PRUDENCE**

N'utilisez pas d'outil pour séparer les modules, car cela risquerait de les endommager.

### 3.3.6 Démontage et remontage du bornier de connexion S7-1200

Les CPU, les SB et les SM comportent des connecteurs amovibles pour faciliter le câblage.

Tableau 3- 10 Démontage du bornier de connexion

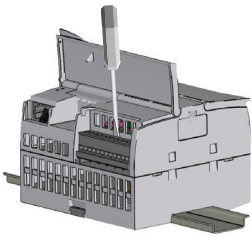
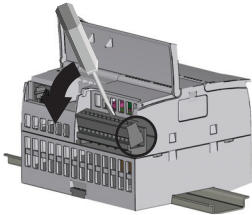
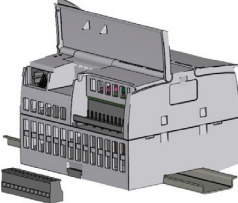
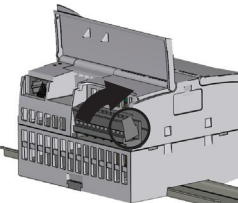
Tâche	Marche à suivre
	<p>Préparez le système au démontage du bornier de connexion en mettant la CPU hors tension et en ouvrant le cache au-dessus du bornier de connexion.</p> <ol style="list-style-type: none"><li>1. Vérifiez que la CPU et tout le matériel S7-1200 sont hors tension.</li><li>2. Inspectez le dessus du connecteur afin de localiser l'encoche pour la pointe du tournevis.</li><li>3. Insérez un tournevis dans l'encoche.</li><li>4. Faites doucement levier sur le haut du connecteur pour le détacher de la CPU. Le connecteur se dégagera avec un bruit sec.</li><li>5. Saisissez le connecteur et retirez-le de la CPU.</li></ol>
	

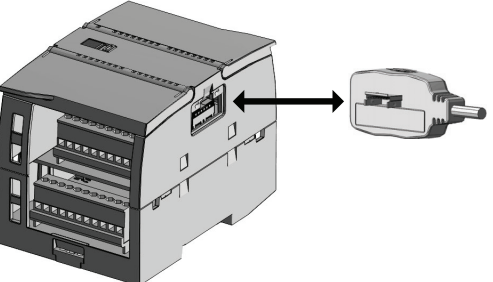
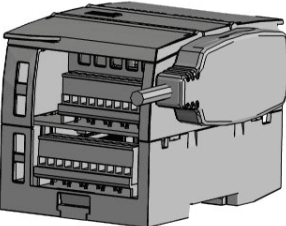
Tableau 3- 11 Installation du bornier de connexion

Tâche	Marche à suivre
	<p>Préparez les composants en vue de l'installation du bornier de connexion en mettant la CPU hors tension et en ouvrant le cache pour le bornier.</p> <ol style="list-style-type: none"> <li>1. Vérifiez que la CPU et tout le matériel S7-1200 sont hors tension.</li> <li>2. Alignez le connecteur avec les broches sur l'unité.</li> <li>3. Alignez la face de câblage du connecteur à l'intérieur du bord du socle de connecteur.</li> <li>4. Appuyez fermement et faites pivoter le connecteur jusqu'à ce qu'il s'enclenche.</li> </ol> <p>Vérifiez soigneusement que le connecteur est correctement aligné et bien enfoncé.</p>
	

### 3.3.7 Installation et désinstallation du câble d'extension

Le câble d'extension S7-1200 offre une souplesse accrue lors de la configuration de la disposition de votre système S7-1200. Un seul câble d'extension est autorisé par système CPU. Vous installez le câble d'extension soit entre la CPU et le premier SM, soit entre deux SM quelconques.

Tableau 3- 12 Installation et désinstallation du connecteur mâle du câble d'extension

Tâche	Marche à suivre
	<p>Pour installer le connecteur mâle :</p> <ol style="list-style-type: none"> <li>1. Vérifiez que la CPU et tout le matériel S7-1200 sont hors tension.</li> <li>2. Enfoncez le connecteur dans le connecteur de bus du côté droit du module d'entrées-sorties ou de la CPU.</li> </ol> <p>Pour désinstaller le connecteur mâle :</p> <ol style="list-style-type: none"> <li>1. Vérifiez que la CPU et tout le matériel S7-1200 sont hors tension.</li> <li>2. Tirez sur le connecteur mâle pour l'extraire du module d'entrées-sorties ou de la CPU.</li> </ol>
	

3.3 Procédures d'installation et de désinstallation

Tableau 3- 13 Installation du connecteur femelle du câble d'extension

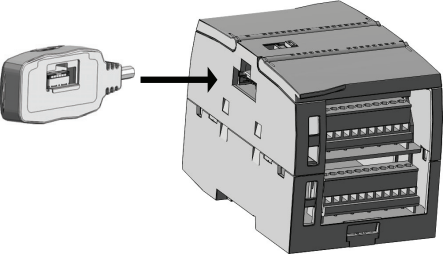
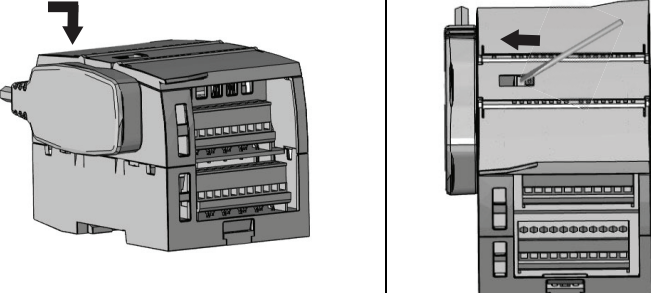
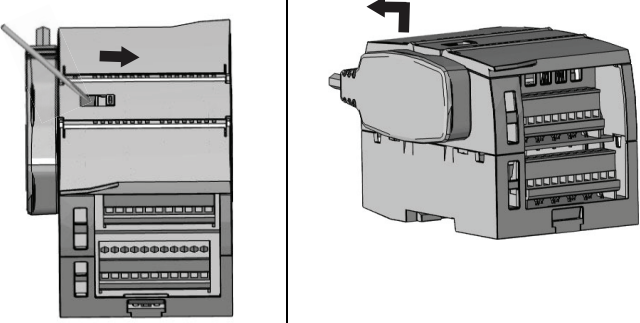
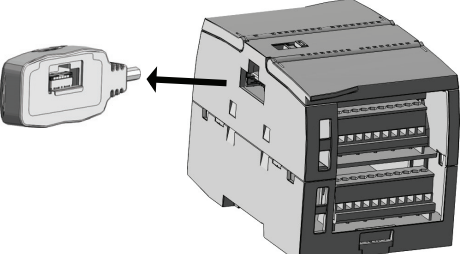
Tâche	Marche à suivre
	<ol style="list-style-type: none"> <li>1. Vérifiez que la CPU et tout le matériel S7-1200 sont hors tension.</li> <li>2. Placez le connecteur femelle sur le connecteur de bus du côté gauche du module d'entrées-sorties.</li> <li>3. Faites glisser l'extension en crochet du connecteur femelle dans le boîtier au niveau du connecteur de bus et appuyez légèrement vers le bas pour enclencher le crochet.</li> </ol>
	<ol style="list-style-type: none"> <li>4. Fixez le connecteur en place : <ul style="list-style-type: none"> <li>- Placez un tournevis à côté de la languette sur le haut du module d'entrées-sorties.</li> <li>- Faites glisser la languette complètement vers la gauche.</li> </ul> </li> </ol> <p>Pour enclencher le connecteur, vous devez faire glisser la languette jusqu'au bout vers la gauche. Le connecteur doit être verrouillé en place.</p>

Tableau 3- 14 Désinstallation du connecteur femelle du câble d'extension

Tâche	Marche à suivre
	<ol style="list-style-type: none"> <li>1. Vérifiez que la CPU et tout le matériel S7-1200 sont hors tension.</li> <li>2. Déverrouillez le connecteur : <ul style="list-style-type: none"> <li>- Placez un tournevis à côté de la languette sur le haut du module d'entrées-sorties.</li> <li>- Appuyez légèrement et faites glisser la languette complètement vers la droite.</li> </ul> </li> <li>3. Soulevez légèrement le connecteur pour désenclencher l'extension en crochet.</li> <li>4. Retirez le connecteur femelle.</li> </ol>
	

### 3.3.8 Adaptateur TS Adapter (TeleService)

#### 3.3.8.1 Connexion de l'adaptateur TeleService

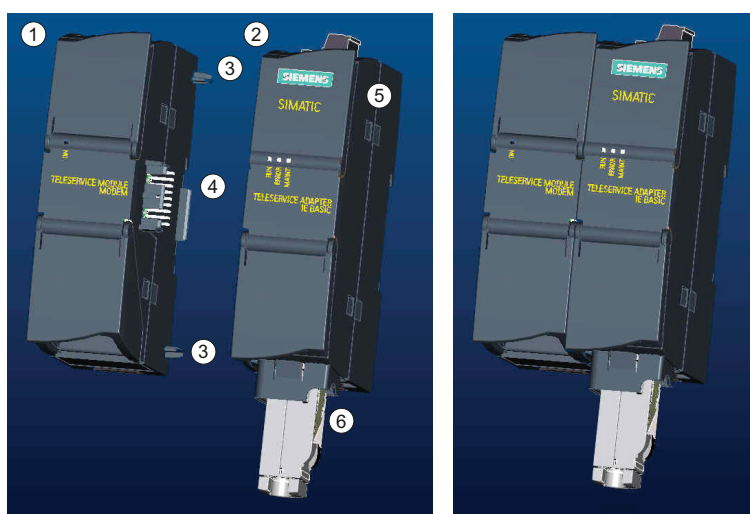
Avant d'installer l'adaptateur TS (TeleService) Adapter IE Basic, vous devez connecter l'adaptateur TS Adapter et un module TS.

Modules TS disponibles :

- Module TS RS232
- Module TS Modem
- Module TS GSM
- Module TS RNIS

#### PRUDENCE

Le module TS peut être endommagé si vous touchez les contacts de son connecteur enfichable ④. Suivez les directives pour composants CSDE afin d'éviter que le module TS ne soit endommagé par des décharges électrostatiques. Avant de connecter un module TS et l'adaptateur TS, assurez-vous que tous deux sont inactifs.



- |   |                       |   |                                    |
|---|-----------------------|---|------------------------------------|
| ① | Module TS             | ④ | Connecteur enfichable du module TS |
| ② | Adaptateur TS Adapter | ⑤ | Ouverture impossible               |
| ③ | Éléments              | ⑥ | Port Ethernet                      |

**PRUDENCE**

Avant de connecter un module TS et l'unité de base TS Adapter, assurez-vous que les broches de contact ④ ne dévient pas. Lors de la connexion, veillez à ce que le connecteur mâle et les éléments de guidage soient positionnés correctement.

Connectez uniquement un module TS à l'adaptateur TS Adapter. N'essayez pas de connecter par la force l'adaptateur TS Adapter à un autre dispositif, tel qu'une CPU S7-1200. Ne modifiez pas la structure mécanique du connecteur et n'ôtez et n'endommagez pas les éléments de guidage.

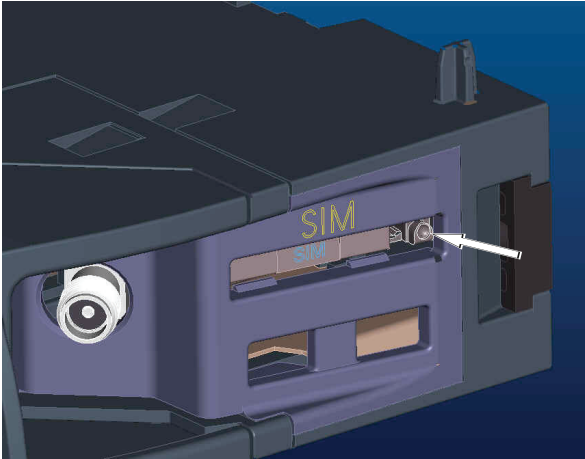
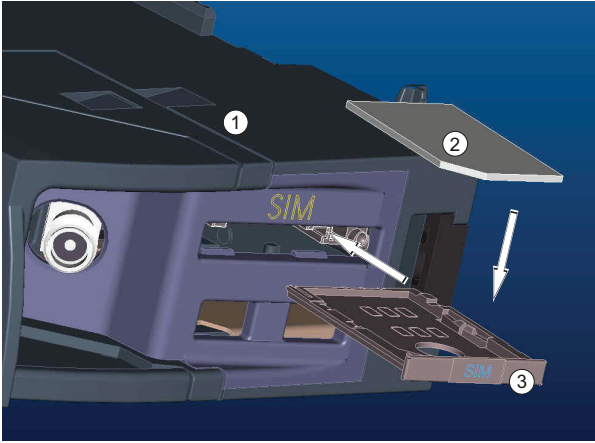
**3.3.8.2 Installation de la carte SIM**

Localisez l'emplacement de la carte SIM sur la face inférieure du TS module GSM.

**IMPORTANT**

La carte SIM doit uniquement être débrochée ou enfichée lorsque le TS module GSM est hors tension.

Tableau 3- 15 Installation de la carte SIM

Marche à suivre	Tâche							
<p>Utilisez un objet pointu pour enfoncer le bouton d'éjection du porte-carte SIM (dans le sens de la flèche) et retirez le porte-carte SIM.</p>								
<p>Placez la carte SIM dans le porte-carte SIM comme illustré et remplacez le porte-carte SIM dans son emplacement.</p>		<table border="1"> <tr> <td data-bbox="1155 891 1219 927">①</td> <td data-bbox="1219 891 1481 927">TS Module GSM</td> </tr> <tr> <td data-bbox="1155 927 1219 963">②</td> <td data-bbox="1219 927 1481 963">Carte SIM</td> </tr> <tr> <td data-bbox="1155 963 1219 999">③</td> <td data-bbox="1219 963 1481 999">Porte-carte SIM</td> </tr> </table>	①	TS Module GSM	②	Carte SIM	③	Porte-carte SIM
①	TS Module GSM							
②	Carte SIM							
③	Porte-carte SIM							

**Remarque**

Assurez-vous que la carte SIM est correctement orientée dans le porte-carte. Sinon, la connexion entre la carte SIM et le module s'avérera impossible et le bouton d'éjection ne permettra éventuellement plus d'éjecter le porte-carte.

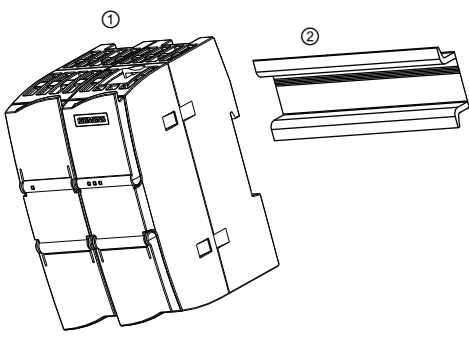
### 3.3.8.3 Installation de l'ensemble adaptateur TeleService

Conditions requises : L'adaptateur TS Adapter doit être raccordé à un module TS et le profilé support doit être installé.

#### Remarque

Si vous montez l'unité TS verticalement ou dans un environnement à fortes vibrations, le module TS peut se détacher de l'adaptateur TS Adapter. Utilisez un crochet d'extrémité 8WA1 808 sur le profilé support pour être sûr que les modules restent connectés.

Tableau 3- 16 Montage et démontage de l'adaptateur TS Adapter

Tâche	Marche à suivre
	<p>Montage :</p> <ol style="list-style-type: none"> <li>1. Accrochez l'ensemble formé par l'adaptateur TS et le module TS ① sur le profilé support ②.</li> <li>2. Faites pivoter l'ensemble vers le bas jusqu'à ce qu'il s'enclenche.</li> <li>3. Repoussez la barrette de fixation sur chaque module pour fixer chaque module au profilé support.</li> </ol> <p>Démontage :</p> <ol style="list-style-type: none"> <li>1. Retirez le câble analogique et le câble Ethernet de la face inférieure de l'adaptateur TS.</li> <li>2. Mettez l'adaptateur TS hors tension.</li> <li>3. Servez-vous d'un tournevis pour dégager les barrettes de fixation sur les deux modules.</li> <li>4. Faites pivoter l'ensemble vers le haut pour le retirer du profilé support.</li> </ol>

#### ATTENTION

Avant la mise hors tension, déconnectez la mise à la terre de l'adaptateur TS Adapter en retirant le câble analogique et le câble Ethernet.

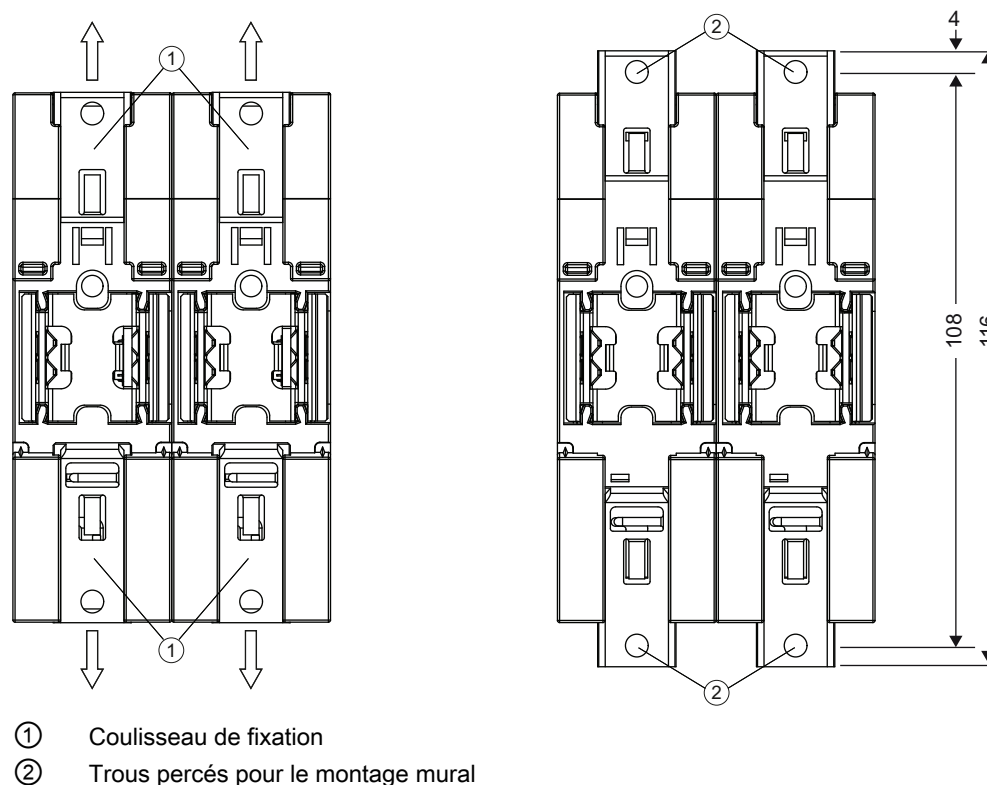
### 3.3.8.4 Montage mural de l'adaptateur TS

Conditions requises : L'adaptateur TS Adapter doit être raccordé à un module TS.

1. Déplacez le coulisseau de fixation ① situé sur la face arrière de l'adaptateur TS et du module TS dans le sens de la flèche jusqu'à ce qu'il s'enclenche.
2. Vissez l'adaptateur TS et le module TS sur le mur de montage prévu à la position repérée par ②.



La figure suivante montre l'adaptateur TS Adapter vu de derrière avec les coulisseaux de fixation ① dans les deux positions.



### 3.4 Conseils de câblage

Une mise à la terre et un câblage corrects de tout l'équipement électrique sont importants pour garantir un fonctionnement optimal de votre système et pour fournir une protection supplémentaire contre le bruit électrique pour votre application et le S7-1200. Vous trouverez les schémas de câblage du S7-1200 dans les caractéristiques techniques (Page 747).

#### Conditions requises

Assurez-vous, avant de mettre à la terre ou de câbler tout appareil électrique, que cet appareil a été mis hors tension. Assurez-vous également que tout équipement associé a été mis hors tension.

### 3.4 Conseils de câblage

Respectez toutes les normes électriques nationales et régionales en vigueur lors du câblage du S7-1200 et des matériels reliés. Installez et exploitez tous les équipements en conformité avec toutes les normes nationales et régionales en vigueur. Demandez l'aide des autorités locales pour déterminer les normes et les réglementations qui s'appliquent à votre cas particulier.

 **ATTENTION**

L'installation ou le câblage du S7-1200 ou d'équipements reliés alors qu'ils sont sous tension peut provoquer un choc électrique ou un fonctionnement imprévisible du matériel. La non-désactivation de l'alimentation complète du S7-1200 et des appareils qui y sont reliés pendant les procédures d'installation ou de désinstallation peut entraîner la mort, des blessures graves ou des dommages matériels importants en raison du choc électrique ou du fonctionnement imprévisible du matériel.

Prenez toujours toutes les mesures de sécurité nécessaires et assurez-vous que l'alimentation de l'automate S7-1200 est coupée avant de tenter d'installer ou de démonter le S7-1200 ou des matériels reliés.

Tenez toujours compte de la sécurité lorsque vous planifiez la mise à la terre et le câblage de votre système S7-1200. Les appareils de commande électroniques, tels que le S7-1200, peuvent subir des défaillances provoquant un fonctionnement inattendu des équipements commandés ou contrôlés. Vous devez, pour cette raison, toujours mettre en oeuvre des dispositifs de sécurité indépendants du S7-1200 pour éviter les blessures ou les dégâts matériels éventuels.

 **ATTENTION**

Les appareils de commande peuvent présenter des défaillances dans des situations non sûres et provoquer un fonctionnement inattendu des appareils pilotés pouvant entraîner la mort, des blessures graves et des dommages matériels importants.


Vous devez donc utiliser une fonction d'arrêt d'urgence, des dispositifs de sécurité électromécaniques ou d'autres sécurités redondantes qui soient indépendants du S7-1200.

### Conseils pour l'isolation

Les barrières d'alimentation en courant alternatif du S7-1200 et les barrières d'E/S vers les circuits en courant alternatif ont été conçues et homologuées comme fournissant une séparation sûre entre les tensions de ligne courant alternatif et les circuits basse tension. Ces barrières incluent une isolation double ou renforcée, ou une isolation de base plus complémentaire, selon diverses normes. Les composants qui traversent ces barrières, tels que coupleurs optiques, condensateurs, transformateurs et relais, ont été homologués comme fournissant une séparation sûre. Les barrières d'isolation qui satisfont à ces exigences ont été identifiées dans les fiches techniques produit du S7-1200 comme ayant une isolation de 1500 V~ ou plus. Cette désignation se base sur un test d'usine de routine de (2Ue + 1000 V~) ou équivalent selon des méthodes approuvées. Les barrières de séparation sûres du S7-1200 ont subi un essai de type à 4242 V-.

La sortie d'alimentation de capteur, les circuits de communication et les circuits logiques internes d'un S7-1200 à alimentation en courant alternatif incluse sont fournis comme TBTS (très basse tension de sécurité) selon EN 61131-2.

Pour conserver le caractère sûr des circuits basse tension du S7-1200, les connexions externes aux ports de communication, les circuits analogiques et tous les circuits d'alimentation et d'E/S 24 V nominal doivent être alimentés par des sources homologuées satisfaisant aux exigences TBTS, TBTP, classe 2, tension limitée ou courant limité, selon diverses normes.

 <b>ATTENTION</b>
L'utilisation d'alimentations non isolées ou à une seule isolation pour alimenter les circuits basse tension à partir d'une ligne en courant alternatif peut provoquer l'apparition de tensions dangereuses sur des circuits censés être sûrs au toucher, tels que des circuits de communication ou un câblage de capteur basse tension.
De telles tensions élevées imprévisibles peuvent entraîner la mort, des blessures graves et des dommages matériels importants.
Utilisez exclusivement des convertisseurs de puissance haute tension-basse tension qui sont homologués comme sources de circuits à tension limitée à toucher sûr.

### Conseils pour la mise à la terre du S7-1200

La meilleure façon de mettre à la terre votre application est de vous assurer que tous les branchements de commun et de terre de votre S7-1200 et des matériels reliés sont mis à la terre en un point unique. Ce point unique doit être relié directement à la masse de terre de votre système.

Tous les fils de terre doivent être aussi courts que possible et doivent avoir une section importante, par exemple 2 mm<sup>2</sup> (14 AWG).

Lorsque vous choisissez des points de référence à la terre, pensez également aux règles de sécurité correspondantes et à l'utilisation correcte des appareils de protection par coupure.

### Conseils pour le câblage du S7-1200

Lorsque vous planifiez le câblage pour votre S7-1200, prévoyez un commutateur de coupure unique qui supprime simultanément le courant pour l'alimentation de la CPU S7-1200, pour tous les circuits d'entrée et pour tous les circuits de sortie. Fournissez une protection contre les surtensions, telle qu'un fusible ou un disjoncteur, afin de limiter les courants de défaut sur le câblage d'alimentation. Pensez également à fournir une protection supplémentaire en plaçant un fusible ou une autre limitation de courant dans chaque circuit de sortie.

Equipez les fils pouvant être menacés par la foudre d'une protection appropriée contre les surtensions.

Évitez de placer les câbles de signaux et de communication de faible tension dans la même goulotte que les câbles d'alimentation en courant alternatif et les câbles pour courant continu à commutation rapide et haute énergie. Posez toujours les fils par paires : conducteur neutre ou commun avec conducteur sous tension ou de signaux.

Utilisez le câble le plus court possible et assurez-vous qu'il est correctement dimensionné pour transporter le courant requis. Le connecteur CPU et SM accepte des sections de fil allant de 2 mm<sup>2</sup> à 0,3 mm<sup>2</sup> (14 AWG à 22 AWG). Le connecteur SB accepte des sections de fil allant de 1,3 mm<sup>2</sup> à 0,3 mm<sup>2</sup> (16 AWG à 22 AWG). Utilisez des fils blindés pour assurer une protection optimale contre le bruit électrique. Typiquement, la mise à la terre du blindage au niveau du S7-1200 donne les meilleurs résultats.

Lorsque vous câblez des circuits d'entrée alimentés par une source externe, incluez un dispositif de protection contre les surtensions dans ce circuit. Une protection externe n'est pas nécessaire pour les circuits alimentés par l'alimentation de capteur 24 V- du S7-1200, car l'alimentation de capteur comprend déjà une limitation de courant.

Tous les modules S7-1200 comportent des connecteurs amovibles pour le câblage utilisateur. Pour éviter que les connexions ne se desserrent, assurez-vous que le connecteur est correctement enfoncé et que le fil est bien enfiché dans le connecteur. Pour éviter d'endommager le connecteur, veillez à ne pas trop serrer les vis. Le couple maximum pour les vis de connecteur CPU et SM est de 0,56 Nm. Le couple maximum pour les vis de connecteur SB est de 0,33 Nm.

Le S7-1200 comporte des barrières d'isolation à certains points pour empêcher des flux de courant indésirables dans votre installation. Tenez compte de ces barrières d'isolation lorsque vous concevez le câblage de votre système. Vous trouverez dans les caractéristiques techniques des informations sur la quantité d'isolation fournie et sur l'emplacement de ces barrières d'isolation. Ne considérez pas les barrières d'isolation estimées à moins de 1500 V~ comme des barrières de sécurité.

### Conseils pour les charges de lampe

Les charges de lampe endommagent les contacts à relais en raison du fort courant de choc à l'activation. Ce courant de choc représente nominale de 10 à 15 fois le courant de régime permanent pour une lampe au tungstène. Un relais d'interposition ou un limiteur de surtension remplaçable est donc recommandé pour des charges de lampe qui seront commutées de nombreuses fois pendant la durée de vie de l'application.

### Conseils pour les charges inductives

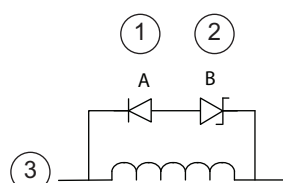
Equipez les charges inductives de circuits de protection qui écrêtent la montée de tension lorsque la sortie de commande est désactivée. Les circuits de protection par écrêtage protègent vos sorties d'une défaillance prématurée due à des tensions élevées associées à la désactivation de charges inductives. En outre, ces circuits limitent le bruit électrique généré lors de la commutation de charges inductives. Placer un circuit de protection par écrêtage externe de façon qu'il soit électriquement aux bornes de la charge et physiquement à proximité de la charge est un moyen très efficace de réduire le bruit électrique.

Les sorties CC du S7-1200 comprennent des circuits de protection interne par écrêtage qui conviennent aux charges inductives dans la plupart des applications. Comme les contacts de sortie relais du S7-1200 peuvent servir pour commuter une charge en courant continu aussi bien qu'une charge en courant alternatif, aucune protection interne n'est fournie.

### Remarque

L'efficacité d'un circuit de protection par écrêtage donné étant toutefois dépendante de l'application, vous devez vérifier qu'il convient à votre cas particulier. Vérifiez toujours que tous les composants utilisés dans votre circuit de protection par écrêtage ont des valeurs nominales convenant à votre application.

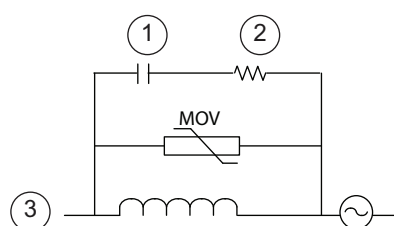
### Circuit de protection par écrêtage typique pour courant continu ou sorties relais qui commutent des charges inductives en courant continu



- ① Diode 1N4001 ou équivalent
- ② Zener 8,2 V (sorties CC),  
Zener 36 V (sorties relais)
- ③ Sortie

Dans la plupart des applications, l'ajout d'une diode (A) en parallèle avec une charge inductive en courant continu suffit, mais si votre application requiert des temps de désactivation plus rapides, nous vous recommandons d'ajouter une diode Zener (B). Veillez à dimensionner votre diode Zener correctement pour la quantité de courant dans votre circuit de sortie.

### Circuit de protection par écrêtage typique pour sorties relais qui commutent des charges inductives en courant alternatif



- ① 0,1 µ F
- ② 100 à 120 Ω
- ③ Sortie

Lorsque vous utilisez une sortie relais pour commuter des charges 115 V/230 V~, placez un circuit résistance/condensateur/varistor métal-oxyde (MOV) approprié en parallèle à la charge CA. Assurez-vous que la tension de travail du varistor est supérieure d'au moins 20 % à la tension de ligne nominale.



## 4.1 Exécution du programme utilisateur

La CPU fournit les types suivants de blocs de code qui vous permettent de créer une structure efficace pour votre programme utilisateur :

- Les blocs d'organisation (OB) définissent la structure du programme. Certains OB ont des événements déclencheurs et un comportement prédéfinis, mais vous pouvez également créer des OB à événements déclencheurs personnalisés. Les plages de numéros d'OB valides sont présentées au paragraphe Priorités d'exécution et mise en file d'attente des événements (Page 79).
- Les fonctions (FC) et blocs fonctionnels (FB) contiennent le code de programme qui correspond à des tâches ou combinaisons de paramètres spécifiques. Chaque FC ou FB fournit un jeu de paramètres d'entrée et de sortie pour partager les données avec le bloc appelant. Un FB utilise également un bloc de données associé - appelé DB d'instance - pour conserver l'état de valeurs entre les exécutions, utilisable par d'autres blocs dans le programme. Les numéros de FC et de FB valides vont de 1 à 65535.
- Les blocs de données (DB) mémorisent des données qui peuvent être utilisées par les blocs de programme. Les numéros de DB valides vont de 1 à 65535.

L'exécution du programme utilisateur commence avec un ou plusieurs blocs d'organisation (OB) de démarrage qui sont exécutés une fois au passage à l'état MARCHE et qui sont suivis par un ou plusieurs OB du cycle de programme qui sont exécutés cycliquement. Un OB peut également être associé à un événement d'alarme - événement standard ou événement d'erreur - et s'exécute à chaque fois que l'événement standard ou d'erreur correspondant se produit.

Une fonction (FC) ou un bloc fonctionnel (FB) est un bloc de code qui peut être appelé dans un OB ou dans un autre FB ou une autre FC, en respectant les profondeurs d'imbrication suivantes :

- 16 en cas d'appel depuis l'OB de cycle de programme ou de démarrage
- 4 en cas d'appel depuis les OB d'alarme temporisée, d'alarme cyclique, d'alarme horaire, d'alarme de processus, d'erreur de temps ou d'erreur de diagnostic

Les FC ne sont pas associées à un bloc de données (DB) particulier alors que les FB sont liés directement à un DB qu'ils utilisent pour transmettre les paramètres et stocker les valeurs intermédiaires et les résultats.

La taille du programme utilisateur, des données et de la configuration est limitée par la mémoire de chargement et la mémoire de travail disponibles dans la CPU. Il n'y a pas de limite spécifique au nombre de blocs de chaque type (OB, FC, FB ou DB). Cependant, le nombre total de blocs est limité à 1024.

Chaque cycle comprend l'écriture dans les sorties, la lecture des entrées, l'exécution des instructions du programme utilisateur et l'exécution du traitement d'arrière-plan. On parle parfois de cycle d'exploration.

4.1 Exécution du programme utilisateur

Les modules (SM, SB, BB, CB, CM ou CP) sont détectés et pris en compte seulement lors de la mise sous tension.

- L'enfichage ou le débrogage d'un module dans le châssis central sous tension (à chaud) ne sont pas pris en charge. N'enfichez jamais de module dans le châssis central et n'en débroguez jamais de module lorsque la CPU est sous tension.

 **ATTENTION**

L'enfichage ou le débrogage d'un module (SM, SB, BB, CD, CM or CP) du châssis central lorsque la CPU est sous tension peut provoquer un comportement imprévisible pouvant entraîner des blessures et des dommages matériels.

Assurez-vous toujours que la CPU et le châssis central sont hors tension avant d'insérer un module dans le châssis central ou d'en débroguer un module.

- Vous pouvez insérer ou retirer une carte mémoire SIMATIC lorsque la CPU est sous tension. Toutefois, enficher ou retirer une carte mémoire lorsque la CPU est à l'état MARCHE provoque le passage de cette dernière à l'état ARRET.

**PRUDENCE**

L'insertion ou le retrait d'une carte mémoire lorsque la CPU est à l'état MARCHE provoque le passage de cette dernière à l'état ARRET, ce qui peut provoquer des dommages aux équipements ou au processus piloté.

Lorsque vous insérez ou retirez une carte mémoire, la CPU passe immédiatement à l'état ARRET. Avant d'insérer ou de retirer une carte mémoire, assurez-vous toujours que la CPU ne pilote pas activement une machine ou un processus. Installez toujours un circuit d'arrêt d'urgence pour votre application ou votre processus.

- Si vous enfichez ou débroguez un module dans un châssis de périphérie décentralisée (PROFINET ou PROFIBUS) alors que la CPU est à l'état MARCHE, la CPU génère une entrée dans la mémoire tampon de diagnostic et reste à l'état MARCHE.

Dans la configuration par défaut, toutes les E/S TOR et analogiques locales sont actualisées en synchronisme avec le cycle à l'aide d'une zone de mémoire interne appelée mémoire image du processus. La mémoire image contient un cliché instantané des entrées et sorties physiques (de la CPU, du Signal Board et des modules d'entrées-sorties).

La CPU exécute les tâches suivantes :

- La CPU écrit les sorties de la mémoire image des sorties dans les sorties physiques.
- La CPU lit les entrées physiques juste avant l'exécution du programme utilisateur et stocke leurs valeurs dans la mémoire image des entrées. Cela garantit que ces valeurs restent cohérentes pendant toute l'exécution des instructions utilisateur.
- La CPU exécute la logique des instructions utilisateur et actualise les valeurs des sorties dans la mémoire image des sorties sans les écrire dans les sorties physiques réelles.

Ce procédé fournit une logique cohérente durant l'exécution des instructions utilisateur pour un cycle donné et empêche le papillotement des sorties physiques dont l'état peut changer à de nombreuses reprises dans la mémoire image des sorties.



Vous pouvez préciser si des entrées et sorties TOR et analogiques doivent être actualisées et stockées automatiquement dans la mémoire image du processus. Si vous insérez un module dans la vue des appareils, ses données se situent dans la mémoire image de la CPU (par défaut). La CPU gère automatiquement l'échange de données entre le module et la zone de mémoire image pendant l'actualisation de la mémoire image. Pour exclure des entrées ou sorties TOR ou analogiques de l'actualisation automatique de la mémoire image, sélectionnez l'appareil approprié dans la Configuration des appareils, affichez l'onglet Propriétés, affichez si nécessaire les détails pour localiser les entrées/sorties désirées, puis sélectionnez "Adresses E/S / ID matériel". Modifiez alors l'entrée pour "Mémoire image" de "MI cyclique" à "---". Pour réintégrer les entrées et sorties dans l'actualisation automatique de la mémoire image, réactivez l'option "MI cyclique".

Vous pouvez lire directement des valeurs d'entrées physiques et écrire directement des valeurs de sorties physiques lors de l'exécution d'une instruction. Une lecture directe accède à l'état en cours de l'entrée physique et n'actualise pas la mémoire image des entrées, que l'entrée soit ou non configurée pour être stockée dans la mémoire image. Une écriture directe dans la sortie physique actualise à la fois la mémoire image des sorties (si la sortie est configurée pour être stockée dans la mémoire image) et la sortie physique. Ajoutez le suffixe ":P" à l'adresse d'entrée/sortie si vous voulez que le programme accède aux données d'E/S directement par l'entrée ou la sortie physique au lieu de la mémoire image.

La CPU prend en charge la périphérie d'E/S à la fois pour les réseaux PROFINET et PROFIBUS (Page 451).

### 4.1.1 Etats de fonctionnement de la CPU

La CPU a trois états de fonctionnement : l'état ARRET (STOP), l'état MISE EN ROUTE et l'état MARCHE (RUN). Des DEL d'état en face avant de la CPU signalent l'état de fonctionnement en cours.

- A l'état ARRET, la CPU n'exécute pas le programme. Vous pouvez y charger un projet.
- A l'état MISE EN ROUTE, les OB de démarrage (s'il y en a) sont exécutés une fois. Les événements d'alarme ne sont pas traités pendant la mise en route.
- A l'état MARCHE, les OB de cycle de programme sont exécutés de manière répétée. Des événements d'alarme peuvent survenir et être traités à tout moment à l'état MARCHE. Certaines parties d'un projet peuvent être chargées dans la CPU à l'état MARCHE (Page 739).

La CPU accepte un démarrage à chaud pour le passage à l'état MARCHE. Le démarrage à chaud ne comprend pas d'effacement général de la mémoire. Toutes les données systèmes et utilisateur non rémanentes sont initialisées lors du démarrage à chaud. Les données utilisateur rémanentes sont conservées.

#### 4.1 Exécution du programme utilisateur

Un effacement général efface toute la mémoire de travail, efface les zones de mémoire rémanentes et non rémanentes et copie la mémoire de chargement dans la mémoire de travail. Un effacement général n'efface pas la mémoire tampon de diagnostic ni les valeurs sauvegardées de manière permanente de l'adresse IP.

---

#### Remarque

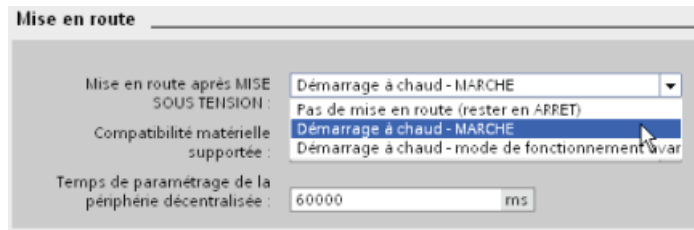
Lorsque vous chargez un ou plusieurs DB de STEP 7 V11 dans une CPU S7-1200 V2, les valeurs rémanentes et non rémanentes de ces DB sont réinitialisées. Le passage suivant à l'état MARCHE prend la forme d'un démarrage à chaud qui entraîne la réinitialisation de toutes les données non rémanentes et l'affectation des valeurs conservées à toutes les données rémanentes.

Lorsque vous chargez des éléments de projet (configuration d'appareil, blocs de code ou DB, par exemple) de STEP 7 V10.5 dans n'importe quelle CPU S7-1200 ou de STEP 7 V11 dans une CPU S7-1200 V1 (ou une CPU V2 qui a été configurée en tant que CPU V1), le passage suivant à l'état MARCHE entraîne la réinitialisation de **tous** les DB dans le projet.

---

Vous pouvez configurer le paramètre "Démarrage après mise sous tension" de la CPU. Ce paramètre de configuration apparaît dans "Configuration d'appareil" pour la CPU, sous "Mise en route". A la mise sous tension, la CPU exécute une séquence de vérifications du diagnostic de mise en route et effectue l'initialisation du système. Pendant l'initialisation du système, la CPU efface tous les mementos non rémanents et redonne à tous les DB non rémanents leurs valeurs initiales provenant de la mémoire de chargement. La CPU conserve les mementos rémanents et le contenu des DB rémanents et passe à l'état de fonctionnement approprié. La détection de certaines erreurs empêche le passage de la CPU à l'état MARCHE. La CPU accepte les configurations suivantes :

- Pas de mise en route (rester en ARRET)
- Démarrage à chaud - MARCHE
- Démarrage à chaud - mode de fonctionnement avant la mise hors tension



## PRUDENCE

La CPU peut passer à l'état ARRET suite à des défauts réparables, tels que la défaillance d'un module d'entrées-sorties remplaçable, ou à des défauts temporaires, tels qu'une perturbation sur la ligne d'alimentation ou un événement erratique à la mise sous tension.

Si la CPU a été configurée à "Démarrage à chaud - mode de fonctionnement avant la mise hors tension", elle ne repassera pas à l'état MARCHE une fois le défaut réparé ou éliminé tant qu'elle n'aura pas reçu de STEP 7 une nouvelle commande de passage à l'état MARCHE. En l'absence de nouvelle commande, l'état ARRET sera conservé comme état de fonctionnement avant la mise hors tension.

Typiquement, les CPU destinées à fonctionner indépendamment d'une liaison STEP 7 doivent être configurées à "Démarrage à chaud - MARCHE" afin qu'elles puissent repasser à l'état MARCHE lors d'une mise hors tension puis sous tension suivant l'élimination de défauts.

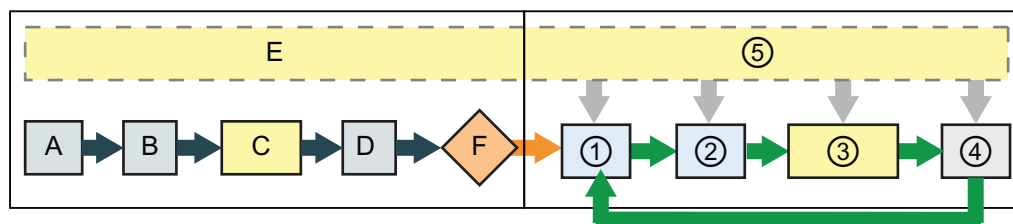
Vous pouvez changer l'état de fonctionnement en cours à l'aide des commandes "ARRET" et "MARCHE" des outils en ligne du logiciel de programmation. Vous pouvez aussi insérer une instruction STP dans votre programme pour faire passer la CPU à l'état ARRET. Cela vous permet d'arrêter l'exécution de votre programme selon la logique.

- A l'état ARRET, la CPU traite toute demande de communication de manière appropriée et effectue un auto-diagnostic. La CPU n'exécute pas le programme utilisateur et il n'y a pas d'actualisation automatique de la mémoire image du processus.

Vous ne pouvez charger votre projet que lorsque la CPU est à l'état ARRET.

- Dans les états MISE EN ROUTE et MARCHE, la CPU effectue les tâches présentées dans la figure suivante.

4.1 Exécution du programme utilisateur



MISE EN ROUTE

- A Efface la mémoire image des entrées (I).
- B Initialise les sorties avec leur dernière valeur ou leur valeur de remplacement.
- C Exécute les OB de démarrage.
- D Copie l'état des entrées physiques dans la mémoire image des entrées.
- E Enregistre tous les événements d'alarme dans la file d'attente en vue de leur traitement après passage à l'état MARCHE.
- F Valide l'écriture de la mémoire image des sorties dans les sorties physiques.

MARCHE

- ① Ecrit la mémoire image des sorties dans les sorties physiques.
- ② Copie l'état des entrées physiques dans la mémoire image des entrées.
- ③ Exécute les OB de cycle de programme.
- ④ Réalise des tests d'auto-diagnostic.
- ⑤ Traite les alarmes et la communication à n'importe quel moment du cycle.

Traitement de MISE EN ROUTE

A chaque fois que l'état de fonctionnement passe de ARRET à MARCHE, la CPU efface la mémoire image des entrées, initialise la mémoire image des sorties et traite les OB de démarrage. Tout accès en lecture à la mémoire image des entrées par des instructions dans les OB de démarrage lit zéro et non l'état en cours de l'entrée physique. Ainsi, pour lire l'état en cours de l'entrée physique pendant la mise en route, vous devez effectuer une lecture directe. Les OB de démarrage ainsi que tous les FB et FC associés sont ensuite exécutés. En présence de plus d'un OB de démarrage, chacun est exécuté par ordre de numéro d'OB, l'OB de plus petit numéro étant exécuté en premier.

Chaque OB de démarrage comprend des informations de démarrage qui vous aident à déterminer la validité des données rémanentes et l'horloge temps réel. Vous pouvez programmer dans les OB de démarrage des instructions qui examinent ces valeurs de démarrage afin de déclencher l'action appropriée. Les informations de démarrage suivantes sont prises en charge par les OB de démarrage :

Tableau 4- 1 Informations de démarrage prises en charge par les OB de démarrage

Entrée	Type de données	Description
LostRetentive	Bool	Ce bit est vrai si les zones de stockage de données rémanentes ont été perdues.
LostRTC	Bool	Ce bit est vrai si l'horloge temps réel a été perdue.

La CPU exécute également les tâches suivantes pendant le traitement de mise en route.

- Les alarmes sont mises en file d'attente mais ne sont pas traitées pendant la phase de mise en route.
- Aucune surveillance du temps de cycle n'est effectuée pendant la phase de mise en route.
- Il est possible de modifier la configuration de HSC (compteur rapide), PWM (modulation de largeur des impulsions) et des modules PtP (communication point à point) pendant la mise en route.
- Mais l'exécution réelle de HSC, PWM et des modules de communication point à point n'est possible qu'à l'état MARCHE.

Une fois l'exécution des OB de démarrage achevée, la CPU passe à l'état MARCHE et traite les tâches de commande lors d'un cycle continu.

### Voir aussi

Instruction Commuter en STOP (Page 251)

Panneau de commande CPU de la CPU en ligne (Page 727)

## 4.1.2 Traitement du cycle à l'état MARCHE

A chaque cycle, la CPU écrit dans les sorties, lit les entrées, exécute le programme utilisateur, actualise les modules de communication et réagit aux événements d'alarme utilisateur et aux demandes de communication. Les demandes de communication sont traitées périodiquement pendant tout le cycle.

Ces actions (à l'exception des événements d'alarme utilisateur) sont assurées régulièrement et à la suite les unes des autres. Les événements d'alarme utilisateur activés sont gérés selon leur priorité dans l'ordre où ils se produisent.

Le système garantit que le cycle s'achèvera au bout d'une durée appelée temps de cycle maximum. Dans le cas contraire, un événement d'erreur de temps est généré.

- Chaque cycle commence par l'extraction des valeurs en cours des sorties TOR et analogiques de la mémoire image et leur écriture dans les sorties physiques de la CPU, du SB et des SM configurées pour l'actualisation automatique des E/S (configuration par défaut). Lorsqu'on accède à une sortie physique par une instruction, la mémoire image des sorties et la sortie physique elle-même sont toutes deux actualisées.
- Chaque cycle se poursuit par la lecture des valeurs en cours des entrées TOR et analogiques dans la CPU, le SB et les SM configurées pour l'actualisation automatique des E/S (configuration par défaut), puis par l'écriture de ces valeurs dans la mémoire image des entrées. Lorsqu'on accède à une entrée physique par une instruction, l'instruction accède à la valeur de l'entrée physique mais la mémoire image des entrées n'est pas actualisée.
- Une fois les entrées lues, le programme utilisateur est exécuté de la première instruction jusqu'à l'instruction finale. Cela comprend tous les OB du cycle de programme avec leurs FC et FB associés. Les OB du cycle de programme sont exécutés par ordre de numéro d'OB, l'OB de plus petit numéro étant exécuté en premier.

#### 4.1 Exécution du programme utilisateur

Le traitement de la communication se produit périodiquement pendant tout le cycle, interrompant éventuellement l'exécution du programme utilisateur.

Les contrôles d'auto-diagnostic comprennent des contrôles périodiques du système ainsi que des contrôles de l'état des modules d'E/S.

Des alarmes peuvent se produire à tout endroit du cycle ; elles sont déclenchées sur événement. Lorsqu'un événement se produit, la CPU interrompt le cycle et appelle l'OB qui a été configuré pour traiter cet événement. Lorsque l'OB a fini de traiter l'événement, la CPU reprend l'exécution du programme utilisateur à l'endroit où elle s'était interrompue.

#### 4.1.3 Blocs d'organisation (OB)

Les OB pilotent l'exécution du programme utilisateur. Chaque OB doit avoir un numéro d'OB unique. Les numéros en dessous de 200 sont réservés aux OB par défaut. Il faut numéroter les autres OB à partir de 200.

Des événements spécifiques dans la CPU déclenchent l'exécution d'un bloc d'organisation. Les OB ne peuvent pas s'appeler les uns les autres ni être appelés dans une FC ou un FB. Seul un événement déclencheur, tel qu'une alarme de diagnostic ou un intervalle de temps, peut lancer l'exécution d'un OB. La CPU traite les OB en fonction de leur classe de priorité respective, les OB de priorité supérieure étant exécutés avant les OB de priorité inférieure. La classe de priorité la plus basse est 1 (pour le cycle du programme principal) et la classe de priorité la plus élevée est 26 (pour les alarmes d'erreur de temps).

Les OB commandent les opérations suivantes :

- Les OB du cycle de programme s'exécutent cycliquement lorsque la CPU est à l'état MARCHE. Le bloc principal du programme est un OB de cycle de programme. C'est là que vous placez les instructions commandant votre programme et que vous appelez d'autres blocs utilisateur. Il est possible d'avoir plusieurs OB de cycle de programme qui sont exécutés par ordre numérique. L'OB 1 est l'OB pris par défaut. Les autres OB de cycle de programme doivent être identifiés en tant que OB 200 ou plus.
- Les OB de démarrage s'exécutent une fois lorsque la CPU passe de l'état ARRET à l'état MARCHE, y compris lors de la mise sous tension en mode MARCHE et en transitions commandées de ARRET à MARCHE. Une fois les OB de démarrage achevés, l'OB principal "Cycle de programme" commence son exécution. Il est possible d'avoir plusieurs OB de démarrage. L'OB 100 est l'OB pris par défaut. Les autres doivent être supérieurs ou égaux à OB 200.
- Les OB d'alarme cyclique s'exécutent à un rythme indiqué. Un OB d'alarme cyclique interrompt l'exécution cyclique du programme à intervalles définis par l'utilisateur, toutes les 2 secondes par exemple. Vous pouvez configurer au plus 4 événements en tout pour les alarmes temporisées et les alarmes cycliques à un moment donné, un OB étant autorisé pour chaque événement d'alarme temporisée et d'alarme cyclique configuré. L'OB doit être supérieur ou égal à OB 200.

- Les OB d'alarme de processus s'exécutent lorsque l'événement d'alarme de processus correspondant se produit, y compris fronts montants et descendants sur les entrées TOR intégrées et événements HSC. Un OB d'alarme de processus interrompt l'exécution cyclique normale du programme en réaction à un signal provenant d'un événement du processus. Vous définissez les événements dans les propriétés de la configuration matérielle. Un OB est autorisé pour chaque événement d'alarme de processus. L'OB doit être supérieur ou égal à OB 200.
- Un OB d'erreur de temps s'exécute lorsque le temps de cycle maximum est dépassé ou qu'un événement erreur de temps se produit. L'OB qui traite l'alarme d'erreur de temps est l'OB 80. S'il est déclenché, il s'exécute en interrompant l'exécution normale du programme cyclique ou tout autre OB d'événement. Les événements qui déclenchent l'alarme d'erreur de temps et la réaction de la CPU à ces événements sont décrits ci-dessous :
  - Dépassement du temps de cycle maximum : Vous configurez le temps de cycle maximum dans les propriétés de la CPU. Si l'OB 80 n'existe pas, la CPU réagit au dépassement du temps maximum en passant à l'état ARRET.
  - Erreurs de temps : Si l'OB 80 n'existe pas, la réaction de la CPU est de rester à l'état MARCHE. Des erreurs de temps se produisent si l'événement horaire est manqué ou répété, si la file d'attente déborde ou si un OB d'événement (alarme temporisée, alarme horaire ou alarme cyclique) démarre avant que la CPU n'ait achevé l'exécution du premier.

L'apparition de l'un de ces événements génère une entrée décrivant l'événement dans la mémoire tampon de diagnostic. L'entrée de mémoire tampon de diagnostic est générée indépendamment de l'existence de l'OB 80.

- Les OB d'alarme de diagnostic s'exécutent lorsqu'une erreur de diagnostic est détectée et signalée. Un OB d'alarme de diagnostic interrompt l'exécution cyclique normale du programme si un module apte au diagnostic identifie une erreur (si l'alarme de diagnostic a été activée pour ce module). OB 82 est le seul numéro d'OB accepté pour l'événement erreur de diagnostic. Vous pouvez inclure une instruction STP dans votre OB 82 si vous souhaitez que la CPU passe à l'état ARRET lors de la réception de ce type d'erreur. S'il n'y a pas d'OB de diagnostic dans le programme, la CPU ne tient pas compte de l'erreur (et reste à l'état MARCHE).

#### 4.1.4 Priorités d'exécution et mise en file d'attente des événements

Le traitement de la CPU est commandé par des événements. Un événement déclenche un OB d'alarme qu'il faut exécuter. Vous pouvez définir l'OB d'alarme pour un événement lors de la création du bloc, lors de la configuration de l'appareil ou à l'aide d'une opération ATTACH ou DETACH. Certains événements se produisent sur une base régulière, tels les événements cycle de programme ou les événements alarme cyclique. D'autres événements ne se produisent qu'une fois, tels l'événement démarrage et les événements alarme temporisée. Certains événements se produisent lorsqu'il y a un changement déclenché par le matériel, par exemple un événement front sur une entrée ou un événement compteur rapide. Il y a également des événements comme l'événement erreur de diagnostic et l'événement erreur de temps qui ne se produisent que lorsqu'il y a une erreur. Les priorités et files d'attente d'événements sont utilisées pour déterminer l'ordre de traitement des OB d'alarme d'événement.

#### 4.1 Exécution du programme utilisateur

L'événement cycle de programme se produit une fois à chaque cycle. Pendant le cycle de programme, la CPU écrit dans les sorties, lit les entrées et exécute les OB de cycle de programme. L'événement cycle de programme est obligatoire et est toujours activé. Vous pouvez n'avoir aucun OB de cycle de programme ou avoir plusieurs OB sélectionnés pour l'événement cycle de programme. Une fois l'événement cycle de programme déclenché, l'OB de cycle de programme ayant le plus petit numéro (généralement l'OB 1) est exécuté. Les autres OB de cycle de programme sont exécutés de façon séquentielle (par ordre numérique) à l'intérieur du cycle de programme.

Les événements alarme cyclique vous permettent de configurer l'exécution d'un OB d'alarme avec une périodicité configurée. La périodicité initiale est configurée lors de la création de l'OB et de sa sélection comme OB d'alarme cyclique. Un événement alarme cyclique interrompt le cycle de programme et exécute l'OB d'alarme cyclique (l'événement alarme cyclique est dans une classe de priorité supérieure à celle de l'événement cycle de programme).

Un seul OB d'alarme cyclique peut être associé à un événement alarme cyclique.

Il est possible de définir un déphasage pour chaque événement alarme cyclique de sorte que l'exécution les unes par rapport aux autres d'alarmes cycliques de même périodicité peut être décalée du temps de déphasage. Le déphasage par défaut est 0. Pour modifier le déphasage initial ou la périodicité initiale d'un événement alarme cyclique, cliquez avec le bouton droit de la souris sur l'OB d'alarme cyclique dans l'arborescence du projet, cliquez sur "Propriétés" puis sur "Alarme cyclique" et entrez les nouvelles valeurs initiales. Vous pouvez également connaître et modifier la périodicité et le déphasage depuis votre programme à l'aide des instructions Interroger alarme cyclique (QRY\_CINT) et Régler alarme cyclique (SET\_CINT). Les valeurs de périodicité et de déphasage définies à l'aide de l'instruction SET\_CINT ne perdurent pas en cas de mise hors tension puis sous tension ou de passage à l'état ARRET ; la périodicité et le déphasage reprennent leur valeur initiale après une mise hors tension puis sous tension ou un passage à l'état ARRET. La CPU prend en charge un total de quatre événements d'alarme cyclique et d'alarme temporisée.

L'événement démarrage se produit une fois lors d'une transition de ARRET à MARCHE et provoque l'exécution des OB de démarrage. Plusieurs OB peuvent être sélectionnés pour l'événement démarrage. Les OB de démarrage sont exécutés par ordre numérique.

Les événements alarme temporisée vous permettent de configurer l'exécution d'un OB d'alarme à l'expiration d'un temps de retard indiqué. Le temps de retard est défini au moyen de l'instruction SRT\_DINT. Les événements alarme temporisée interrompent le cycle de programme afin d'exécuter l'OB d'alarme temporisée. Un seul OB d'alarme temporisée peut être associé à un événement alarme temporisée. La CPU accepte quatre événements alarme temporisée.

Les événements alarme de processus sont déclenchés par un changement dans le matériel, par exemple un front montant ou descendant sur une entrée ou un événement HSC (compteur rapide). Un OB d'alarme peut être sélectionné pour chaque événement alarme de processus. Les événements alarme de processus sont activés dans la configuration des appareils. Les OB pour l'événement sont indiqués dans la configuration des appareils ou au moyen d'une instruction ATTACH dans le programme utilisateur. La CPU accepte plusieurs événements alarme de processus. Le nombre exact d'événements dépend du modèle de CPU et du nombre d'entrées.



Les événements erreur de temps et erreur de diagnostic sont déclenchés lorsque la CPU détecte une erreur. Ces événements constituent une classe de priorité supérieure à celle des autres événements d'alarme et peuvent interrompre l'exécution des événements alarme temporisée, alarme cyclique et alarme de processus. Un OB d'alarme peut être indiqué pour chacun des événements erreur de temps et erreur de diagnostic.

### Comprendre les priorités d'exécution et la mise en file d'attente des événements

Le nombre d'événements en attente provenant d'une source unique est limité, avec une file d'attente différente pour chaque type d'événement. Lorsque la limite d'événements en attente pour un type d'événement donné est atteinte, l'événement suivant qui survient est perdu. Reportez-vous au paragraphe suivant sur la compréhension des événements d'erreur de temps pour plus d'informations concernant les débordements de file d'attente.

Chaque événement de CPU a une priorité associée. Vous ne pouvez pas modifier la priorité d'un OB. En général, les événements sont traités selon l'ordre de priorité (priorité la plus élevée en premier). Les événements de même priorité sont traités selon le principe "premier arrivé, premier servi".

Tableau 4-2 Événements d'OB

Événement	Numéro d'OB	Quantité autorisée	Événement déclencheur	Priorité d'OB
Cycle de programme	OB 1, OB 200 à OB 65535	1 événement de cycle de programme Plusieurs OB autorisés	<ul style="list-style-type: none"> <li>Fin de l'OB de démarrage</li> <li>Fin du dernier OB de cycle de programme</li> </ul>	1
Mise en route	OB 100, OB 200 à OB 65535	1 événement de démarrage <sup>1,2</sup> Plusieurs OB autorisés	Transition ARRET/MARCHE	1
Temps	OB 200 à OB 65535	Jusqu'à 4 événements horaires <sup>3</sup> 1 OB par événement	Un événement OB d'alarme temporisée est programmé.	3
			Un événement OB d'alarme cyclique est programmé.	7
Processus	OB 200 à OB 65535	Jusqu'à 50 événements d'alarme de processus <sup>4</sup> 1 OB par événement	Fronts : <ul style="list-style-type: none"> <li>Événements front montant : 16 max.</li> <li>Événements front descendant : 16 max.</li> </ul>	5
			Pour HSC : <ul style="list-style-type: none"> <li>CV=PV : 6 max.</li> <li>Inversion du sens : 6 max.</li> <li>Réinitialisation externe : 6 max.</li> </ul>	6

4.1 Exécution du programme utilisateur

Événement	Numéro d'OB	Quantité autorisée	Événement déclencheur	Priorité d'OB
Erreur de diagnostic	OB 82	1 événement (uniquement si l'OB 82 a été chargé)	Le module transmet une erreur.	9
Erreur de temps	OB 80	1 événement (uniquement si l'OB 80 a été chargé) <sup>5</sup>	<ul style="list-style-type: none"> <li>Le temps de cycle maximal a été dépassé.</li> <li>Une deuxième alarme (cyclique ou horaire) a été démarrée avant que la CPU n'ait fini d'exécuter la première alarme.</li> </ul>	26

- <sup>1</sup> L'événement démarrage et l'événement cycle de programme ne se produisent jamais en même temps, car l'événement démarrage doit s'achever avant que l'événement cycle de programme ne commence (gestion par le système d'exploitation).
- <sup>2</sup> Seul l'événement erreur de diagnostic (OB 82) interrompt l'événement démarrage. Tous les autres événements sont mis en attente afin d'être traités une fois l'événement de démarrage terminé.
- <sup>3</sup> La CPU met à disposition un total de 4 événements horaires que les OB d'alarme temporisée et les OB d'alarme cyclique se partagent. Le nombre d'OB d'alarme temporisée et d'OB d'alarme cyclique dans votre programme utilisateur ne peut pas être supérieur à 4.
- <sup>4</sup> Vous pouvez avoir plus de 50 événements alarme de processus si vous utilisez les instructions DETACH et ATTACH.
- <sup>5</sup> Vous pouvez configurer la CPU afin qu'elle reste à l'état MARCHE si le temps de cycle maximum a été dépassé ou vous pouvez réinitialiser le temps de cycle avec l'instruction RE\_TRIGR. Toutefois, la CPU passera à l'état ARRET au deuxième dépassement du temps de cycle maximum pendant le même cycle.

Une fois que l'exécution d'un OB de priorité comprise entre 2 et 25 a commencé, le traitement de cet OB ne peut pas être interrompu par l'apparition d'un autre événement, à l'exception de l'OB 80 (événement erreur de temps dont la priorité est égale à 26). Tous les autres événements sont mis en file d'attente en vue de leur traitement ultérieur, ce qui permet à l'OB en cours de s'achever.

**Temps d'attente**

Le temps d'attente d'un événement d'alarme (le temps qui s'écoule entre la notification à la CPU qu'un événement s'est produit et le moment où la CPU exécute la première instruction dans l'OB gérant cet événement) est d'environ 175 µs, à condition qu'un OB de cycle de programme soit le seul programme de service d'événement actif au moment de l'événement d'alarme.

**Comprendre les événements d'erreur de temps**

L'apparition d'une quelconque des différentes situations d'erreur de temps provoque un événement d'erreur de temps. Les erreurs de temps suivantes sont prises en charge :

- Temps de cycle maximum dépassé
- Impossible de démarrer l'OB requis
- Débordement de file d'attente

L'erreur Temps de cycle maximum dépassé se produit si le cycle du programme ne s'achève pas pendant le temps de cycle maximum défini. Référez-vous au paragraphe "Surveillance du temps de cycle" du "Manuel système S7-1200 (Page 85)" pour plus d'informations sur le temps de cycle maximum, la manière de configurer le temps de cycle maximum et la réinitialisation de la temporisation de cycle.

L'erreur Impossible de démarrer l'OB requis se produit si un OB est demandé par une alarme cyclique, une alarme temporisée ou une alarme horaire mais qu'il est déjà en cours d'exécution.

L'erreur Débordement de file d'attente apparaît si les alarmes se produisent plus vite qu'elles ne peuvent être traitées. Le nombre d'événements en attente (dans une file) est limité grâce à une file d'attente différente pour chaque type d'événement. Si un événement se produit alors que la file d'attente correspondante est pleine, un événement d'erreur de temps est généré.

Tous les événements erreur de temps déclenchent l'exécution de l'OB 80 si ce dernier existe. Si le programme utilisateur ne contient pas d'OB 80, la configuration d'appareil de la CPU détermine la réaction de la CPU à l'erreur de temps :

- La configuration par défaut pour les erreurs de temps, comme par exemple le démarrage d'une deuxième alarme cyclique avant que la CPU n'ait fini d'exécuter la première, est que la CPU reste à l'état MARCHE.
- La configuration par défaut pour le dépassement du temps de cycle maximum est que la CPU passe à l'état ARRET.

Vous pouvez utiliser l'instruction RE\_TRIGR pour réinitialiser le temps de cycle maximum. Toutefois, si deux erreurs "Temps de cycle maximum dépassé" se produisent pendant le même cycle de programme sans réinitialisation de la temporisation du cycle, la CPU passe à l'état ARRET que l'OB 80 existe ou non. Référez-vous au paragraphe "Surveillance du temps de cycle" du "Manuel système S7-1200" (Page 85).

L'OB 80 comprend des informations de déclenchement qui vous aident à déterminer quel événement et quel OB ont généré l'erreur de temps. Vous pouvez programmer dans l'OB 80 des instructions pour examiner ces valeurs de déclenchement et prendre les mesures appropriées.

Tableau 4- 3 Informations de déclenchement pour l'OB 80

Entrée	Type de données	Description
fault_id	BYTE	16#01 : temps de cycle maximum dépassé 16#02 : impossible de démarrer l'OB requis 16#07 et 16#09 : débordement de file d'attente
csg_OBnr	OB_ANY	Numéro de l'OB en cours d'exécution lorsque l'erreur s'est produite
csg_prio	UINT	Priorité de l'OB à l'origine de l'erreur

Il n'existe pas d'OB 80 d'erreur de temps à la création d'un nouveau projet. Si vous le désirez, vous ajoutez l'OB 80 d'erreur de temps à votre projet en double-cliquant sur "Ajouter nouveau bloc" sous "Blocs de programme" dans l'arborescence, puis en choisissant "Bloc d'organisation" et enfin "Erreur de temps".

### Comprendre les événements d'erreur de diagnostic

Certains appareils analogiques (locaux), PROFINET et PROFIBUS sont capables de détecter et de signaler des erreurs de diagnostic. L'apparition ou l'élimination d'une quelconque des différentes situations d'erreur de diagnostic provoque un événement d'erreur de diagnostic. Les erreurs de diagnostic suivantes sont prises en charge :

- Pas de courant utilisateur
- Limite supérieure dépassée
- Limite inférieure dépassée
- Rupture de fil
- Court-circuit

Les événements erreur de diagnostic déclenchent l'exécution de l'OB 82 si ce dernier existe. Si l'OB 82 n'existe pas, la CPU ne tient pas compte de l'erreur. Il n'existe pas d'OB 82 d'alarme de diagnostic à la création d'un nouveau projet. Si vous le désirez, vous ajoutez l'OB 82 d'alarme de diagnostic à votre projet en double-cliquant sur "Ajouter nouveau bloc" sous "Blocs de programme" dans l'arborescence, puis en choisissant "Bloc d'organisation" et enfin "Alarme de diagnostic".

---

#### Remarque

#### Erreurs de diagnostic pour les appareils analogiques locaux multivoies (E/S, RTD et Thermocouple)

L'OB 82 d'alarme de diagnostic ne peut signaler qu'une erreur de diagnostic de voie à la fois.

Si deux voies d'un appareil multivoie présentent une erreur, la deuxième erreur ne déclenche l'OB 82 que dans les conditions suivantes : la première erreur de voie est corrigée, l'exécution de l'OB 82 déclenchée par la première erreur est achevée et la deuxième erreur persiste.

---

L'OB 82 comprend des informations de déclenchement qui vous aident à déterminer si l'événement est dû à l'apparition ou à l'élimination d'une erreur, ainsi que l'appareil et la voie ayant signalé l'erreur. Vous pouvez programmer dans l'OB 82 des instructions pour examiner ces valeurs de déclenchement et prendre les mesures appropriées.

Tableau 4- 4 Informations de déclenchement pour l'OB 82

Entrée	Type de données	Description
IOnstate	WORD	Etat E/S de l'appareil : <ul style="list-style-type: none"> <li>• Bit 0 = 1 si la configuration est correcte et = 0 si la configuration n'est plus correcte.</li> <li>• Bit 4 = 1 si une erreur est présente (rupture de fil, par exemple). Bit 4 = 0 en l'absence d'erreur.</li> <li>• Bit 5 = 1 si la configuration <b>n'est pas</b> correcte et = 0 si la configuration est à nouveau correcte.</li> <li>• Bit 6 = 1 si une erreur d'accès aux E/S s'est produite. Reportez-vous à l'addr qui contient l'identificateur matériel de l'E/S en erreur d'accès. Bit 6 = 0 en l'absence d'erreur.</li> </ul>
laddr	HW_ANY	Identificateur matériel de l'appareil ou de l'unité fonctionnelle ayant signalé l'erreur <sup>1</sup>
channel	UINT	Numéro de voie
multierror	BOOL	VRAI en présence de plusieurs erreurs

<sup>1</sup> L'entrée laddr contient l'identificateur matériel de l'appareil ou de l'unité fonctionnelle qui a renvoyé l'erreur. L'identificateur matériel est affecté automatiquement lorsque des composants sont insérés dans la vue des appareils ou du réseau et il apparaît dans l'onglet Constantes des variables API. Un nom est également défini automatiquement pour l'identificateur matériel. Ces entrées dans l'onglet Constantes des variables API ne peuvent pas être modifiées.

### 4.1.5 Surveillance du temps de cycle

Le temps de cycle est le temps nécessaire au système d'exploitation de la CPU pour exécuter la phase cyclique de l'état MARCHE. La CPU fournit deux méthodes de surveillance du temps de cycle :

- Temps de cycle maximum
- Temps de cycle minimum fixe

La surveillance du cycle commence une fois l'événement démarrage achevé. La configuration de cette fonction se fait dans la "Configuration d'appareil" pour la CPU, sous "Temps de cycle".

La CPU surveille toujours le temps de cycle et réagit si le temps de cycle maximum est dépassé. Si le temps de cycle maximum configuré est dépassé, une erreur est générée et est traitée de l'une des deux manières suivantes :

- Si le programme utilisateur ne contient pas d'OB 80, la CPU génère une erreur et passe à l'état ARRET. Vous pouvez modifier la configuration de la CPU afin qu'elle ne tienne pas compte de cette erreur de temps et reste à l'état MARCHE. La configuration par défaut est que la CPU passe à l'état ARRET.
- Si le programme utilisateur contient un OB 80, la CPU exécute l'OB 80.

4.1 Exécution du programme utilisateur

L'instruction RE\_TRIGR (Redémarrer surveillance du temps de cycle) vous permet de réinitialiser la temporisation qui mesure le temps de cycle. Toutefois, cette instruction ne fonctionne que si elle est exécutée dans un OB de cycle de programme ; elle n'est pas prise en compte si elle est exécutée dans l'OB 80. Si le temps de cycle maximum est dépassé deux fois pendant le même cycle de programme, sans exécution de l'instruction RE\_TRIGR entre les deux dépassements, la CPU passe immédiatement à l'état ARRET. Des exécutions répétées de l'instruction RE\_TRIGR peuvent créer une boucle sans fin ou un cycle très long.

Typiquement, le cycle s'exécute aussi rapidement qu'il le peut et le cycle suivant commence dès que le cycle en cours s'achève. La durée d'un cycle peut varier d'un cycle à l'autre en fonction du programme utilisateur et des tâches de communication. Pour éliminer cette variation, la CPU prend en charge un temps de cycle minimum fixe optionnel, également appelé cycle fixe. Lorsque cette fonction optionnelle est activée et qu'un temps de cycle minimum en ms est fixé, la CPU maintient le temps de cycle minimum à  $\pm 1$  ms près pour l'achèvement de chaque cycle CPU.

Si la CPU achève le cycle normal en moins de temps que le temps de cycle minimum spécifié, elle passe le temps restant du cycle à effectuer le diagnostic d'exécution et/ou à traiter des demandes de communication. De cette manière, la CPU utilise toujours une quantité de temps fixe pour achever le cycle.

Si la CPU n'achève pas le cycle pendant le temps de cycle minimum indiqué, elle continue à exécuter le cycle normalement jusqu'à son achèvement (traitement de la communication inclus) et ne crée aucune réaction système en réponse à ce dépassement du temps de cycle minimum. Le tableau suivant présente les plages et les valeurs par défaut pour les fonctions de surveillance du temps de cycle.

Tableau 4- 5 Plage pour le temps de cycle

Temps de cycle	Plage (ms)	Valeur par défaut
Temps de cycle maximum <sup>1</sup>	1 à 6000	150 ms
Temps de cycle minimum fixe <sup>2</sup>	1 à temps de cycle maximum	Désactivé

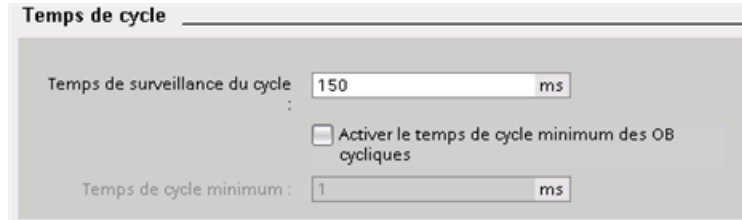
<sup>1</sup> Le temps de cycle maximum est toujours activé. Configurez un temps de cycle compris entre 1 ms et 6000 ms. La valeur par défaut est 150 ms.

<sup>2</sup> Le temps de cycle minimum fixe est facultatif ; il est désactivé par défaut. Si nécessaire, configurez un temps de cycle compris entre 1 ms et le temps de cycle maximum.

## Configuration du temps de cycle et de la charge due à la communication

Vous vous servez des propriétés CPU dans la configuration des appareils pour configurer les paramètres suivants :

- Temps de cycle : Vous pouvez entrer un temps de cycle maximum. Vous pouvez également entrer un temps de cycle minimum fixe.



The screenshot shows a configuration window titled "Temps de cycle". It contains two input fields: "Temps de surveillance du cycle" with the value "150" and unit "ms", and "Temps de cycle minimum" with the value "1" and unit "ms". There is also a checkbox labeled "Activer le temps de cycle minimum des OB cycliques" which is currently unchecked.

- Charge due à la communication : Vous pouvez configurer un pourcentage du temps à réserver aux tâches de communication.



The screenshot shows a configuration window titled "Charge due à la communication". It contains one input field: "Charge du cycle due à la communication" with the value "20" and unit "%".

Reportez-vous au paragraphe "Surveillance du temps de cycle" (Page 85) pour plus d'informations sur le cycle.

## 4.1.6 Mémoire de la CPU

### Gestion de la mémoire

La CPU fournit les zones de mémoire suivantes pour stocker le programme utilisateur, les données et la configuration :

- La mémoire de chargement est une mémoire rémanente pour le programme utilisateur, les données et la configuration. Lorsqu'un projet est chargé dans la CPU, il est d'abord stocké dans la mémoire de chargement. Cette zone se trouve soit sur une carte mémoire (en présence d'une telle carte) soit dans la CPU. Cette zone de mémoire rémanente est conservée en cas de coupure de courant. La carte mémoire offre plus d'espace de stockage que la mémoire de chargement intégrée dans la CPU.
- La mémoire de travail est une mémoire volatile pour certains éléments du projet utilisateur pendant l'exécution du programme utilisateur. La CPU copie certains éléments du projet de la mémoire de chargement dans la mémoire de travail. Cette zone de mémoire volatile est perdue en cas de coupure de courant et est restaurée par la CPU au retour de la tension.
- La mémoire rémanente constitue un stockage permanent pour une quantité limitée de valeurs de mémoire de travail. La zone de mémoire rémanente sert à stocker les valeurs d'adresses de mémoire utilisateur choisies pendant une coupure de courant. Si une mise hors tension ou une coupure de courant se produit, la CPU restaurera ces valeurs rémanentes à la mise sous tension suivante.

Pour afficher l'utilisation de la mémoire dans le projet en cours, cliquez avec le bouton droit de la souris sur la CPU (ou l'un de ses blocs) et sélectionnez "Ressources" dans le menu contextuel. Pour afficher l'utilisation de la mémoire dans la CPU en cours, double-cliquez sur "En ligne & Diagnostic", affichez le détail de "Diagnostic" et sélectionnez "Mémoire".



## Mémoire rémanente

Vous pouvez éviter la perte de données en cas de coupure de courant en définissant certaines données comme rémanentes. Les données suivantes peuvent être configurées comme rémanentes :

- **Mémentos (M) :** Vous pouvez définir la taille précise de la mémoire pour les mémentos dans la table des variables API ou dans le tableau d'affectation. Les mémentos rémanents commencent toujours à MB0 et comprennent un nombre indiqué d'octets consécutifs. Indiquez cette valeur dans la table des variables API ou dans le tableau d'affectation en cliquant sur le bouton "Rémanence" de la barre d'outils. Entrez le nombre d'octets de mémentos à définir comme rémanents à compter de MB0.
- **Variables d'un bloc fonctionnel :** Si un FB a été créé avec l'option "Optimisé" activée, l'éditeur d'interface pour ce FB comprend une colonne "Rémanence". Dans cette colonne, vous pouvez sélectionner "Rémanente", "Non rémanente" ou "Définir dans IDB" individuellement pour chaque variable. Le DB d'instance créé lors de l'insertion de ce FB dans l'éditeur de programmes comprend également cette colonne "Rémanence". Vous ne pouvez modifier l'état de rémanence d'une variable dans l'éditeur d'interface de DB d'instance que si vous avez sélectionné "Définir dans IDB" (définir dans le bloc de données d'instance) dans la colonne "Rémanence" de la variable concernée dans le FB optimisé.

Si un FB a été créé avec l'option "Standard - compatible avec S7-300/400" activée, l'éditeur d'interface pour ce FB ne comporte pas de colonne "Rémanence". Le DB d'instance créé lors du placement de ce FB dans l'éditeur de programme comprend une colonne "Rémanence" dont le contenu peut être modifié. Dans ce cas, sélectionner l'option "Rémanente" pour n'importe quelle variable entraîne son activation pour **toutes** les variables. De même, désactiver cette option pour n'importe quelle variable entraîne sa désactivation pour **toutes** les variables. Pour un FB configuré comme étant "Standard - compatible avec S7-300/400", vous pouvez modifier l'état de rémanence dans l'éditeur de DB d'instance, mais toutes les variables prennent le même état de rémanence.

Une fois le FB créé, vous ne pouvez pas activer a posteriori l'option "Standard - compatible avec S7-300/400". Vous ne pouvez sélectionner cette option que lors de la création du FB. Pour déterminer si un FB existant a été créé avec l'option "Optimisé" ou "Standard - compatible avec S7-300/400", cliquez avec le bouton droit de la souris sur le FB dans le navigateur du projet, sélectionnez "Propriétés", puis sélectionnez "Attributs". Si la case "Accès optimisé" est cochée, cela signifie que le bloc est optimisé. Sinon, il s'agit d'un FB standard compatible avec les CPU S7-300/400.

- **Variables d'un bloc de données global :** Le comportement d'un DB global quant à l'affectation de l'état de rémanence est similaire à celui d'un FB. Selon le paramétrage d'accès au bloc, vous pouvez définir l'état de rémanence soit pour des variables individuelles soit pour toutes les variables d'un bloc de données global.
  - Si vous avez sélectionné "Optimisé" lors de la création du DB, vous pouvez définir l'état de rémanence pour chaque variable individuelle.
  - Si vous avez sélectionné "Standard - compatible avec S7-300/400" lors de la création du DB, le paramétrage de l'état de rémanence s'applique à toutes les variables du DB : soit toutes les variables sont rémanentes, soit aucune variable n'est rémanente.

#### 4.1 Exécution du programme utilisateur


10240 octets de données au total peuvent être définis comme rémanents. Pour voir combien d'octets sont disponibles, cliquez sur le bouton "Rémanence" de la barre d'outils dans la table des variables API ou dans le tableau d'affectation. Les informations affichées mentionnent certes la plage de rémanence pour les mémentos, mais également, dans la deuxième ligne, la mémoire restante totale disponible pour les mémentos et les DB combinés. Notez que, pour que cette valeur soit exacte, vous devez compiler tous les blocs de données ayant des variables rémanentes.

##### 4.1.6.1 Mémento système et mémento de cadence

Vous vous servez des propriétés CPU pour activer les octets "mémento système" et "mémento de cadence". La logique de votre programme peut faire référence aux bits individuels de ces fonctions par leur nom de variable.

- Vous pouvez affecter un octet dans la zone de mémoire M au mémento système. L'octet de mémento système fournit les quatre bits suivants auxquels votre programme utilisateur peut faire référence par les noms de variables suivants :
  - Premier cycle (nom de variable "FirstScan") : Ce bit est mis à 1 pendant la durée du premier cycle une fois l'OB de démarrage terminé. Une fois l'exécution du premier cycle achevée, le bit "Premier cycle" est mis à 0.
  - Etat de diagnostic modifié (nom de variable "DiagStatusUpdate") : Ce bit est mis à 1 pendant un cycle après que la CPU a consigné un événement de diagnostic. Comme la CPU ne définit pas le bit "Diagramme de diagnostic modifié" avant la fin de la première exécution des OB de cycle de programme, votre programme utilisateur ne peut pas détecter s'il y a eu une modification du diagnostic pendant l'exécution des OB de démarrage ou la première exécution des OB de cycle de programme.
  - Toujours 1 (high) (nom de variable "AlwaysTRUE") : Ce bit est toujours à 1.
  - Toujours 0 (low) (nom de variable "AlwaysFALSE") : Ce bit est toujours à 0.
- Vous pouvez affecter un octet dans la zone de mémoire M au mémento de cadence. Chaque bit de l'octet configuré comme mémento de cadence génère une impulsion en signaux carrés. L'octet de mémento de cadence fournit 8 fréquences différentes, de 0,5 Hz (lent) à 10 Hz (rapide). Vous pouvez utiliser ces bits comme bits de commande, en particulier en combinaison avec des instructions sur front, pour déclencher des actions dans le programme utilisateur sur une base cyclique.

La CPU initialise ces octets lors du passage de l'état ARRET à l'état MISE EN ROUTE. Les bits du memento de cadence changent de manière synchrone avec l'horloge CPU aux états MISE EN ROUTE et MARCHÉ.

 <b>PRUDENCE</b>
<p>L'écrasement des bits du memento système ou du memento de cadence peut altérer les données dans ces fonctions et provoquer un fonctionnement incorrect du programme utilisateur, pouvant entraîner la mort et des blessures graves.</p> <p>Comme le memento de cadence et le memento système ne sont pas réservés en mémoire M, des instructions ou des tâches de communication peuvent écrire dans ces adresses et altérer les données.</p> <p>Évitez d'écrire des données dans ces adresses afin de garantir le bon fonctionnement de ces fonctions et prévoyez toujours un circuit d'arrêt d'urgence pour votre processus ou votre installation.</p>

La fonction memento système configure un octet dont les bits sont activés (valeur = 1) pour un événement spécifique.

**Bits de memento système**

Activer l'utilisation de l'octet de memento système

Adresse de l'octet de memento système (MBx) :

Premier cycle :

Diagramme de diagnostic modifié :

Toujours 1 (high) :

Toujours 0 (low) :

Tableau 4- 6 Memento système

7	6	5	4	3	2	1	0
Réservés Valeur 0				Toujours désactivé Valeur 0	Toujours activé Valeur 1	Indicateur Etat de diagnostic • 1: Modification • 0: Pas de modification	Indicateur Premier cycle • 1: Premier cycle après la mise en route • 0: Pas le premier cycle

La fonction memento de cadence configure un octet qui met les différents bits à 1 et à 0 à intervalles fixes. Chaque bit de cadence génère une impulsion en signaux carrés sur le bit M correspondant. Ces bits peuvent être utilisés comme bits de commande, en particulier en combinaison avec des instructions sur front, pour déclencher des actions dans le code utilisateur sur une base cyclique.

4.1 Exécution du programme utilisateur

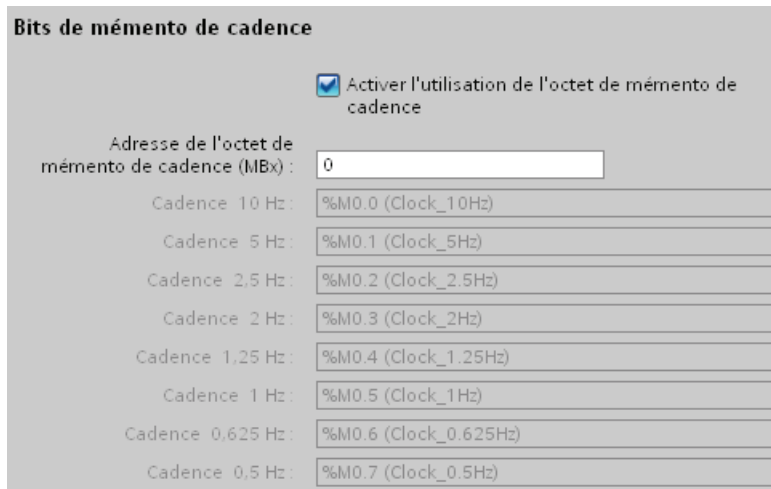


Tableau 4- 7 Memento de cadence

Numéro du bit	7	6	5	4	3	2	1	0
Nom de variable								
Période (s)	2.0	1.6	1.0	0.8	0.5	0.4	0.2	0.1
Fréquence (Hz)	0.5	0.625	1	1.25	2	2.5	5	10

Comme le memento de cadence opère de manière asynchrone par rapport au cycle de la CPU, son état peut changer plusieurs fois au cours d'un cycle long.

4.1.7 Mémoire tampon de diagnostic

La CPU fournit une mémoire tampon de diagnostic qui contient une entrée pour chaque événement de diagnostic. Chaque entrée inclut la date et l'heure auxquelles l'événement s'est produit, une catégorie d'événement, ainsi qu'une description de l'événement. Les entrées sont affichées par ordre chronologique, l'événement le plus récent se trouvant en haut. Les 50 événements les plus récents (maximum) sont disponibles dans ce journal. Lorsque le journal est plein, l'événement le plus ancien dans le journal est remplacé par un nouvel événement. En cas de coupure de courant, les événements sont sauvegardés.

Les types d'événements suivants sont enregistrés dans la mémoire tampon de diagnostic :

- Chaque événement de diagnostic système, par exemple les erreurs de CPU et les erreurs de module
- Chaque changement d'état de la CPU (chaque mise sous tension, chaque passage à ARRÊT, chaque passage à MARCHE)

Vous devez être en ligne pour accéder à la mémoire tampon de diagnostic (Page 728). Dans la vue "En ligne & Diagnostic, localisez la mémoire tampon de diagnostic sous "Diagnostic > Tampon de diagnostic".

### 4.1.8 Horloge temps réel

La CPU prend en charge une horloge temps réel. Un supercondensateur fournit l'énergie nécessaire pour que l'horloge continue à fonctionner lorsque la CPU est hors tension. Le supercondensateur se recharge lorsque la CPU est sous tension. Une mise sous tension de la CPU d'au moins 24 heures charge suffisamment le supercondensateur pour assurer le fonctionnement de l'horloge pendant 20 jours typiquement.

STEP 7 règle l'horloge temps réel sur l'heure système qui a une valeur par défaut au déballage ou suite à une réinitialisation aux réglages d'usine. Pour utiliser l'horloge temps réel, vous devez la régler. Les horodatages tels ceux pour les entrées de la mémoire de diagnostic, les journaux de données et les entrées de journaux se basent sur l'heure système. Vous réglez l'heure grâce à la fonction "Réglage de l'heure" (Page 726) dans la vue "En ligne & Diagnostic" de la CPU en ligne. STEP 7 calcule alors l'heure système à partir de l'heure que vous avez réglée plus ou moins le décalage du système d'exploitation Windows par rapport au temps universel coordonné (UTC). Régler l'heure sur l'heure locale en cours donne une heure système UTC si les paramétrages de votre système d'exploitation Windows pour le fuseau horaire et l'heure d'été correspondent à vos paramètres régionaux.

STEP 7 comporte des instructions (Page 265) pour lire et écrire l'heure système (RD\_SYS\_T et WR\_SYS\_T), pour lire l'heure locale (RD\_LOC\_T) et pour régler le fuseau horaire (SET\_TIMEZONE). L'instruction RD\_LOC\_T calcule l'heure locale à l'aide des décalages de fuseau horaire et d'heure d'été que vous paramétrez dans la configuration "Heure" des propriétés générales de la CPU (Page 131). Ces paramètres vous permettent de définir votre fuseau horaire pour l'heure locale, d'activer éventuellement l'heure d'été et d'indiquer les dates et heures de début et de fin de l'heure d'été. Vous pouvez également utiliser l'instruction SET\_TIMEZONE pour configurer ces paramètres.

### 4.1.9 Configuration des sorties lors d'un passage de MARCHE à ARRET

Vous pouvez configurer le comportement des sorties TOR et analogiques lorsque la CPU est à l'état ARRET. Vous pouvez paramétrer n'importe quelle sortie d'une CPU, d'un SB ou d'un SM de sorte que sa valeur est figée ou qu'une valeur de remplacement est utilisée :

- Remplacement par une valeur de sortie spécifique (par défaut) : Vous entrez une valeur de remplacement pour chaque sortie (voie) de cette CPU, ce SB ou ce SM.

La valeur de remplacement par défaut pour les voies de sortie TOR est "désactivé" et la valeur de remplacement par défaut pour les voies de sortie analogiques est 0.

- Maintenir les sorties à leur dernier état : Les sorties conservent la valeur qu'elles avaient lors du passage de MARCHE à ARRET. Après une mise sous tension, les sorties sont définies à la valeur de remplacement par défaut.

Vous configurez le comportement des sorties dans la Configuration d'appareil. Sélectionnez les différents appareils et utilisez l'onglet "Propriétés" pour configurer les sorties de chaque appareil.

Lorsque la CPU passe de MARCHE à ARRET, elle conserve la mémoire image et écrit les valeurs appropriées pour les sorties TOR et analogiques en fonction de la configuration.

## 4.2 Stockage des données, zones de mémoire, E/S et adressage

### 4.2.1 Accès aux données du S7-1200

STEP 7 facilite la programmation symbolique. Vous créez des noms symboliques ou "variables" pour les adresses des données, soit sous forme de variables API associées à des adresses mémoire et à des E/S, soit sous forme de variables locales utilisées dans un bloc de code. Pour utiliser ces variables dans votre programme utilisateur, il vous suffit d'entrer le nom de la variable comme paramètre de l'instruction.

Pour une meilleure compréhension de la manière dont la CPU organise et adresse les zones de mémoire, les paragraphes suivants expliquent l'adressage "absolu" référencé par les variables API. La CPU offre plusieurs possibilités pour stocker les données pendant l'exécution du programme utilisateur :

- **Mémoire globale :** La CPU fournit diverses zones de mémoire spécialisées, à savoir les entrées (I), les sorties (Q) et les mémentos (M). Cette mémoire est accessible à tous les blocs de code sans restriction.
- **Table de variables API :** Vous pouvez entrer des mnémoniques pour des adresses de mémoire spécifiques dans la table de variables API STEP 7. Ces variables sont globales pour le programme STEP 7 et permettent de programmer avec des noms qui ont une signification dans votre application.
- **Blocs de données (DB) :** Vous pouvez inclure des blocs de données (DB) dans votre programme utilisateur afin de sauvegarder les données des blocs de code. Les données sauvegardées sont conservées une fois l'exécution du bloc de code associé achevée. Un DB "global" contient des données pouvant être utilisées par tous les blocs de code alors qu'un DB d'instance contient les données d'un FB spécifique et a une structure correspondant aux paramètres du FB.
- **Mémoire temporaire :** Lors de l'appel d'un bloc de code, le système d'exploitation de la CPU alloue de la mémoire temporaire - ou locale (L) - utilisable pendant l'exécution de ce bloc. Lorsque l'exécution de ce bloc de code s'achève, la CPU réalloue la mémoire locale pour l'exécution d'autres blocs de code.

Chaque emplacement de mémoire différent a une adresse unique. Votre programme utilisateur utilise ces adresses pour accéder aux informations contenues dans l'emplacement de mémoire. Les références aux zones de mémoire d'entrée (I) ou de sortie (Q), telles que I0.3 ou Q1.7, permettent d'accéder à la mémoire image. Pour accéder directement à l'entrée ou à la sortie physique, ajoutez ":P" à la référence (par exemple, I0.3:P, Q1.7:P ou "Arret:P").

Tableau 4- 8 Zones de mémoire

Zone de mémoire	Description	Forçage permanent	Rémanence
I Mémoire image des entrées I_:P <sup>1</sup> (entrée physique)	Copiée des entrées physiques au début du cycle	Non	Non
	Lecture directe des entrées physiques sur la CPU, le SB et le SM	Oui	Non
Q	Copiée dans les sorties physiques au début du cycle	Non	Non

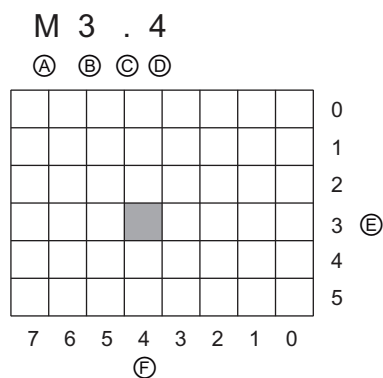
Zone de mémoire	Description	Forçage permanent	Rémanence
Mémoire image des sorties Q_:P <sup>1</sup> (sortie physique)	Ecriture directe dans les entrées physiques sur la CPU, le SB et le SM	Oui	Non
M Mémentos	Mémoire de commande et de données	Non	Oui (facultatif)
L Mémoire temporaire	Données temporaires pour un bloc, locales à ce bloc	Non	Non
DB Bloc de données	Mémoire de données ainsi que mémoire de paramètres pour les FB	Non	Oui (facultatif)

<sup>1</sup> Pour accéder directement (en lecture ou en écriture) aux entrées et sorties physiques, ajoutez ":P" à l'adresse ou à la variable (par exemple, I0.3:P, Q1.7:P ou "Arret:P").

Chaque emplacement de mémoire différent a une adresse unique. Votre programme utilisateur utilise ces adresses pour accéder aux informations contenues dans l'emplacement de mémoire. L'adresse absolue est constituée des éléments suivants :

- Identificateur de zone de mémoire (comme I, Q ou M)
- Taille des données en accès ("B" pour Byte, "W" pour Word ou "D" for DWord)
- Adresse de début des données (comme octet 3 ou mot 3)

Lors de l'accès à un bit dans l'adresse pour une valeur booléenne, vous ne précisez pas la taille. Vous saisissez uniquement la zone de mémoire, l'adresse d'octet et l'adresse de bit pour cette donnée (I0.0, Q0.1 ou M3.4, par exemple).



- |   |                                               |   |                              |
|---|-----------------------------------------------|---|------------------------------|
| A | Identificateur de zone de mémoire             | E | Octets de la zone de mémoire |
| B | Adresse d'octet : octet 3                     | F | Bits de l'octet sélectionné  |
| C | Séparateur ("octet.bit")                      |   |                              |
| D | Emplacement du bit dans l'octet (bit 4 sur 8) |   |                              |

Dans l'exemple, la zone de mémoire et l'adresse d'octet (M = zone des mémentos et 3 = Byte 3) sont suivies par un point (".") pour séparer l'adresse du bit (bit 4).

### Accès aux données dans les zones de mémoire de la CPU

STEP 7 facilite la programmation symbolique. Typiquement, des variables sont créées dans les variables API, dans un bloc de données ou dans l'interface en haut d'un OB, d'une FC ou d'un FB. Ces variables comprennent un nom, un type de données, un décalage et un commentaire. Une valeur initiale peut en outre être indiquée dans un bloc de données. Vous pouvez utiliser ces variables lors de la programmation en entrant le nom de variable au niveau du paramètre de l'instruction. Vous pouvez optionnellement entrer l'opérande absolu (zone de mémoire, taille et décalage) au niveau du paramètre de l'instruction. Les exemples dans les paragraphes suivants montrent comment entrer des opérandes absolus. Le caractère % est inséré automatiquement devant l'opérande absolu par l'éditeur de programmes. Vous pouvez activer divers modes d'affichage dans l'éditeur de programmes : affichage symbolique, affichage symbolique et absolu ou affichage absolu.

**I (mémoire image des entrées) :** La CPU lit les entrées de périphérie (entrées physiques) juste avant l'exécution de l'OB cyclique de chaque cycle et écrit ces valeurs dans la mémoire image des entrées. Vous pouvez accéder à la mémoire image des entrées par bits, octets, mots ou doubles mots. L'accès en lecture et l'accès en écriture sont tous deux autorisés mais, typiquement, les entrées de la mémoire image sont uniquement lues.

Tableau 4- 9 Adressage absolu pour la mémoire I

Bit	I[adresse d'octet].[adresse de bit]	I0.1
Octet, mot ou double mot	I[taille][adresse d'octet de départ]	IB4, IW5 ou ID12

En ajoutant un ":P" à l'adresse, vous pouvez lire directement les entrées TOR et analogiques de la CPU, du SB ou du SM. La différence entre un accès via I\_:P et un accès via I est que les données proviennent directement des entrées en accès et non de la mémoire image des entrées. On parle d'accès en "lecture directe" pour cet accès I\_:P, car les données sont extraites directement de la source et non d'une copie qui a été faite lors de la dernière actualisation de la mémoire image des entrées.

Comme les entrées physiques reçoivent leur valeur directement des appareils de terrain connectés à ces entrées, écrire dans ces entrées est interdit. Ainsi, les accès I\_:P sont en lecture seule alors que les accès I peuvent être en lecture ou en écriture.

Les accès I\_:P sont également limités à la taille des entrées prises en charge par une CPU, un SB ou un SM unique, arrondie à l'octet le plus proche. Par exemple, si les entrées d'un SB 2 DI / 2 DQ sont configurées pour commencer en I4.0, il est possible d'y accéder en tant que I4.0:P et I4.1:P ou en tant que IB4:P. Les accès de I4.2:P à I4.7:P ne sont pas refusés mais n'ont aucun sens puisque les entrées correspondantes ne sont pas utilisées. Les accès à IW4:P et ID4:P sont interdits puisqu'ils dépassent le décalage d'octet associé au SB.

Les accès via I\_:P n'affectent pas la valeur correspondante sauvegardée dans la mémoire image des entrées.

Tableau 4- 10 Adressage absolu pour la mémoire I (accès direct)

Bit	I[adresse d'octet].[adresse de bit]:P	I0.1:P
Octet, mot ou double mot	I[taille][adresse d'octet de départ]:P	IB4:P, IW5:P ou ID12:P

**Q (mémoire image des sorties) :** La CPU copie dans les sorties physiques les valeurs sauvegardées dans la mémoire image des sorties. Vous pouvez accéder à la mémoire image des sorties par bits, octets, mots ou doubles mots. L'accès en lecture et l'accès en écriture sont tous deux autorisés pour les sorties de la mémoire image.



Tableau 4- 11 Adressage absolu pour la mémoire Q

Bit	Q[adresse d'octet].[adresse de bit]	Q1.1
Octet, mot ou double mot	Q[taille][adresse d'octet de départ]	QB5, QW10, QD40

En ajoutant un ":P" à l'adresse, vous pouvez écrire directement dans les sorties TOR et analogiques physiques de la CPU, du SB ou du SM. La différence entre un accès via Q\_:P et un accès via Q est que les données sont directement écrites dans les sorties en accès en plus d'être écrites dans la mémoire image des sorties. On parle parfois d'accès en "écriture directe" pour cet accès Q\_:P, car les données sont envoyées directement à leur destination qui n'a pas besoin d'attendre la prochaine actualisation à partir de la mémoire image des sorties.

Comme les sorties physiques pilotent directement les appareils de terrain qui y sont connectés, la lecture de ces sorties est interdite. Ainsi, les accès Q\_:P sont en écriture seule alors que les accès Q peuvent être en lecture ou en écriture.

Les accès Q\_:P sont également limités à la taille des sorties prises en charge par une CPU, un SB ou un SM unique, arrondie à l'octet le plus proche. Par exemple, si les sorties d'un SB 2 DI / 2 DQ sont configurées pour commencer en Q4.0, il est possible d'y accéder en tant que Q4.0:P et Q4.1:P ou en tant que QB4:P. Les accès de Q4.2:P à Q4.7:P ne sont pas refusés mais n'ont aucun sens puisque les sorties correspondantes ne sont pas utilisées. Les accès à QW4:P et QD4:P sont interdits puisqu'ils dépassent le décalage d'octet associé au SB.

Les accès via Q\_:P affectent à la fois la sortie physique et la valeur correspondante sauvegardée dans la mémoire image des sorties.

Tableau 4- 12 Adressage absolu pour la mémoire Q (accès direct)

Bit	Q[adresse d'octet].[adresse de bit]:P	Q1.1:P
Octet, mot ou double mot	Q[taille][adresse d'octet de départ]:P	QB5:P, QW10:P ou QD40:P

**M (mémentos) :** Utilisez la zone des mémentos (mémoire M) pour les relais de commande et les données afin de stocker l'état intermédiaire d'une opération ou d'autres informations de commande. Vous pouvez accéder à la zone des mémentos par bits, octets, mots ou doubles mots. L'accès en lecture et l'accès en écriture sont tous deux autorisés pour la mémoire M.

Tableau 4- 13 Adressage absolu pour la mémoire M

Bit	M[adresse d'octet].[adresse de bit]	M26.7
Octet, mot ou double mot	M[taille][adresse d'octet de départ]	MB20, MW30, MD50

**Temp (mémoire temporaire) :** La CPU affecte la mémoire temporaire selon les besoins. La CPU alloue la mémoire temporaire au bloc de code au moment où le bloc de code est démarré (pour un OB) ou est appelé (pour une FC ou un FB). L'affectation de mémoire temporaire à un bloc de code peut entraîner la réutilisation d'adresses de mémoire temporaire utilisées précédemment par d'autres OB, FC ou FB. La mémoire temporaire n'est pas initialisée par la CPU lors de son affectation et elle peut donc contenir n'importe quelle valeur.

4.2 Stockage des données, zones de mémoire, E/S et adressage

La mémoire temporaire est similaire à la mémoire M à une exception majeure près. En effet, la mémoire M a une portée globale alors que la mémoire temporaire a une portée locale.

- Mémoire M : Tout OB, FC ou FB peut accéder aux données dans la mémoire M, ce qui signifie que les données sont globalement disponibles pour tous les éléments du programme utilisateur.
- Mémoire temporaire : L'accès aux données en mémoire temporaire est limité à l'OB, à la FC ou au FB qui a créé ou déclaré l'adresse de mémoire temporaire. Les adresses de mémoire temporaire restent locales et ne sont pas partagées par différents blocs de code même lorsque le bloc de code appelle un autre bloc de code. Par exemple, lorsqu'un OB appelle une FC, la FC ne peut pas accéder à la mémoire temporaire de l'OB qui l'a appelée.

La CPU fournit de la mémoire temporaire (locale) pour chacune des trois classes de priorité d'OB :

- 16 Ko pour le démarrage et le cycle du programme, FB et FC associés inclus
- 4 Ko pour les événements d'alarme standard, FB et FC inclus
- 4 Ko pour les événements d'alarme d'erreur, FB et FC inclus

Vous accédez à la mémoire temporaire par adressage symbolique uniquement.

**DB (bloc de données) :** Utilisez les blocs de données pour sauvegarder divers types de données, notamment l'état intermédiaire d'une opération ou d'autres informations de commande, les paramètres de FB et des structures de données nécessaires pour de nombreuses instructions telles que temporisations et compteurs. Vous pouvez accéder aux blocs de données par bits, octets, mots ou doubles mots. L'accès en lecture et l'accès en écriture sont tous deux autorisés pour les blocs de données en lecture/écriture. Seul l'accès en lecture est autorisé pour les blocs de données en lecture seule.

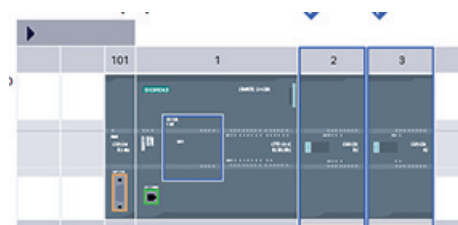
Tableau 4- 14 Adressage absolu pour la mémoire DB

Bit	DB[numéro de bloc de données].DBX[adresse d'octet].[adresse de bit]	DB1.DBX2.3
Octet, mot ou double mot	DB[numéro de bloc de données].DB[taille][adresse d'octet de départ]	DB1.DBB4, DB10.DBW2, DB20.DBBD8

**Remarque**

Lorsque vous indiquez une adresse absolue, STEP 7 insère le caractère "%" devant cette adresse pour signaler qu'il s'agit d'une adresse absolue. Lors de la programmation, vous pouvez saisir une adresse absolue soit avec, soit sans le caractère "%" (par exemple, %I0.0 ou I.0). Si vous omettez le caractère "%", STEP 7 l'ajoutera.

## Configuration des E/S dans la CPU et les modules E/S



Lorsque vous ajoutez une CPU et des modules d'E/S dans votre écran de configuration, les adresses I et Q sont automatiquement affectées. Vous pouvez modifier l'adressage par défaut en sélectionnant le champ d'adresse dans l'écran de configuration et en saisissant de nouveaux nombres.

Vue d'ensemble des appareils					
Module	Empla.	Adresse E	Adresse S	Type	N° de r
	103				
	102				
RS485_1	101			CM 1241 (RS485)	6ES7
PLC_1	1			CPU 1214C DCDC	6ES7
DI14/DO10	1.1	0...1	0...1	DI14/DO10	
AI2	1.2	64...67		AI2	
AO1 x 12Bi	1.3		80...81	Signal Board AO1	6ES7
HSC_1	1.16	1000.....		Compteur rapide (.)	
HSC_2	1.17			Compteur rapide (.)	
HSC_3	1.18			Compteur rapide (.)	
HSC_4	1.19			Compteur rapide (.)	
HSC_5	1.20			Compteur rapide (.)	
HSC_6	1.21			Compteur rapide (.)	
Pulse_1	1.32			Générateur d'impu.	
Pulse_2	1.33			Générateur d'impu.	
Interface P. X1				Interface PROFINET	
DIB x DC24V.	2	8		SM 1221 DIB x DC.	6ES7

- Les entrées et sorties numériques sont attribuées dans des groupes de 8 points (1 octet), que le module utilise tous les points ou non.
- Les entrées et sorties analogiques sont affectées par groupes de deux entrées ou deux sorties (4 octets).

La figure montre un exemple de CPU 1214C avec deux SM et un SB. Dans cet exemple, vous pourriez modifier l'adresse du module DI8 à 2 au lieu de 8. L'outil vous aidera en modifiant les plages d'adresse qui en sont pas de la bonne dimension ou qui sont en conflit avec d'autres adresses.

## 4.3 Traitement des valeurs analogiques

Les modules d'entrées-sorties analogiques fournissent des signaux d'entrée ou attendent des valeurs de sortie qui représentent soit une plage de tension, soit une plage de courant. Il s'agit des plages  $\pm 10$  V,  $\pm 5$  V,  $\pm 2,5$  V ou 0 à 20 mA. Les valeurs renvoyées par les modules sont des valeurs entières, 0 à 27648 représentant la plage nominale pour le courant et -27648 à 27648 la plage nominale pour la tension. Toute valeur en dehors de ces plages représente soit un débordement haut, soit un débordement bas. Reportez-vous aux tableaux sur la représentation des entrées analogiques (Page 822) et la représentation des sorties analogiques (Page 823) pour plus de détails.

Dans votre programme de commande, vous aurez probablement besoin de ces valeurs en unités de mesure physiques, par exemple pour représenter un volume, une température, un poids ou une autre valeur de quantité. S'il s'agit d'une entrée analogique, vous devez d'abord normaliser la valeur analogique à une valeur réelle (à virgule flottante) comprise entre 0,0 et 1,0. Vous devez ensuite la mettre à l'échelle des valeurs minimum et maximum des unités physiques qu'elle représente. Dans le cas de valeurs en unités physiques que vous avez besoin de convertir en valeurs de sortie analogiques, vous devez d'abord normaliser la valeur en unités physiques à une valeur comprise entre 0,0 et 1,0, puis la mettre à l'échelle entre 0 et 27648 ou -27648 et 27648 en fonction de la plage du module analogique. STEP 7 fournit les instructions NORM\_X et SCALE\_X (Page 233) pour réaliser ces opérations. Vous pouvez également utiliser l'instruction CALCULATE (Page 211) pour mettre les valeurs analogiques (Page 33) à l'échelle.

## 4.4 Types de données

Les types de données servent à indiquer à la fois la taille d'un élément de données et la façon dont les données doivent être interprétées. Chaque paramètre d'une instruction accepte au moins un type de données et certains paramètres acceptent plusieurs types de données. Maintenez le curseur au-dessus du champ de paramètre d'une instruction pour voir les types de données acceptés par un paramètre particulier.

Un paramètre formel est un identificateur qui repère l'emplacement des données qu'une instruction doit utiliser (par exemple, l'entrée IN1 d'une instruction ADD). Un paramètre effectif est l'adresse de mémoire (précédée d'un caractère "%") ou la constante contenant les données que l'instruction doit utiliser (exemple : %MD400 "nombre\_de\_widgets"). Le type de données du paramètre effectif que vous indiquez doit correspondre à l'un des types de données acceptés par le paramètre formel indiqué par l'instruction.

Lorsque vous indiquez un paramètre effectif, vous devez indiquer soit une variable (mnémonique), soit une adresse de mémoire (directe) absolue. Les variables sont associées à un mnémonique (nom de la variable) avec un type de données, une zone de mémoire, un décalage en mémoire et un commentaire et elles peuvent être créées soit dans l'éditeur de variables API, soit dans l'éditeur d'interface d'un bloc (OB, FC, FB et DB). Si vous entrez une adresse absolue n'ayant pas de variable associée, vous devez utiliser une taille appropriée qui correspond à un type de données accepté et une variable par défaut sera créée lors de l'entrée.

Tous ces types de données excepté STRING sont disponibles à la fois dans l'éditeur de variables API et dans les éditeurs d'interface de bloc. STRING est disponible uniquement dans les éditeurs d'interface de bloc. Vous pouvez également entrer une valeur constante pour de nombreux paramètres d'entrée.

- Bit et séquences de bits (Page 101) : Bool (valeur booléenne ou binaire), Byte (octet de 8 bits), Word (mots de 16 bits), DWord (double mot de 32 bits)
- Entier (Page 102)
  - USInt (entier de 8 bits non signé), SInt (entier de 8 bits signé)
  - UInt (entier de 16 bits non signé), Int (entier de 16 bits signé)
  - UInt (entier de 32 bits non signé), DInt (entier de 32 bits signé)
- Réel à virgule flottante (Page 103) : Real (valeur réelle ou virgule flottante de 32 bits), LReal (valeur réelle ou virgule flottante de 64 bits)

- Date et heure (Page 103) : Time (durée CEI de 32 bits), Date (date de 16 bits), TOD (heure de 32 bits), DT (date et heure de 64 bits)
- Caractère et chaîne de caractères (Page 105) : Char (caractère individuel de 8 bits), String (chaîne de longueur variable allant jusqu'à 254 caractères)
- Tableau (Page 107)
- Structure de données (Page 108) : Struct
- Type de données API (Page 108)
- Pointeurs (Page 109) : Pointer, Any, Variant

Bien que non disponibles en tant que types de données, les formats numériques BCD suivants sont pris en charge par les instructions de conversion.

Tableau 4- 15 Taille et plage du format DCB

Format	Taille (bits)	Plage numérique	Exemples d'entrée de constantes
BCD16	16	-999 à 999	123, -123
BCD32	32	-9999999 à 9999999	1234567, -1234567

#### 4.4.1 Types de données Bool, Byte, Word et DWord

Tableau 4- 16 Types de données pour bit et séquence de bits

Type de données	Taille en bits	Type de nombre	Plage de nombres	Exemples de constantes	Exemples d'adresses
Bool	1	Booléen	FALSE ou TRUE	TRUE, 1	I1.0 Q0.1 M50.7 DB1.DBX2.3 nom_variable
		Binaire	0 ou 1	0, 2#0	
		Octal	8#0 ou 8#1	8#1	
		Hexadécimal	16#0 ou 16#1	16#1	
Byte	8	Binaire	2#0 à 2#11111111	2#00001111	IB2 MB10 DB1.DBB4 nom_variable
		Entier non signé	0 à 255	15	
		Octal	8#0 à 8#377	8#17	
		Hexadécimal	B#16#0 à B#16#FF	B#16#F, 16#F	
Word	16	Binaire	2#0 à 2#1111111111111111	2#1111000011110000	MW10 DB1.DBW2 nom_variable
		Entier non signé	0 à 65535	61680	
		Octal	8#0 à 8#177777	8#170360	
		Hexadécimal	W#16#0 à W#16#FFFF, 16#0 à 16#FFFF	W#16#F0F0, 16#F0F0	
DWord	32	Binaire	2#0 à 2#11111111111111111111111111111111 11111111	2#111100001111111100 001111	MD10 DB1.DBD8 nom_variable
		Entier non signé	0 à 4294967295	15793935	
		Octal	8#0 à 8#3777777777	8#74177417	

4.4 Types de données

Type de données	Taille en bits	Type de nombre	Plage de nombres	Exemples de constantes	Exemples d'adresses
		Hexadécimal	DW#16#0000_0000 à DW#16#FFFF_FFFF, 16#0000_0000 à 16#FFFF_FFFF	DW#16#F0FF0F, 16#F0FF0F	

4.4.2 Types de données entiers

Tableau 4- 17 Types de données entiers (U = non signé, S = court, D= double)

Type de données	Taille en bits	Plage de nombres	Exemples de constantes	Exemples d'adresses
USInt	8	0 à 255	78, 2#01001110	MB0, DB1.DBB4, nom_variable
SInt	8	-128 à 127	+50, 16#50	
UInt	16	0 à 65 535	65295, 0	MW2, DB1.DBW2, nom_variable
Int	16	-32 768 à 32 767	30000, +30000	
UDInt	32	0 à 4 294 967 295	4042322160	MD6, DB1.DBD8, nom_variable
DInt	32	-2 147 483 648 à 2 147 483 647	-2131754992	

### 4.4.3 Types de données réels à virgule flottante

Les nombres réels (ou nombres à virgule flottante) sont représentés sous forme de nombres de 32 bits à simple précision (type Real) ou de nombres de 64 bits à double précision (type LReal) comme décrit dans la norme ANSI/IEEE 754-1985. Les nombres à virgule flottante simple précision sont précis jusqu'à 6 chiffres significatifs et les nombres à virgule flottante double précision sont précis jusqu'à 15 chiffres significatifs. Vous pouvez donc indiquer 6 (Real) et 15 (LReal) chiffres significatifs au maximum lorsque vous entrez une constante à virgule flottante afin de conserver la précision.

Tableau 4- 18 Types de données réels à virgule flottante (L=long)

Type de données	Taille en bits	Plage de nombres	Exemples de constantes	Exemples d'adresses
Real	32	-3,402823e+38 à -1,175 495e-38, ±0, +1,175 495e-38 à +3,402823e+38	123.456, -3.4, 1.0e-5	MD100, DB1.DB8, nom_variable
LReal	64	-1,7976931348623158e+308 à -2,2250738585072014e-308, ±0, +2,2250738585072014e-308 à +1,7976931348623158e+308	12345,123456789e40, 1,2E+40	nom_DB.nom_var Règles : <ul style="list-style-type: none"> <li>• L'adressage direct n'est pas pris en charge.</li> <li>• Peut être défini dans une table d'interface de bloc OB, FB ou FC</li> </ul>

Les calculs impliquant une longue série de valeurs comprenant des nombres très grands et très petits peuvent produire des résultats qui manquent de précision. Cela peut arriver si ces nombres diffèrent de 10 puissance x, x étant supérieur à 6 (Real) ou à 15 (LReal). Exemple pour Real : 100 000 000 + 1 = 100 000 000.

### 4.4.4 Types de données "date et heure"

Tableau 4- 19 Types de données "date et heure"

Type de données	Taille	Plage	Exemples d'entrée de constantes
TIME	32 bits	T#-24d_20h_31m_23s_648ms à T#24d_20h_31m_23s_647ms Sauvegardé en tant que : -2 147 483 648 ms à +2 147 483 647 ms	T#5m_30s T#1d_2h_15m_30s_45ms TIME#10d20h30m20s630ms 500h10000ms 10d20h30m20s630ms
DATE	16 bits	D#1990-1-1 à D#2168-12-31	D#2009-12-31 DATE#2009-12-31 2009-12-31

4.4 Types de données

Type de données	Taille	Plage	Exemples d'entrée de constantes
TIME_OF_DAY	32 bits	TOD#0:0:0.0 à TOD#23:59:59.999	TOD#10:20:30.400 TIME_OF_DAY#10:20:30.400 23:10:1
DTL (date et heure long)	12 octets	Min. : DTL#1970-01-01-00:00:00.0 Max. : DTL#2554-12-31-23:59:59.999 999 999	DTL#2008-12-16-20:30:20.250

**TIME**

Les données TIME sont sauvegardées sous forme d'entier double signé interprété en tant que millisecondes. Le format d'éditeur peut utiliser des informations pour le jour (d), les heures (h), les minutes (m), les secondes (s) et les millisecondes (ms).

Il n'est pas nécessaire de préciser toutes les unités de durée. T#5h10s et 500h sont, par exemple, valables.

La valeur combinée de toutes les valeurs d'unité indiquées ne doit pas dépasser les limites supérieure et inférieure en millisecondes pour le type de données TIME (-2 147 483 648 ms à +2 147 483 647 ms).

**DATE**

Les données DATE sont sauvegardées sous forme d'entier non signé interprété en tant que nombre de jours ajoutés à la date de base, 01/01/1990, pour obtenir la date précisée. Le format d'éditeur doit indiquer une année, un mois et un jour.

**TOD**

Les données TOD (TIME\_OF\_DAY) sont sauvegardées sous forme d'entier double non signé interprété en tant que nombre de millisecondes depuis minuit pour l'heure indiquée (minuit = 0 ms). Les heures (24 heures/jour), les minutes et les secondes doivent être indiquées. L'indication des fractions de seconde est facultative.

**DTL**

Le type de données DTL (date et heure long) utilise une structure de 12 octets qui mémorise les informations sur la date et l'heure. Vous pouvez définir des données DTL soit dans la mémoire temporaire d'un bloc, soit dans un DB. Il faut entrer une valeur pour chaque élément dans la colonne Valeur initiale de l'éditeur de DB.

Tableau 4- 20 Taille et plage pour DTL

Longueur (octets)	Format	Plage de valeurs	Exemple d'entrée de valeur
12	Horloge et calendrier année-mois-jour:heures:minutes: secondes.nanosecondes	Min. : DTL#1970-01-01-00:00:00.0 Max. : DTL#2554-12-31-23:59:59.999 999 999	DTL#2008-12-16-20:30:20.250



Chaque élément de la structure DTL contient un type de données et une plage de valeurs différents. Le type de données d'une valeur indiquée doit correspondre au type de données des éléments correspondants.

Tableau 4- 21 Eléments de la structure DTL

Octet	Elément	Type de données	Plage de valeurs
0	Année	UINT	1970 à 2554
1			
2	Mois	USINT	1 à 12
3	Jour	USINT	1 à 31
4	Jour de la semaine <sup>1</sup>	USINT	1 (dimanche) à 7 (samedi) <sup>1</sup>
5	Heures	USINT	0 à 23
6	Minutes	USINT	0 à 59
7	Secondes	USINT	0 à 59
8	Nanosecondes	UDINT	0 à 999 999 999
9			
10			
11			

<sup>1</sup> Le jour de la semaine n'est pas pris en compte dans l'entrée de la valeur.

#### 4.4.5 Types de données "caractère et chaîne de caractères"

Tableau 4- 22 Types de données "caractère et chaîne de caractères"

Type de données	Taille	Plage	Exemples d'entrée de constantes
Char	8 bits	Codes de caractères ASCII : 16#00 à 16#FF	'A', 't', '@'
String	n+ 2 octets	n = (0 à 254 octets caractères)	'ABC'

### CHAR

Les données CHAR occupent un octet en mémoire et contiennent un caractère unique codé en format ASCII. La syntaxe de l'éditeur utilise une apostrophe avant et après le caractère ASCII. Il est possible d'utiliser des caractères visibles et des caractères de commande. Un tableau de caractères de commande autorisés est donné dans la description du type de données String.

## STRING

La CPU prend en charge le type de données String pour sauvegarder une séquence de caractères d'un octet. Le type de données String contient le nombre de caractères total (nombre de caractères dans la chaîne) et le nombre de caractères en cours. Le type String fournit jusqu'à 256 octets pour le stockage du nombre de caractères total maximum (1 octet), du nombre de caractères en cours (1 octet) et de 254 caractères au maximum dont chacun est stocké dans 1 octet.

Vous pouvez utiliser des chaînes littérales (constantes) pour les paramètres d'instruction de type IN avec des apostrophes. Par exemple, 'ABC' est une chaîne de trois caractères qui pourrait être utilisée en tant qu'entrée pour le paramètre IN de l'instruction S\_CONV. Vous pouvez également créer des variables chaînes de caractères en sélectionnant le type de données STRING dans les éditeurs d'interface des blocs OB, FC, FB et DB. Vous ne pouvez pas créer de chaîne dans l'éditeur de variables API.

Vous pouvez indiquer la taille maximale d'une chaîne en octets à l'aide de crochets après le mot-clé "String" (après sélection du type de données "String" dans une liste déroulante de types de données). "MaChaîne String[10]" indiquerait par exemple une taille maximale de 10 octets pour MaChaîne. Si vous ne mentionnez pas de taille maximale entre crochets, 254 est pris par défaut.

L'exemple suivant définit une chaîne ayant un nombre maximum de caractères égal à 10 et un nombre de caractères en cours égal à 3. Cela signifie que la chaîne contient actuellement 3 caractères à un octet mais pourrait être allongée jusqu'à contenir 10 caractères à un octet.

Tableau 4- 23 Exemple de type de données String

Nombre total de caractères	Nombre de caractères en cours	Caractère 1	Caractère 2	Caractère 3	...	Caractère 10
10	3	'C' (16#43)	'A' (16#41)	'T' (16#54)	...	-
Octet 0	Octet 1	Octet 2	Octet 3	Octet 4	...	Octet 11

Des caractères de commande ASCII peuvent être utilisés dans des données de type Char et String. Le tableau suivant présente des exemples de syntaxe avec caractères de commande.

Tableau 4- 24 Caractères de commande ASCII autorisés

Caractères de commande	Valeur ASCII hexa	Fonction de commande	Exemples
\$L ou \$l	0A	Changement de ligne	'\$LTexte', '\$0ATexte'
\$N ou \$n	0A et 0D	Saut de ligne La nouvelle ligne montre deux caractères dans la chaîne.	'\$NTexte', '\$0A\$0DTexte'
\$P ou \$p	0C	Changement de page	'\$PTexte', '\$0CTexte'
\$R ou \$r	0D	Retour chariot	'\$RTexte', '\$0DTexte'
\$T ou \$t	09	Tabulation	'\$TTexte', '\$09Texte'
\$\$	24	Signe dollar	'100\$\$', '100\$24'
\$'	27	Apostrophe	'\$'Texte\$', '\$27Texte\$27'

## 4.4.6 Type de données "tableau"

### Tableaux

Vous pouvez créer un tableau qui contient plusieurs éléments de même type de données. Il est possible de créer des tableaux dans les éditeurs d'interface des blocs OB, FC, FB et DB. Vous ne pouvez pas créer de tableau dans l'éditeur de variables API.

Pour créer un tableau dans l'éditeur d'interface de bloc, donnez un nom au tableau et choisissez le type de données "Array [lo .. hi] of type", puis éditez "lo", "hi" et "type" de la manière suivante :

- lo : indice de départ (le plus bas) pour votre tableau
- hi : indice de fin (le plus élevé) pour votre tableau
- type : un type de données, tels que BOOL, SINT, UDINT

Tableau 4- 25 Règles pour le type de données ARRAY

Type de données	Syntaxe d'un tableau		
ARRAY	nom [indice1_min..indice1_max, indice2_min..indice2_max] of <type de données>		
	<ul style="list-style-type: none"> <li>• Tous les éléments du tableau doivent avoir le même type de données.</li> <li>• L'indice peut être négatif, mais la limite inférieure doit être inférieure ou égale à la limite supérieure.</li> <li>• Un tableau peut avoir de une à six dimensions.</li> <li>• Les déclarations des indices min..max des différentes dimensions sont séparées par des virgules.</li> <li>• Les tableaux imbriqués (tableaux de tableaux) ne sont pas autorisés.</li> <li>• La taille d'un tableau en mémoire est égale à (taille d'un élément * nombre total d'éléments dans le tableau)</li> </ul>		
	Indice de tableau	Types de données autorisés pour les indices	Règles pour les indices de tableau
Constante ou variable	USInt, SInt, UInt, Int, UDInt, DInt	<ul style="list-style-type: none"> <li>• Valeurs limites : -32768 à +32767</li> <li>• Correct : constantes et variables mélangées</li> <li>• Correct : expressions constantes</li> <li>• Incorrect : expressions variables</li> </ul>	

**Exemple :  
déclarations de  
tableaux**

ARRAY[1..20] of REAL  
 ARRAY[-5..5] of INT  
 ARRAY[1..2, 3..4] of CHAR

Une dimension, 20 éléments  
 Une dimension, 11 éléments  
 Deux dimensions, 4 éléments

**Exemple : adresses  
de tableaux**

ARRAY1[0]  
 ARRAY2[1,2]  
 ARRAY3[i,j]

Tableau ARRAY1, élément 0  
 Tableau ARRAY2, élément [1,2]  
 Si i=3 et j=4, on adresse l'élément [3, 4] du tableau ARRAY3.

#### 4.4.7 Type de données "structure de données"

Vous pouvez utiliser le type de données "Struct" pour définir une structure de données constituée de types de données différents. Le type de données Struct peut servir à gérer un groupe de données de processus apparentées en tant qu'unité de données unique. Vous donnez un nom à la structure et vous déclarez son organisation de données interne dans l'éditeur de bloc de données ou dans un éditeur d'interface de bloc.

Vous pouvez regrouper des tableaux et des structures dans une structure plus grande. Une structure peut être imbriquée jusqu'à huit niveaux de profondeur. Par exemple, vous pouvez créer une structure de structures qui contiennent des tableaux.

Une variable Struct commence à une adresse d'octet paire et utilise la mémoire jusqu'à la frontière de mot suivante.

#### 4.4.8 Type de données API

L'éditeur de type de données API vous permet de définir des structures de données utilisables à plusieurs reprises dans votre programme. Pour créer un type de données API, ouvrez la branche "Types de données API" de l'arborescence de projet et double-cliquez sur "Ajouter nouveau type de données". Dans le type de données API nouvellement créé, servez-vous de deux clics simples pour modifier le nom par défaut et d'un double clic pour ouvrir l'éditeur de type de données API.

Pour créer une structure de type de données API personnalisée, vous utilisez les mêmes méthodes d'édition que dans l'éditeur de bloc de données. Ajoutez de nouvelles lignes pour tout type de données dont vous avez besoin pour créer la structure de données voulue.

Si un nouveau type de données API est créé, son nom apparaît dans les listes déroulantes de sélection de type de données dans l'éditeur de DB et dans l'éditeur d'interface de bloc de code.

Usages potentiels des types de données API :

- Vous pouvez utiliser les types de données API directement comme types de données dans une interface de bloc de code ou dans des blocs de données.
- Vous pouvez utiliser les types de données API comme modèles pour la création de plusieurs blocs de données globaux utilisant la même structure de données.

Un type de données API pourrait, par exemple, être une recette pour mélanger des couleurs. Vous pouvez alors affecter ce type de données API à plusieurs blocs de données, les variables de chaque bloc de données étant définies de manière à créer une couleur spécifique.

#### 4.4.9 Types de données "pointeur"

Les types de données "pointeur" (Pointer, Any et Variant) sont utilisables dans les tables d'interface des blocs de code FB et FC. Vous pouvez sélectionner un type de données "pointeur" dans les listes déroulantes de types de données d'une interface de bloc.

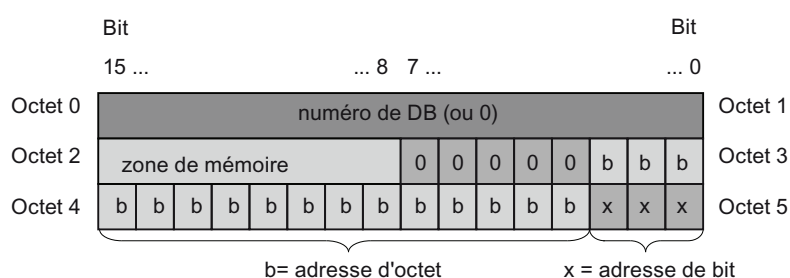
Le type de données Variant peut également être utilisé pour des paramètres d'instructions.

### 4.4.9.1 Type de données Pointer

Le type de données Pointer désigne une variable particulière. Il occupe 6 octets (48 bits) en mémoire et peut inclure les informations suivantes :

- numéro de DB ou 0 si les données ne sont pas sauvegardées dans un DB
- zone de mémoire dans la CPU
- adresse d'une variable

Format de pointeur



Vous pouvez déclarer les trois types de pointeurs suivants selon l'instruction :

- Pointeur intrazone : contient des données sur l'adresse d'une variable
- Pointeur interzone : contient des données sur la zone de mémoire et l'adresse d'une variable
- Pointeur de DB : contient un numéro de bloc de données et l'adresse d'une variable

Tableau 4- 26 Types de pointeurs :

Type	Format	Exemple de saisie
Pointeur intrazone	P#Octet.Bit	P#20.0
Pointeur interzone	P#Zone_mémoire_Octet.Bit	P#M20.0
Pointeur de DB	P#Bloc_données.Elément_données	P#DB10.DBX20.0

Vous pouvez entrer un paramètre de type Pointer sans le préfixe (P #). Votre entrée sera automatiquement convertie au format pointeur.

Tableau 4- 27 Codage des zones de mémoire dans les données Pointer :

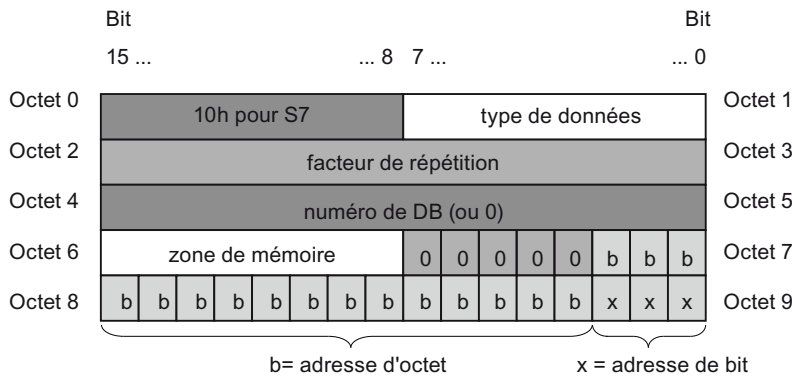
Code hexadécimal	Type de données	Description
b#16#81	I	Zone des entrées
b#16#82	Q	Zone des sorties
b#16#83	M	Zone des mémentos
b#16#84	DBX	Bloc de données
b#16#85	DIX	Bloc de données d'instance
b#16#86	L	Données locales
b#16#87	V	Données locales précédentes

### 4.4.9.2 Type de données Any

Le type de données ANY désigne le début d'une zone de données et précise sa longueur. Un pointeur ANY occupe 10 octets en mémoire et peut inclure les informations suivantes :

- Type de données : type de données des éléments de données
- Facteur de répétition : nombre d'éléments de données
- Numéro de DB : bloc de données dans lequel les éléments de données sont sauvegardés
- Zone de sauvegarde : zone de mémoire de la CPU dans laquelle les éléments de données sont sauvegardés
- Adresse de début : adresse de début des données, de type "octet.bit"

La figure suivante montre la structure du pointeur ANY :



Un pointeur ne peut pas détecter les structures ANY. Seule l'affectation à des variables locales est possible.

Tableau 4- 28 Format et exemples de pointeur ANY :

Format	Exemple de saisie	Description
P#Bloc_données.Zone_mémoire Adresse_données Type Nombre	P#DB 11.DBX 20.0 INT 10	10 mots dans le DB 11 global à partir de DBB 20.0
P#Zone_mémoire Adresse_données Type Nombre	P#M 20.0 BYTE 10	10 octets à partir de MB 20.0
	P#I 1.0 BOOL 1	Entrée I1.0

Tableau 4- 29 Codage des types de données dans un pointeur ANY

Code hexadécimal	Type de données	Description
b#16#00	Null	Pointeur nul
b#16#01	Bool	Bit
b#16#02	Byte	Octet, 8 bits
b#16#03	Char	Caractère de 8 bits
b#16#04	Word	Mot de 16 bits

Code hexadécimal	Type de données	Description
b#16#05	Int	Entier de 16 bits
b#16#37	SInt	Entier de 8 bits
b#16#35	UInt	Entier de 16 bits non signé
b#16#34	USInt	Entier de 8 bits non signé
b#16#06	DWord	Double mot de 32 bits
b#16#07	DInt	Entier double de 32 bits
b#16#36	UDInt	Entier double de 32 bits non signé
b#16#08	Real	Nombre à virgule flottante de 32 bits
b#16#0B	Time	Durée
b#16#13	String	Chaîne de caractères

Tableau 4- 30 Codage des zones de mémoire dans un pointeur ANY :

Code hexadécimal	Zone de mémoire	Description
b#16#81	I	Zone des entrées
b#16#82	Q	Zone des sorties
b#16#83	M	Zone des mémentos
b#16#84	DBX	Bloc de données
b#16#85	DIX	Bloc de données d'instance
b#16#86	L	Données locales
b#16#87	V	Données locales précédentes

#### 4.4.9.3 Type de données Variant

Le type de données Variant peut désigner des variables de différents types de données ou des paramètres. Le pointeur Variant peut désigner des structures et des éléments individuels de structures. Le pointeur Variant n'occupe aucun espace en mémoire.

Tableau 4- 31 Propriétés du pointeur Variant

Longueur (octets)	Représentation	Format	Exemple de saisie
0	Symbolique	Opérande	MaVariable
		Nom_DB.Nom_struct.Nom_élément	MonDB.Struct1.pression1
	Absolue	Opérande	%MW10
		Numéro_DB.Opérande Type Longueur	P#DB10.DBX10.0 INT 12

### 4.4.10 Accès à une "tranche" d'un type de données de variable

Vous pouvez accéder aux variables API et aux variables de blocs de données au niveau bit, octet ou mot selon leur taille. Voici la syntaxe pour accéder à une telle tranche de données :

- "<nom variable API>".xn (accès bit)
- "<nom variable API>".bn (accès octet)
- "<nom variable API>".wn (accès mot)
- "<nom bloc de données>".<nom variable>.xn (accès bit)
- "<nom bloc de données>".<nom variable>.bn (accès octet)
- "<nom bloc de données>".<nom variable>.wn (accès mot)

Il est possible d'accéder à une variable double mot par les bits 0 à 31, les octets 0 à 3 ou les mots 0 et 1. Il est possible d'accéder à une variable mot par les bits 0 à 15, les octets 0 à 2 ou le mot 0. Il est possible d'accéder à une variable octet par les bits 0 à 8 ou l'octet 0. Les tranches bit, octet et mot peuvent être utilisées partout où on attend des bits, octets ou mots comme opérandes.

																BYTE															
																WORD															
DWORD																															
x31	x30	x29	x28	x27	x26	x25	x24	x23	x22	x21	x20	x19	x18	x17	x16	x15	x14	x13	x12	x11	x10	x9	x8	x7	x6	x5	x4	x3	x2	x1	x0
b3								b2								b1								b0							
w1																w0															

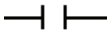

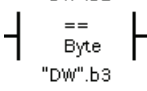
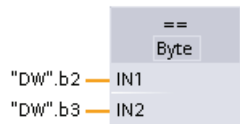
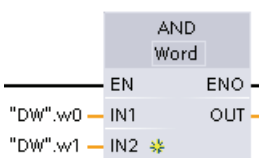
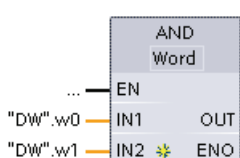
#### Remarque

Les types de données auxquels il est possible d'accéder par tranche sont les types Byte, Char, Conn\_Any, Date, DInt, DWord, Event\_Any, Event\_Att, Hw\_Any, Hw\_Device, HW\_Interface, Hw\_Io, Hw\_Pwm, Hw\_SubModule, Int, OB\_Any, OB\_Att, OB\_Cyclic, OB\_Delay, OB\_WHINT, OB\_PCYCLE, OB\_STARTUP, OB\_TIMEERROR, OB\_Tod, Port, Rtm, SInt, Time, Time\_Of\_Day, UDIInt, UInt, USInt et Word. Il est possible d'accéder aux variables API de type Real par tranche, mais pas aux variables de blocs de données de type Real.



## Exemples

Dans la table de variables API, "DW" est une variable déclarée de type DWORD. Les exemples suivants montrent l'accès à cette variable par tranche de bit, octet et mot :

	CONT	LOG	SCL
Accès bit	"DW".x11 		<pre>IF "DW".x11 THEN ... END_IF;</pre>
Accès octet	"DW".b2 == Byte "DW".b3 		<pre>IF "DW".b2 = "DW".b3 THEN ... END_IF;</pre>
Accès mot			<pre>out := "DW".w0 AND "DW".w1;</pre>

## Voir aussi

SCL (Page 166)

## 4.4.11 Accès à une variable par un type de données ajouté AT

L'ajout d'un type de données à une variable via AT vous permet d'accéder à une variable déjà déclarée d'un bloc à accès standard avec une déclaration de type de données différent ajoutée. Vous pouvez, par exemple, accéder aux bits individuels d'une variable de type Byte, Word ou DWord au moyen d'un tableau de booléens (Array of Bool).

## Déclaration

Pour ajouter un type de données à un paramètre, déclarez un paramètre supplémentaire directement après le paramètre concerné et sélectionnez le type de données "AT". L'éditeur crée l'ajout et vous pouvez alors choisir le type de données, la structure ou le tableau que vous voulez utiliser comme type de données ajouté.

4.4 Types de données

Exemple

Cet exemple montre les paramètres d'entrée d'un FB à accès standard. On ajoute comme type de données à la variable B1 de type octet un tableau de booléens :

B1	Byte
AT	AT "B1" Array [0..7] of Bool
AT[0]	Bool
AT[1]	Bool
AT[2]	Bool
AT[3]	Bool
AT[4]	Bool
AT[5]	Bool
AT[6]	Bool
AT[7]	Bool

Tableau 4- 32 Ajout d'un tableau de booléens comme type de données à un octet

7	6	5	4	3	2	1	0
AT[0]	AT[1]	AT[2]	AT[3]	AT[4]	AT[5]	AT[6]	AT[7]

Dans cet autre exemple, une variable de type DWord reçoit une structure comme nouveau type de données :

DW1	DWord
DW1_Struct	AT "DW1" Struct
S1	Word
S2	Byte
S3	Byte

Les types de données ajoutés peuvent être adressés directement dans la logique du programme :

CONT	LOG	SCL
		<pre>IF #AT[1] THEN ... END_IF;</pre>
		<pre>IF (#DW1_Struct.S1 = W#16#000C) THEN ... END_IF;</pre>
		<pre>out1 := #DW1_Struct.S2;</pre>

## Règles

- L'ajout d'un type de données à des variables n'est possible que dans les FB et FC à accès standard.
- Vous pouvez ajouter un type de données à des paramètres pour tous les types de blocs et toutes les sections de déclaration.
- Un paramètre auquel on a ajouté un type de données peut être utilisé comme n'importe quel autre paramètre du bloc.
- Vous ne pouvez pas ajouter de type de données aux paramètres de type VARIANT.
- La taille du paramètre d'ajout doit être inférieure ou égale à la taille du paramètre auquel on ajoute le type de données.
- La variable d'ajout de type de données doit être déclarée immédiatement après la variable à laquelle on ajoute le type de données et doit être identifiée au moyen du mot-clé "AT".

## Voir aussi

SCL (Page 166)

## 4.5 Utilisation d'une carte mémoire

### IMPORTANT

La CPU accepte uniquement la carte mémoire SIMATIC (Page 878) préformatée.

Avant de copier un programme sur la carte mémoire formatée, effacez de la carte mémoire tout programme précédemment sauvegardé.

Utilisez la carte mémoire soit comme carte transfert, soit comme carte programme. Tout programme que vous copiez dans la carte mémoire contient tous les blocs de code et de données, les objets technologiques éventuels et la configuration des appareils. Un programme copié **ne contient pas** les valeurs forcées de manière permanente.

- Utilisez une carte transfert (Page 118) pour copier un programme dans la mémoire de chargement interne de la CPU sans recourir à STEP 7. Une fois la carte transfert insérée, la CPU efface d'abord le programme utilisateur et les valeurs forcées de la mémoire de chargement interne, puis copie le programme de la carte transfert dans la mémoire de chargement interne. Vous devez retirer la carte transfert une fois le transfert achevé.

Vous pouvez utiliser une carte transfert vide pour accéder à une CPU protégée par mot de passe en cas d'oubli du mot de passe (Page 126). Insérer une carte transfert vide efface le programme protégé par mot de passe dans la mémoire de chargement interne de la CPU. Vous pouvez alors charger un nouveau programme dans la CPU.

4.5 Utilisation d'une carte mémoire

- Utilisez une carte programme (Page 120) comme mémoire de chargement externe pour la CPU. Insérer une carte programme dans la CPU efface toute la mémoire de chargement interne de la CPU (le programme utilisateur et les valeurs forcées en permanence). La CPU exécute alors le programme en mémoire de chargement externe (la carte programme). Effectuer un chargement dans une CPU équipée d'une carte programme actualise uniquement la mémoire de chargement externe (la carte programme).

Comme la mémoire de chargement interne de la CPU a été effacée lorsque vous y avez inséré la carte programme, cette dernière **doit** rester dans la CPU. Si vous retirez la carte programme, la CPU passe à l'état ARRET. La DEL de défaut clignote pour signaler que la carte programme a été enlevée.

Le programme copié sur une carte mémoire comprend les blocs de code, les blocs de données, les objets technologiques et la configuration des appareils. La carte mémoire **ne contient** aucune valeur forcée de manière permanente. Les valeurs forcées de manière permanente ne font pas partie du programme mais sont stockées dans la mémoire de chargement, qu'il s'agisse de la mémoire de chargement interne de la CPU ou de la mémoire de chargement externe (une carte programme). Si une carte programme est insérée dans la CPU, STEP 7 applique les valeurs forcées uniquement à la mémoire de chargement externe sur la carte programme.

Vous utiliserez également une carte mémoire pour télécharger les mises à jour du firmware (Page 123).

### 4.5.1 Insertion d'une carte mémoire dans la CPU

<b>PRUDENCE</b>
Des décharges électrostatiques peuvent endommager la carte mémoire ou son emplacement sur la CPU.
Pour éviter tout risque lorsque vous manipulez la carte mémoire, soyez en contact avec un tapis conducteur mis à la terre ou portez un bracelet spécial avec chaînette. Conservez la carte mémoire dans une boîte conductrice.



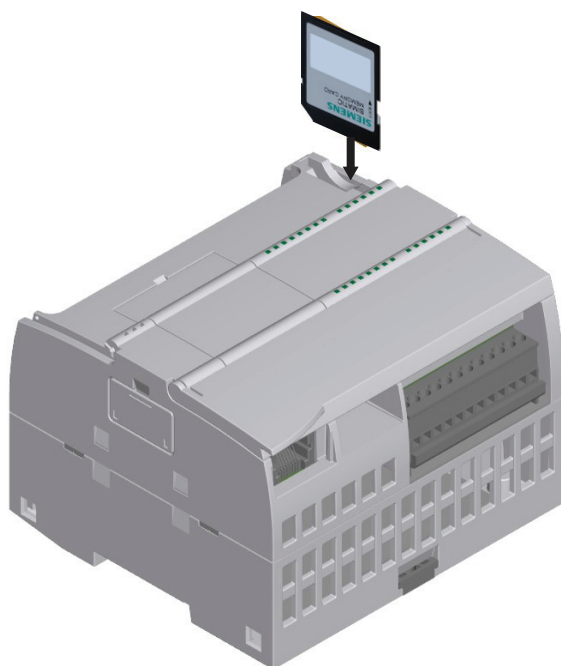
Vérifiez que la carte mémoire n'est pas protégée en écriture. Faites glisser le commutateur de protection pour l'éloigner de la position "verrou".

### PRUDENCE

Si vous insérez une carte mémoire (qu'elle soit configurée comme carte programme ou comme carte transfert) dans une CPU à l'état MARCHE, la CPU passe immédiatement à l'état ARRET, ce qui peut provoquer des dommages aux équipements ou au processus piloté. Avant d'insérer ou de retirer une carte mémoire, assurez-vous toujours que la CPU ne pilote pas activement une machine ou un processus. Installez toujours un circuit d'arrêt d'urgence pour votre application ou votre processus.

### Remarque

Si vous insérez une carte mémoire alors que la CPU est à l'état ARRET, la mémoire tampon de diagnostic affiche un message indiquant que l'évaluation de la carte mémoire a commencé. La CPU évaluera la carte mémoire la prochaine fois que vous ferez passer la CPU à l'état MARCHE, que vous effectuerez un effacement général de la CPU via MRES ou que vous mettrez la CPU hors tension puis sous tension.

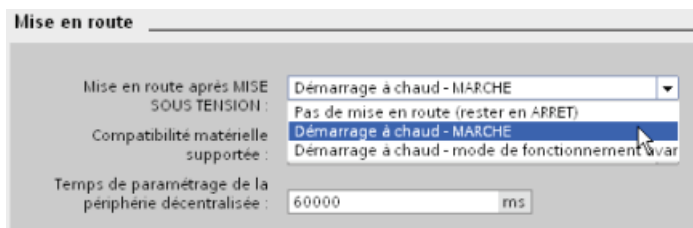


Pour insérer une carte mémoire, ouvrez le volet supérieur de la CPU et insérez la carte mémoire dans la fente. Un connecteur de type "push-push" facilite l'insertion et l'extraction de la carte.

La carte mémoire est dotée d'un détrompeur pour garantir une installation correcte.

### 4.5.2 Configuration des paramètres de mise en route de la CPU avant copie du projet dans la carte mémoire

Lorsque vous copiez un programme dans une carte transfert ou une carte programme, le programme comprend les paramètres de mise en route pour la CPU. Assurez-vous toujours, avant de copier le programme dans la carte mémoire, que vous avez configuré l'état de fonctionnement de la CPU au retour de la tension. Choisissez si la CPU doit démarrer à l'état ARRÊT, à l'état MARCHÉ ou à l'état qu'elle avait avant la mise hors puis sous tension.



### 4.5.3 Utilisation de la carte mémoire en tant que carte "transfert"

#### PRUDENCE

Des décharges électrostatiques peuvent endommager la carte mémoire ou son emplacement sur la CPU.

Pour éviter tout risque lorsque vous manipulez la carte mémoire, soyez en contact avec un tapis conducteur mis à la terre ou portez un bracelet spécial avec chaînette. Conservez la carte mémoire dans une boîte conductrice.

#### Création d'une carte transfert

Pensez toujours à configurer les paramètres de mise en route de la CPU (Page 118) avant de copier un programme dans la carte transfert. Procédez comme suit pour créer une carte transfert :

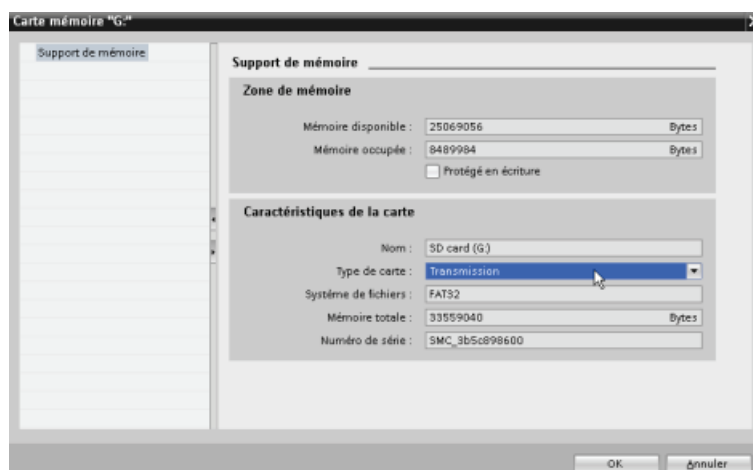
1. Insérez une carte mémoire SIMATIC vierge dans un lecteur/graveur de carte SD relié à votre ordinateur.

Si vous réutilisez une carte mémoire SIMATIC contenant un programme utilisateur ou une mise à jour du firmware, vous **devez** effacer les fichiers programme avant de réutiliser la carte. Utilisez l'Explorateur Windows pour afficher le contenu de la carte mémoire et effacez le fichier "S7\_JOB.S7S" ainsi que tout dossier "Data Logs" existant et tout dossier de répertoire (tel que "SIMATIC.S7S" ou "FWUPDATE.S7S").

2. Dans l'arborescence du projet (vue du projet), affichez le détail du dossier "SIMATIC Card Reader" et sélectionnez votre lecteur de carte.
3. Ouvrez la boîte de dialogue "Carte mémoire" en cliquant avec le bouton droit de la souris sur la lettre de lecteur correspondant à la carte mémoire dans le lecteur de carte et en sélectionnant "Propriétés" dans le menu contextuel.


- Sélectionnez "Transfert" dans le menu déroulant "Type de carte" de la boîte de dialogue "Carte mémoire".

STEP 7 crée alors la carte transfert vide. Si vous créez une carte transfert vide, par exemple pour restaurer une CPU dont vous avez oublié le mot de passe (Page 126), retirez la carte transfert du lecteur de carte.



- Ajoutez le programme en sélectionnant la CPU (par exemple, API\_1 [CPU 1214 DC/DC/DC]) dans l'arborescence du projet et en la faisant glisser dans la carte mémoire. Une autre méthode consiste à copier-coller la CPU dans la carte mémoire. Copier la CPU dans la carte mémoire ouvre la boîte de dialogue "Aperçu du chargement".
- Dans la boîte de dialogue "Aperçu du chargement", cliquez sur le bouton "Charger" pour copier la CPU dans la carte mémoire.
- Cliquez sur le bouton "Terminer" lorsqu'un message vous signale que la CPU (programme) a été chargée sans erreur.

## Utilisation d'une carte transfert

 <b>ATTENTION</b>
<p><b>Vérifiez qu'aucun processus n'est actif dans la CPU avant d'insérer la carte mémoire.</b></p> <p>L'insertion d'une carte mémoire fait passer la CPU à l'état ARRET, ce qui peut affecter le fonctionnement d'une machine ou d'un processus en ligne avec comme conséquence éventuelle la mort, des blessures graves et des dommages matériels importants.</p> <p>Avant d'insérer une carte mémoire, assurez-vous que la CPU est hors ligne et dans un état sûr.</p>

Procédez comme suit pour transférer le programme dans une CPU :

- Insérez la carte transfert dans la CPU (Page 116). Si la CPU est à l'état MARCHE, elle passe à l'état ARRET. La DEL de maintenance (MAINT) clignote pour signaler que la carte mémoire doit être évaluée.
- Mettez la CPU hors tension puis sous tension pour évaluer la carte mémoire. Les autres méthodes pour redémarrer la CPU consistent à exécuter une transition ARRET-MARCHE ou un effacement général (MRES) à partir de STEP 7.

4.5 Utilisation d'une carte mémoire

- Après redémarrage et évaluation de la carte mémoire, la CPU copie le programme dans la mémoire de chargement interne de la CPU.

La DEL RUN/STOP clignote alternativement en vert et en jaune pour indiquer que le programme est en cours de copie. Lorsque la DEL RUN/STOP passe au feu fixe (jaune) et que la DEL MAINT clignote, l'opération de copie est terminée. Vous pouvez alors retirer la carte mémoire.

- Redémarrez la CPU (en la remettant sous tension ou par l'une des méthodes alternatives de redémarrage) pour évaluer le nouveau programme transféré en mémoire de chargement interne.

La CPU passe alors au mode (MARCHE ou ARRET) que vous avez configuré pour la mise en route dans le projet.

---

**Remarque**

Vous devez retirer la carte transfert avant de faire passer la CPU à l'état MARCHE.

---

#### 4.5.4 Utilisation de la carte mémoire en tant que carte "programme"

<b>PRUDENCE</b>
Des décharges électrostatiques peuvent endommager la carte mémoire ou son emplacement sur la CPU.
Pour éviter tout risque lorsque vous manipulez la carte mémoire, soyez en contact avec un tapis conducteur mis à la terre ou portez un bracelet spécial avec chaînette. Conservez la carte mémoire dans une boîte conductrice.



Vérifiez que la carte mémoire n'est pas protégée en écriture. Faites glisser le commutateur de protection pour l'éloigner de la position "verrou".

Avant de copier des éléments de programme sur la carte programme, effacez de la carte mémoire tout programme précédemment sauvegardé.



## Création d'une carte programme

Lorsque vous l'utilisez comme carte programme, la carte mémoire constitue la mémoire de chargement externe de la CPU. Si vous retirez la carte programme, la mémoire de chargement interne de la CPU est vide.

### Remarque

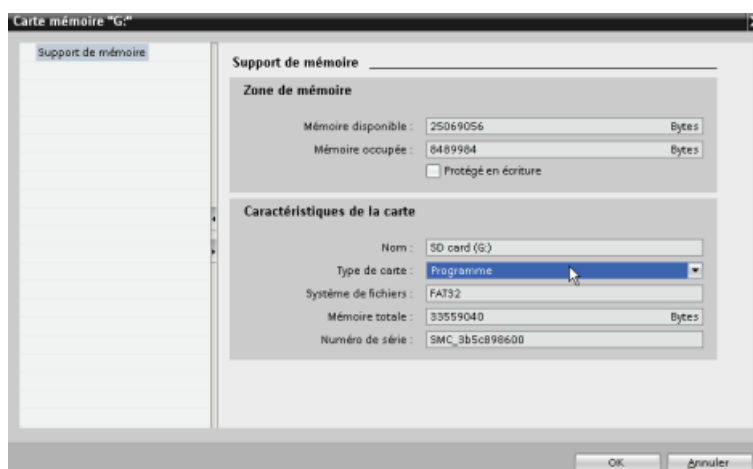
Si vous insérez une carte mémoire vierge dans la CPU et effectuez une évaluation de carte mémoire, soit en mettant la CPU hors tension puis sous tension, en effectuant un passage de l'état ARRET à l'état MARCHE ou en exécutant un effacement général (MRES), le programme et les valeurs de forçage dans la mémoire de chargement interne de la CPU sont copiés dans la carte mémoire. La carte mémoire est maintenant une carte programme. Une fois la copie achevée, le programme dans la mémoire de chargement interne de la CPU est effacé. La CPU passe alors au mode de mise en route configuré (MARCHE ou ARRET).

Pensez toujours à configurer les paramètres de mise en route de la CPU (Page 118) avant de copier un projet dans la carte programme. Procédez comme suit pour créer une carte programme :

1. Insérez une carte mémoire SIMATIC vierge dans un lecteur/graveur de carte SD relié à votre ordinateur.

Si vous réutilisez une carte mémoire SIMATIC contenant un programme utilisateur ou une mise à jour du firmware, vous **devez** effacer les fichiers programme avant de réutiliser la carte. Utilisez l'Explorateur Windows pour afficher le contenu de la carte mémoire et effacez le fichier "S7\_JOB.S7S" ainsi que tout dossier "Data Logs" existant et tout dossier de répertoire (tel que "SIMATIC.S7S" ou "FWUPDATE.S7S").

2. Dans l'arborescence du projet (vue du projet), affichez le détail du dossier "SIMATIC Card Reader" et sélectionnez votre lecteur de carte.
3. Ouvrez la boîte de dialogue "Carte mémoire" en cliquant avec le bouton droit de la souris sur la lettre de lecteur correspondant à la carte mémoire dans le lecteur de carte et en sélectionnant "Propriétés" dans le menu contextuel.
4. Sélectionnez "Programme" dans le menu déroulant de la boîte de dialogue "Carte mémoire".



4.5 Utilisation d'une carte mémoire

5. Ajoutez le programme en sélectionnant la CPU (par exemple, API\_1 [CPU 1214 DC/DC/DC]) dans l'arborescence du projet et en la faisant glisser dans la carte mémoire. Une autre méthode consiste à copier-coller la CPU dans la carte mémoire. Copier la CPU dans la carte mémoire ouvre la boîte de dialogue "Aperçu du chargement".
6. Dans la boîte de dialogue "Aperçu du chargement", cliquez sur le bouton "Charger" pour copier la CPU dans la carte mémoire.
7. Cliquez sur le bouton "Terminer" lorsqu'un message vous signale que la CPU (programme) a été chargée sans erreur.

Utilisation d'une carte programme comme mémoire de chargement pour votre CPU

 **ATTENTION**

**Vérifiez qu'aucun processus n'est actif dans la CPU avant d'insérer la carte mémoire.**

L'insertion d'une carte mémoire fait passer la CPU à l'état ARRET, ce qui peut affecter le fonctionnement d'une machine ou d'un processus en ligne avec comme conséquence éventuelle la mort, des blessures graves et des dommages matériels importants.

Avant d'insérer une carte mémoire, assurez-vous que la CPU est hors ligne et dans un état sûr.

Procédez comme suit pour utiliser une carte programme avec votre CPU :

1. Insérez la carte programme dans la CPU. Si la CPU est à l'état MARCHE, elle passe à l'état ARRET. La DEL de maintenance (MAINT) clignote pour signaler que la carte mémoire doit être évaluée.
2. Mettez la CPU hors tension puis sous tension pour évaluer la carte mémoire. Les autres méthodes pour redémarrer la CPU consistent à exécuter une transition ARRET-MARCHE ou un effacement général (MRES) à partir de STEP 7.
3. Après que la CPU a redémarré et a évalué la carte programme, elle efface sa mémoire de chargement interne.

La CPU passe alors au mode (MARCHE ou ARRET) que vous avez configuré pour la mise en route de la CPU.

La carte programme doit rester dans la CPU. Retirer la carte programme laisse la CPU sans aucun programme dans la mémoire de chargement interne.

 **ATTENTION**

Si vous retirez la carte programme, la CPU perd sa mémoire de chargement externe et génère une erreur. La CPU passe à l'état ARRET et la DEL de défaut se met à clignoter.

Les appareils de commande peuvent présenter des défaillances dans des situations non sûres et provoquer un fonctionnement inattendu des appareils pilotés pouvant entraîner la mort, des blessures graves et des dommages matériels importants.

## 4.5.5 Mise à jour du firmware

### PRUDENCE

Des décharges électrostatiques peuvent endommager la carte mémoire ou son emplacement sur la CPU.

Pour éviter tout risque lorsque vous manipulez la carte mémoire, soyez en contact avec un tapis conducteur mis à la terre ou portez un bracelet spécial avec chaînette. Conservez la carte mémoire dans une boîte conductrice.

Vous utilisez une carte mémoire pour télécharger les mises à jour du firmware depuis le Customer Support (<http://www.siemens.com/automation/>). Naviguez dans ce site Web jusqu'à **Automation Technology > Systèmes d'automatisation > Systèmes d'automatisation industrie SIMATIC > Automates programmables > Modular Controller SIMATIC S7 > SIMATIC S7-1200**. De là, continuez à naviguer jusqu'au type spécifique de module que vous devez mettre à jour. Sous "Support", cliquez sur le lien "Téléchargements logiciels" pour poursuivre.

Vous pouvez aussi accéder directement à la page Web des téléchargements S7-1200 (<http://support.automation.siemens.com/WW/view/fr/34612486/133100>).

### Remarque

Il n'est pas possible d'actualiser une CPU S7-1200 de version V2.2 ou antérieure à S7-1200 V3.0 par mise à jour du firmware.

### PRUDENCE

**N'utilisez pas l'utilitaire de formatage de Windows ou tout autre utilitaire de formatage pour reformater la carte mémoire.**

Si une carte mémoire Siemens est reformatée avec l'utilitaire de formatage de Microsoft Windows, la carte mémoire ne peut plus être utilisée par une CPU S7-1200.

Pour charger la mise à jour du firmware sur votre carte mémoire, exécutez les étapes suivantes :

1. Insérez une carte mémoire vierge SIMATIC MC 24 MB dans le lecteur/graveur de carte SD relié à votre ordinateur.

Si vous réutilisez une carte mémoire SIMATIC contenant un programme utilisateur ou une autre mise à jour du firmware, vous devez effacer un certain nombre de fichiers sur la carte mémoire.

<b>PRUDENCE</b>
-----------------

<b>N'effacez pas les fichiers cachés "__LOG__" et "crdinfo.bin" sur la carte mémoire.</b>
-------------------------------------------------------------------------------------------

Les fichiers "__LOG__" et "crdinfo.bin" sont requis pour la carte mémoire. Si vous supprimez ces fichiers, vous ne pourrez plus utiliser la carte mémoire avec la CPU.
------------------------------------------------------------------------------------------------------------------------------------------------------------------------

Pour réutiliser une carte mémoire, vous **devez** effacer le fichier "S7\_JOB.S7S" et tout dossier "Archives de variables" existant ou tout autre dossier (tel que "SIMATIC.S7S" ou "FWUPDATE.S7S") avant de télécharger la mise à jour du firmware. N'effacez **pas** toutefois les fichiers "\_\_LOG\_\_" et "crdinfo.bin". (Ces fichiers sont en général cachés et sont nécessaires.) Utilisez Windows Explorer pour afficher le contenu de la carte mémoire et effacer le fichier ou les dossiers.

2. Sélectionnez le fichier autoextractible (.exe) de la mise à jour du firmware correspondant à votre module et chargez-le sur votre ordinateur. Double-cliquez sur le fichier de mise à jour, sélectionnez le chemin cible qui sera pris comme répertoire racine de la carte mémoire SIMATIC et lancez l'extraction. Une fois l'extraction terminée, le répertoire racine de la carte mémoire contiendra un répertoire "FWUPDATE.S7S" et le fichier "S7\_JOB.S7S".

Procédez comme suit pour installer la mise à jour du firmware :

**! ATTENTION**

**Vérifiez qu'un processus ne soit pas en cours sur la CPU avant d'installer la mise à jour du firmware.**

L'installation de la mise à jour du firmware fait passer la CPU à l'arrêt, ce qui peut affecter l'exécution d'un process ou d'une machine en ligne, dont la mise en route inattendue peut entraîner la mort, des blessures graves et des dommages matériels importants.

Avant d'insérer la carte mémoire, assurez-vous que la CPU se trouve hors ligne et dans un état sûr.

1. Insérez la carte mémoire dans la CPU. Si la CPU est à l'état MARCHE, elle passe à l'état ARRET. La DEL de maintenance clignote pour signaler que la carte mémoire doit être évaluée.
2. Mettez la CPU hors tension puis sous tension avant de lancer la mise à jour du firmware. Les autres méthodes pour redémarrer la CPU consistent à exécuter une transition ARRET-MARCHE ou un effacement général (MRES) à partir de STEP 7.

**IMPORTANT**

Pour achever la mise à niveau du firmware du module, assurez-vous que l'alimentation 24 VCC externe est maintenue sur le module.

Après le redémarrage de la CPU, la mise à jour du firmware démarre. La DEL RUN/STOP clignote alternativement en vert et en jaune pour indiquer que la mise à jour est copiée. Lorsque la DEL RUN/STOP passe au feu fixe (jaune) et que la DEL MAINT clignote, l'opération de copie est terminée. Vous devez alors retirer la carte mémoire.

3. Après avoir retiré la carte mémoire, redémarrez la CPU (soit en la remettant sous tension, soit par les méthodes alternatives de redémarrage) pour charger le nouveau firmware.

Le programme utilisateur et la configuration matérielle ne sont pas affectés par la mise à jour du firmware. Après sa mise sous tension, la CPU passe à l'état de mise en route configuré. (si le mode de mise en route paramétré pour votre CPU était "Démarrage à chaud dans le mode avant la mise hors tension", la CPU sera sur ARRET, car le dernier état de fonctionnement de la CPU était ARRET.)

**Remarque****Mise à jour de multiples modules connectés à la CPU**

Si votre configuration matérielle contient de multiples modules qui correspondent à un seul fichier de mise à jour du firmware sur la carte mémoire, la CPU applique les mises à jour à tous les modules appropriés (CM, SM, SB) dans l'ordre de configuration c'est-à-dire dans l'ordre croissant de la position des modules dans la Configuration de l'Appareil dans STEP 7.

Si vous avez téléchargé plusieurs mises à jour du firmware sur la carte mémoire pour plusieurs modules, la CPU applique les mises à jour dans l'ordre dans lequel vous les avez téléchargées sur la carte mémoire.

## 4.6 Récupération en cas d'oubli du mot de passe

Si vous avez oublié le mot de passe d'une CPU protégée par mot de passe, utilisez une carte transfert vide pour effacer le programme protégé par mot de passe. La carte transfert vide efface la mémoire de chargement interne de la CPU. Vous pouvez alors charger un nouveau programme utilisateur de STEP 7 dans la CPU.

Pour plus d'informations sur la création et l'utilisation d'une carte transfert vide, reportez-vous au paragraphe sur les cartes transfert (Page 118).

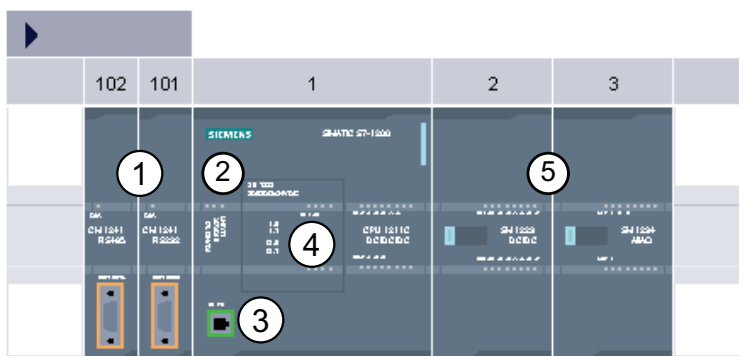
 **ATTENTION**

Si vous insérez une carte transfert dans une CPU à l'état MARCHE, la CPU passe à l'état ARRET. Les appareils de commande peuvent présenter des défaillances dans des situations non sûres et provoquer un fonctionnement inattendu des appareils pilotés pouvant entraîner la mort, des blessures graves et des dommages matériels importants.

Vous devez retirer la carte transfert avant de faire passer la CPU à l'état MARCHE.

## Configuration des appareils

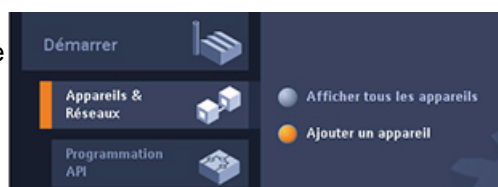
Vous créez la configuration des appareils pour votre automate en ajoutant une CPU et des modules supplémentaires à votre projet.



- ① Module de communication (CM) et processeur de communication (CP) : jusqu'à 3, insérés aux emplacements 101, 102 et 103
- ② CPU : emplacement 1
- ③ port Ethernet de la CPU
- ④ Signal Board (SB), Communication Board (CB) ou Battery Board (BB) : au plus 1, inséré dans la CPU
- ⑤ Module d'entrées-sorties (SM) TOR ou analogiques : jusqu'à 8, insérés aux emplacements 2 à 9  
(les CPU 1214C et CPU 1215C en autorisent 8, la CPU 1212C en autorise 2, la CPU 1211C n'en autorise aucun)

Pour créer la configuration des appareils, ajoutez un appareil à votre projet.

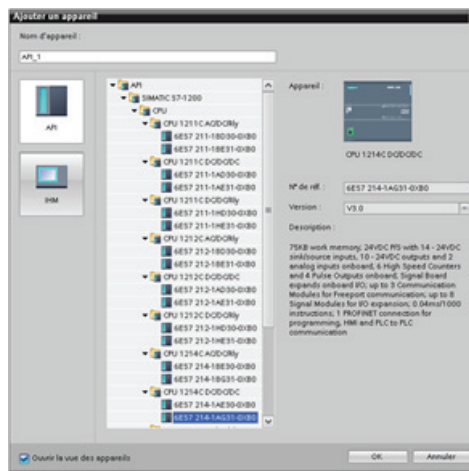
- Dans la vue du portail, sélectionnez "Appareils & réseaux" et cliquez sur "Ajouter un appareil".
- Dans la vue du projet, double-cliquez sur "Ajouter un appareil" sous le nom de projet.



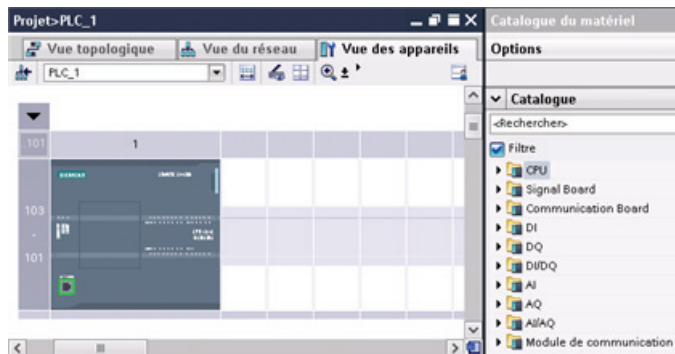
## 5.1 Insertion d'une CPU

Vous créez votre configuration d'appareil en insérant une CPU dans votre projet. Veillez à bien insérer le modèle et la version de firmware corrects à partir de la liste. Sélectionner la CPU dans la boîte de dialogue "Ajouter un appareil" crée le châssis et la CPU.

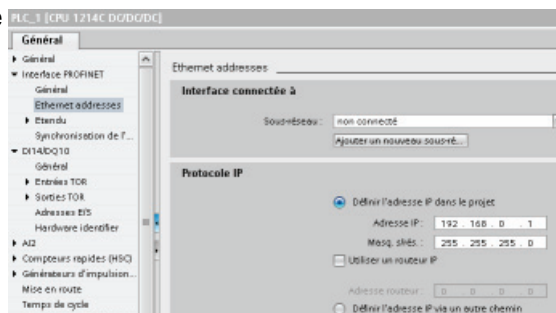
Boîte de dialogue "Ajouter un appareil"



Vue des appareils de la configuration matérielle



Sélectionner la CPU dans la vue des appareils provoque l'affichage des propriétés de la CPU dans la fenêtre d'inspection.

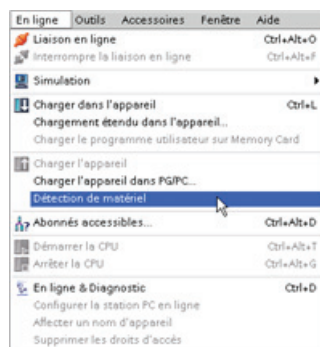




### Remarque

La CPU ne dispose pas d'une adresse IP préconfigurée. Vous devez affecter manuellement une adresse IP à la CPU pendant la configuration des appareils. Si votre CPU est connectée à un routeur sur le réseau, vous entrez aussi l'adresse IP du routeur.

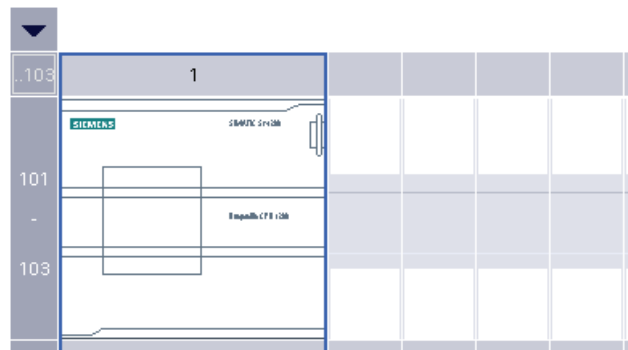
## 5.2 Détection de la configuration d'une CPU non spécifiée



Lorsque vous êtes connecté à une CPU, vous pouvez charger la configuration de cette CPU, y compris tous les modules, dans votre projet. Il vous suffit de créer un nouveau projet et de sélectionner la "CPU non spécifiée" au lieu d'une CPU spécifique (vous pouvez également ne pas passer par la configuration matérielle en sélectionnant "Créer un programme API" dans "Mise en route". STEP 7 crée ensuite automatiquement une CPU non spécifiée.)

Dans l'éditeur de programmes, vous sélectionnez la commande "Détection du matériel" du menu "En ligne".

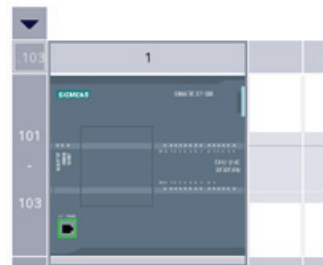
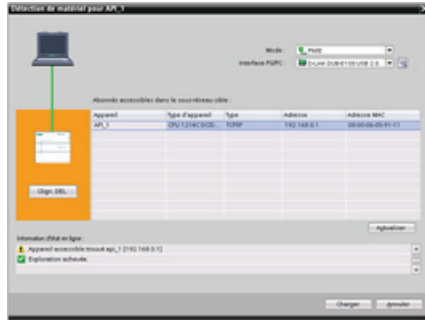
Dans l'éditeur de configuration des appareils, vous sélectionnez l'option de détection de configuration de l'appareil connecté.



L'appareil n'est pas spécifié.

- Veuillez utiliser [Catalogue du matériel](#) pour spécifier la CPU.
- ou [déterminez](#) la configuration de l'appareil connecté.

Une fois que vous avez sélectionné la CPU dans la boîte de dialogue en ligne et cliqué sur le bouton Charger, STEP 7 télécharge la configuration matérielle depuis la CPU, en incluant tous les modules (SM, SB ou CM). Vous pouvez alors configurer les paramètres de la CPU et des modules.



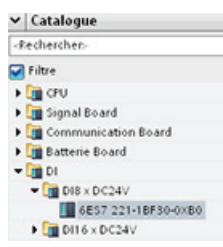

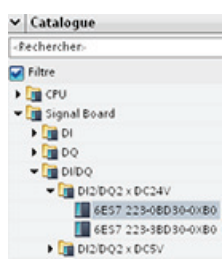



### 5.3 Ajout de modules à la configuration

Utiliser le catalogue de matériel pour ajouter des modules à la CPU :

- Le module d'entrées-sorties (SM) fournit des points d'E/S numériques ou analogiques supplémentaires. Ces modules se raccordent sur le côté droit de la CPU.
- Le Signal Board (SB) fournit seulement quelques points d'E/S supplémentaires pour la CPU. Le SB s'installe à l'avant de la CPU.
- Le Battery Board 1297 fournit une sauvegarde à long terme de l'horloge temps réel. Il s'installe à l'avant de la CPU.
- Le Communication Board (CB) fournit un port de communication supplémentaire (comme RS485). Le Communication Board s'installe à l'avant de la CPU.
- Le module de communication (CM) et le processeur de communication (CP) fournissent un port de communication supplémentaire, comme PROFIBUS ou GPRS. Ces modules se raccordent sur le côté gauche de la CPU.

Pour insérer un module dans la configuration des appareils, sélectionnez le module concerné dans le catalogue du matériel et double-cliquez ou faites glisser le module dans l'emplacement mis en évidence. Pour que les modules soient opérationnels, vous devez les ajouter à la configuration des appareils et charger dans la CPU la configuration matérielle pour les modules.

Tableau 5- 1 Ajouter un module à la configuration de l'appareil

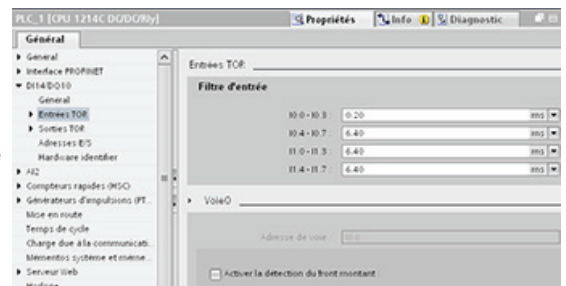
Module	Sélectionnez le module	Insérez le module	Résultat
SM			
SB, BB ou CB			
CM ou CP			

## 5.4 Configuration du fonctionnement de la CPU

Pour configurer les paramètres de fonctionnement de la CPU, sélectionnez la CPU dans la vue des appareils (CPU entourée de bleu) et servez-vous de l'onglet "Propriétés" de la fenêtre d'inspection.

Pour configurer les temps de filtre des entrées, sélectionnez "Entrées TOR". Le temps de filtre par défaut pour les entrées TOR est de 6,4 ms.

Chaque entrée présente une configuration de filtre unique qui s'applique à toutes les utilisations : entrées du processus, alarmes, capture d'impulsions et entrées HSC.



**Remarque**

Si un compteur rapide (HSC) n'est pas configuré pour utiliser une entrée, le paramétrage de filtre choisi dans cet écran s'applique. Si un compteur rapide est configuré pour utiliser une entrée, le filtre pour cette entrée est automatiquement paramétré à 800 ns et n'est pas affecté par la configuration dans cet écran.


 <b>ATTENTION</b>
<p>Si le temps de filtre pour une voie d'entrée TOR est modifié par rapport à un paramétrage précédent, une nouvelle valeur d'entrée de niveau "0" peut devoir être présente pendant une durée cumulée allant jusqu'à 20,0 ms pour que le filtre réagisse pleinement aux nouvelles entrées. Pendant ce temps, les événements d'impulsion "0" courts de moins de 20,0 ms peuvent ne pas être détectés ni comptés.</p> <p>Cette modification des temps de filtre peut provoquer un fonctionnement inattendu des machines ou du processus, pouvant entraîner la mort, des blessures graves et des dommages matériels importants.</p> <p>Pour garantir la prise en compte immédiate d'un nouveau temps de filtre, mettez la CPU hors tension puis sous tension.</p>

Tableau 5- 2 Propriétés de la CPU

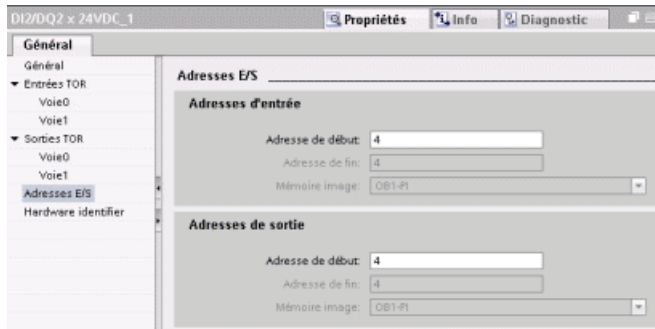
Propriété	Description
Interface PROFINET	Définit l'adresse IP pour la CPU et la synchronisation d'horloge.
DI, DO et AI	Configure le comportement des E/S TOR et analogiques locales (intégrées), par exemple les temps de filtre des entrées TOR et la réaction des sorties TOR à un arrêt de la CPU.
Compteurs rapides (Page 359) et générateurs d'impulsions (Page 332)	Valide et configure les compteurs rapides (HSC) et les générateurs d'impulsions utilisés pour les fonctions PTO (sortie de trains d'impulsions) et PWM (modulation de largeur des impulsions).  Lorsque vous configurez les sorties de la CPU ou du Signal Board en tant que générateurs d'impulsions (pour les instructions PWM ou de commande de mouvement), les adresses des sorties correspondantes (Q0.0, Q0.1, Q4.0 et Q4.1) sont supprimées de la mémoire Q et ne peuvent pas être utilisées à d'autres fins dans le programme utilisateur. Si votre programme utilisateur écrit une valeur dans une sortie utilisée comme générateur d'impulsions, la CPU n'écrit pas cette valeur dans la sortie physique.
Mise en route (Page 73)	<b>Démarrage après mise sous tension</b> : Sélectionne le comportement de la CPU après une mise sous tension, par exemple démarrage à l'état ARRET ou passage à l'état MARCHE après un démarrage à chaud.

Propriété	Description
	<p><b>Compatibilité matérielle prise en charge</b> : Configure la stratégie de remplacement pour tous les composants système (SM, SB, CM, CP et CPU) :</p> <ul style="list-style-type: none"> <li>• Autoriser un module de remplacement compatible</li> <li>• Autoriser tout module de remplacement (option par défaut)</li> </ul> <p>Chaque module contient en interne les conditions de compatibilité de remplacement requises basées sur le nombre d'E/S, la compatibilité électrique et d'autres points de comparaison correspondants. Par exemple, un SM à 16 voies serait un module de remplacement compatible pour un SM à 8 voies, mais un SM à 8 voies ne le serait pas pour un SM à 16 voies. Si vous sélectionnez "Autoriser un module de remplacement compatible", STEP 7 applique les règles de remplacement ; sinon, STEP 7 permet tous les remplacements.</p> <p><b>Temps d'affectation des paramètres pour la périphérie décentralisée</b> : Configure une durée maximale (par défaut : 60000 ms) pour que la périphérie décentralisée soit mise en ligne. (Les CM et les CP reçoivent les paramètres de communication et d'alimentation de la CPU pendant la mise en route. Ce temps d'affectation fournit du temps pour la mise en ligne des E/S connectées au CM ou au CP.)</p> <p>La CPU passe à l'état MARCHE dès que la périphérie décentralisée est en ligne, sans tenir compte du temps d'affectation. Si la périphérie décentralisée n'a pas été mise en ligne à l'expiration de ce temps, la CPU passe quand même à l'état MARCHE, mais sans la périphérie décentralisée.</p> <p><b>Remarque</b> : Si votre configuration utilise un CM 1243-5 (maître PROFIBUS), ne donnez pas une valeur inférieure à 15 secondes (15000 ms) à ce paramètre afin de garantir que le module sera mis en ligne.</p>
Cycle (Page 85)	Définit un temps de cycle maximum ou un temps de cycle minimum fixe.
Charge due à la communication	Réserve un pourcentage du temps CPU aux tâches de communication.
Mémoire système et mémoire de cadence (Page 90)	Valide un octet pour les fonctions "mémoire système" et valide un octet pour les fonctions "mémoire de cadence" (où chaque bit bascule entre 1 et 0 selon une fréquence prédéfinie).
Serveur Web (Page 533)	Valide et configure la fonction serveur Web.
Heure	Permet de sélectionner le fuseau horaire et de configurer l'heure d'été.
Protection (Page 175)	Définit la protection en lecture/écriture et le mot de passe pour accéder à la CPU.
Ressources de liaison (Page 452)	Fournit un récapitulatif des liaisons de communication disponibles pour la CPU et le nombre de liaisons qui ont été configurées.
Présentation des adresses	Fournit un récapitulatif des adresses d'E/S qui ont été configurées pour la CPU.

## 5.5 Configuration des paramètres des modules

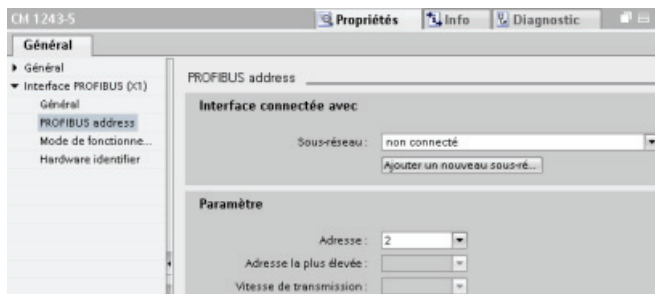
Pour configurer les paramètres de fonctionnement d'un module, sélectionnez le module dans la vue des appareils et servez-vous de l'onglet "Propriétés" de la fenêtre d'inspection.

### Configuration d'un module d'entrées-sorties (SM) ou d'un Signal Board (SB)



- E/S TOR : Il est possible de configurer des entrées pour la détection de front montant ou de front descendant (en associant chaque entrée à un événement et une alarme de processus) ainsi que pour la capture d'impulsions (afin qu'elles restent à 1 après une impulsion momentanée) jusqu'à l'actualisation suivante de la mémoire image des entrées. Les sorties peuvent être figées ou utiliser une valeur de remplacement.
- E/S analogiques : Configurez les paramètres des entrées individuelles, par exemple le type de mesure (tension ou courant), la plage et le lissage, et activez le diagnostic de débordement bas ou haut. Les sorties analogiques fournissent des paramètres relatifs au type de sortie (tension ou courant) et au diagnostic, de court-circuit (pour les sorties de tension) ou de limite supérieure/inférieure par exemple. Vous ne configurez pas les plages des entrées et sorties analogiques en unités physiques dans la boîte de dialogue Propriétés, car ce point doit être géré dans la logique de votre programme comme décrit à la rubrique "Traitement des valeurs analogiques (Page 99)".
- Adresses de diagnostic des E/S : Configure l'adresse de début pour le jeu d'entrées et de sorties du module.

### Configuration d'une interface de communication (CM, CP ou CB)




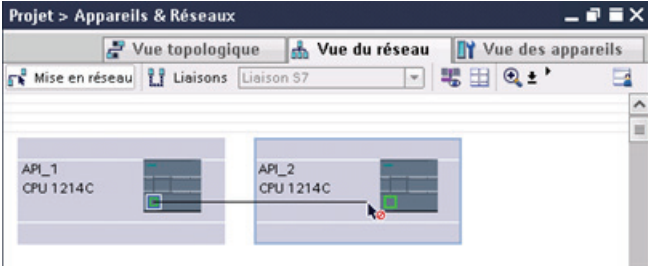
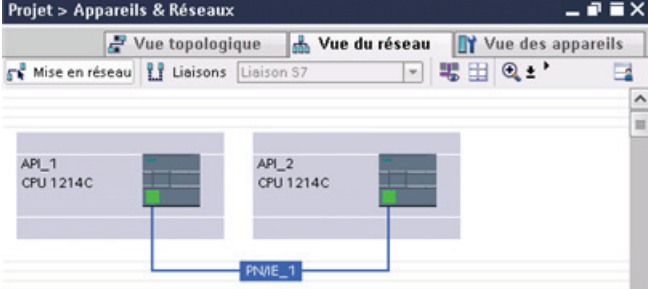
Vous configurez les paramètres pour le réseau en fonction du type d'interface de communication.

## 5.6 Configuration de la CPU pour la communication

### 5.6.1 Création d'une liaison réseau

Utilisez la "Vue du réseau" de la configuration des appareils pour créer les liaisons réseau entre les appareils dans votre projet. Une fois la liaison réseau créée, servez-vous de l'onglet "Propriétés" de la fenêtre d'inspection pour configurer les paramètres du réseau.

Tableau 5- 3 Création d'une liaison réseau

Action	Résultat
Sélectionnez la "Vue du réseau" pour afficher les appareils à connecter.	
Sélectionnez le port sur un appareil et tracez la liaison vers le port sur le deuxième appareil.	
Relâchez le bouton de la souris pour créer la liaison réseau.	

### 5.6.2 Configuration du routage local/partenaire

La fenêtre d'inspection affiche les propriétés de la liaison lorsque vous sélectionnez n'importe quelle partie de l'instruction. Indiquez les paramètres de communication dans l'onglet "Configuration" des "Propriétés" de l'instruction de communication.

Tableau 5- 4 Configuration du routage (à l'aide des propriétés de l'instruction)

TCP, ISO sur TCP et UDP :	Propriétés de liaison
<p>Pour les protocoles Ethernet TCP, ISO sur TCP et UDP, utilisez les "Propriétés" de l'instruction (TSEND_C, TRCV_C ou TCON) pour configurer les liaisons "locale/partenaire".</p> <p>La figure montre les "Propriétés" d'une liaison ISO sur TCP dans l'onglet "Configuration".</p>	

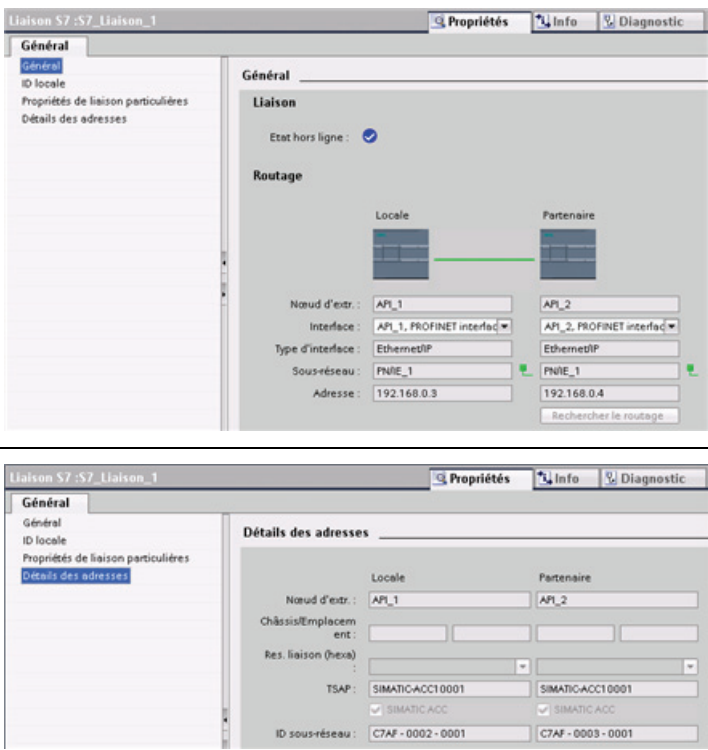
#### Remarque

Lorsque vous configurez les propriétés de liaison pour une CPU, STEP 7 vous permet de sélectionner dans la CPU partenaire un DB de liaison spécifique (s'il existe) ou de créer le DB de liaison pour la CPU partenaire. La CPU partenaire doit avoir été préalablement créée pour le projet et ne peut pas être une CPU "non spécifiée".

Vous devez toujours insérer une instruction TSEND\_C, TRCV\_C ou TCON dans le programme utilisateur de la CPU partenaire. Lorsque vous insérez l'instruction, sélectionnez le DB de liaison qui a été créé par la configuration.



Tableau 5- 5 Configuration du routage pour la communication S7 (configuration des appareils)

Communication S7 (GET et PUT)	Propriétés de liaison
<p>Pour la communication S7, utilisez l'éditeur "Appareils &amp; réseaux" du réseau pour configurer les liaisons locale/partenaire. Vous pouvez cliquer sur le bouton "Mis en valeur : Liaison" pour accéder aux "Propriétés".</p> <p>L'onglet "Général" fournit plusieurs propriétés :</p> <ul style="list-style-type: none"> <li>• "Général" (montré)</li> <li>• "ID locale"</li> <li>• "Propriétés de liaison particulières"</li> <li>• "Détails de l'adresse" (montré)</li> </ul>	

Reportez-vous à "Protocoles" (Page 458) dans le chapitre "PROFINET" ou à "Création d'une liaison S7" (Page 524) dans le chapitre "Communication S7" pour obtenir plus d'informations ainsi que la liste des instructions de communication disponibles.

Tableau 5- 6 Paramètres pour la liaison entre CPU

Paramètre	Définition	
Adresse	Adresses IP affectées	
Général	Noeud d'extrémité	Nom affecté à la CPU partenaire (réceptrice)
	Interface	Nom affecté aux interfaces
	Sous-réseau	Nom affecté aux sous-réseaux
	Type d'interface	<i>Communication S7 uniquement</i> : Type de l'interface
	Type de liaison	Type de protocole Ethernet
	ID de liaison	Numéro d'identification
	Données de liaison	Emplacement de stockage des données des CPU locale et partenaire
	Etablissement d'une liaison active	Bouton radio pour sélectionner la CPU locale ou partenaire en tant que liaison active
Détails de l'adresse	Noeud d'extrémité	<i>Communication S7 uniquement</i> : Nom affecté à la CPU partenaire (réceptrice)
	Châssis/emplacement	<i>Communication S7 uniquement</i> : Adresse du châssis et de l'emplacement
	Ressources de liaison	<i>Communication S7 uniquement</i> : Composant du TSAP utilisé lors de la configuration d'une liaison S7 avec une CPU S7-300 ou S7-400
	Port (décimal)	TCP et UDP : Port de la CPU partenaire en format décimal

5.6 Configuration de la CPU pour la communication

Paramètre		Définition
	TSAP <sup>1</sup> et ID de sous-réseau :	ISO sur TCP (RFC 1006) et communication S7 : TSAP des CPU locale et partenaire en formats ASCII et hexadécimal

<sup>1</sup> Lorsque vous configurez une liaison avec une CPU S7-1200 pour ISO sur TCP, utilisez uniquement des caractères ASCII dans l'extension TSAP pour les partenaires de communication passifs.

**TSAP (points d'accès au service transport)**

Grâce aux points d'accès au service transport (TSAP), le protocole ISO sur TCP et la communication S7 autorisent des liaisons multiples à une adresse IP unique (liaisons 64K maximum). Les TSAP identifient de manière unique ces liaisons de noeud d'extrémité à une adresse IP.

Vous définissez les points TSAP à utiliser dans la section "Détails de l'adresse" de la boîte de dialogue "Paramètres de liaison". Vous indiquez le point TSAP d'une liaison dans la CPU dans la zone "TSAP local". Vous indiquez le point TSAP affecté à la liaison dans votre CPU partenaire dans la zone "TSAP partenaire".

**Numéros de port**

Avec les protocoles TCP et UDP, la configuration des paramètres de liaison de la CPU locale (active) doit indiquer l'adresse IP éloignée et le numéro de port de la CPU partenaire (passive).

Vous définissez les ports à utiliser dans la section "Détails de l'adresse" de la boîte de dialogue "Paramètres de liaison". Vous indiquez le port d'une liaison dans la CPU dans la zone "Port local". Vous indiquez le port affecté à la liaison dans votre CPU partenaire dans la zone "Port partenaire".

**5.6.3 Paramètres pour la liaison PROFINET**

Les instructions TSEND\_C, TRCV\_C et TCON nécessitent l'indication de paramètres relatifs à la liaison pour que la liaison au partenaire soit établie. Ces paramètres sont indiqués par la structure TCON\_Param pour les protocoles TCP, ISO sur TCP et UDP. Typiquement, vous utilisez l'onglet "Configuration" des "Propriétés" de l'instruction pour fournir ces paramètres. Si l'onglet "Configuration" n'est pas accessible, vous devez fournir la structure TCON\_Param par programme.

Tableau 5- 7 Structure de la description de la liaison (TCON\_Param)

Octet	Paramètre et type de données		Description
0 ... 1	block_length	UInt	Longueur : 64 octets (fixe)
2 ... 3	id	CONN_OUC (Word)	Référence à cette liaison : Plage de valeurs : 1 (valeur par défaut) à 4095. Indiquez la valeur de ce paramètre pour l'instruction TSEND_C, TRCV_C ou TCON sous ID.

Octet	Paramètre et type de données		Description
4	connection_type	USInt	Type de liaison : <ul style="list-style-type: none"> <li>• 17: TCP (par défaut)</li> <li>• 18: ISO sur TCP</li> <li>• 19: UDP</li> </ul>
5	active_est	Bool	ID pour le type de liaison : <ul style="list-style-type: none"> <li>• TCP et ISO sur TCP : <ul style="list-style-type: none"> <li>– FAUX : liaison passive</li> <li>– VRAI : liaison active (valeur par défaut)</li> </ul> </li> <li>• UDP : FAUX</li> </ul>
6	local_device_id	USInt	ID pour l'interface PROFINET ou Industrial Ethernet locale : 1 (valeur par défaut)
7	local_tsap_id_len	USInt	Longueur du paramètre local_tsap_id utilisé, en octets ; valeurs possibles : <ul style="list-style-type: none"> <li>• TCP : 0 (actif, valeur par défaut) ou 2 (passif)</li> <li>• ISO sur TCP : 2 à 16</li> <li>• UDP : 2</li> </ul>
8	rem_subnet_id_len	USInt	Ce paramètre n'est pas utilisé.
9	rem_staddr_len	USInt	Longueur de l'adresse du noeud d'extrémité partenaire, en octets : <ul style="list-style-type: none"> <li>• 0: non précisé (paramètre rem_staddr sans objet)</li> <li>• 4 (valeur par défaut) : adresse IP valide dans le paramètre rem_staddr (uniquement pour TCP et ISO sur TCP)</li> </ul>
10	rem_tsap_id_len	USInt	Longueur du paramètre rem_tsap_id utilisé, en octets ; valeurs possibles : <ul style="list-style-type: none"> <li>• TCP : 0 (passif) ou 2 (actif, valeur par défaut)</li> <li>• ISO sur TCP : 2 à 16</li> <li>• UDP : 0</li> </ul>
11	next_staddr_len	USInt	Ce paramètre n'est pas utilisé.

5.6 Configuration de la CPU pour la communication

Octet	Paramètre et type de données		Description
12 ... 27	local_tsap_id	Array [1..16] of Byte	<p>Elément d'adresse local de la liaison :</p> <ul style="list-style-type: none"> <li>TCP et ISO sur TCP : n° de port local (valeurs possibles : 1 à 49151 ; valeurs recommandées : 2000...5000): <ul style="list-style-type: none"> <li>local_tsap_id[1] = octet de poids fort du numéro de port en notation hexadécimale</li> <li>local_tsap_id[2] = octet de poids faible du numéro de port en notation hexadécimale</li> <li>local_tsap_id[3-16] = sans objet</li> </ul> </li> <li>ISO sur TCP : ID TSAP local : <ul style="list-style-type: none"> <li>local_tsap_id[1] = B#16#E0</li> <li>local_tsap_id[2] = châssis et emplacement du noeud d'extrémité local (bits 0 à 4 : numéro d'emplacement, bits 5 à 7 : numéro de châssis)</li> <li>local_tsap_id[3-16] = extension TSAP, facultative</li> </ul> </li> <li>UDP : Ce paramètre n'est pas utilisé.</li> </ul> <p>Remarque : Veillez à ce que chaque valeur de local_tsap_id soit unique dans la CPU.</p>
28 ... 33	rem_subnet_id	Array [1..6] of USInt	Ce paramètre n'est pas utilisé.
34 ... 39	rem_staddr	Array [1..6] of USInt	<p>TCP et ISO sur TCP uniquement : Adresse IP du noeud d'extrémité partenaire (sans objet pour les liaisons passives). L'adresse IP 192.168.002.003 est, par exemple, stockée dans les éléments suivants du tableau :</p> <p>rem_staddr[1] = 192  rem_staddr[2] = 168  rem_staddr[3] = 002  rem_staddr[4] = 003  rem_staddr[5-6]= sans objet</p>
40 ... 55	rem_tsap_id	Array [1..16] of Byte	<p>Elément d'adresse partenaire de la liaison :</p> <ul style="list-style-type: none"> <li>TCP : numéro de port partenaire. Plage : 1 à 49151 ; valeurs recommandées : 2000 à 5000) : <ul style="list-style-type: none"> <li>rem_tsap_id[1] = octet de poids fort du numéro de port en notation hexadécimale</li> <li>rem_tsap_id[2] = octet de poids faible du numéro de port en notation hexadécimale</li> <li>rem_tsap_id[3-16] = sans objet</li> </ul> </li> <li>ISO sur TCP : ID TSAP partenaire : <ul style="list-style-type: none"> <li>rem_tsap_id[1] = B#16#E0</li> <li>rem_tsap_id[2] = châssis et emplacement du noeud d'extrémité partenaire (bits 0 à 4 : numéro d'emplacement, bits 5 à 7 : numéro de châssis)</li> <li>rem_tsap_id[3-16] = extension TSAP, facultative</li> </ul> </li> <li>UDP : Ce paramètre n'est pas utilisé.</li> </ul>
56 ... 61	next_staddr	Array [1..6] of Byte	Ce paramètre n'est pas utilisé.
62 ... 63	spare	Word	Réservé : W#16#0000

**Voir aussi**

Configuration du routage local/partenaire (Page 136)

**5.6.4 Affectation d'adresses IP (Internet Protocol)****5.6.4.1 Affectation d'adresses IP à des consoles de programmation et des dispositifs réseau**

Si votre console de programmation utilise une carte adaptateur intégrée reliée à votre réseau local d'usine (et éventuellement à Internet), l'ID réseau de l'adresse IP et le masque de sous-réseau de votre CPU et de la carte adaptateur intégrée de la console de programmation doivent être exactement les mêmes. L'ID réseau est la première partie de l'adresse IP (trois premiers octets ; par exemple, **211.154.184.16**) qui détermine sur quel réseau IP vous vous trouvez. Le masque de sous-réseau a normalement la valeur **255.255.255.0** ; toutefois, comme votre ordinateur se trouve dans un réseau local d'usine, le masque de sous-réseau peut avoir diverses valeurs (par exemple, **255.255.254.0**) afin de configurer des sous-réseaux uniques. Lorsqu'il est combiné avec l'adresse IP d'appareil par une opération mathématique ET, le masque de sous-réseau définit les limites d'un sous-réseau IP.

---

**Remarque**

Pour un scénario Internet où les consoles de programmation, les dispositifs réseau et les routeurs IP communiqueront avec l'extérieur, il faut affecter des adresses IP uniques afin d'éviter des conflits avec d'autres utilisateurs du réseau. Adressez-vous au personnel du service informatique de votre entreprise qui connaît vos réseaux d'usine pour l'affectation de vos adresses IP.

---

Si votre console de programmation utilise une carte adaptateur Ethernet à USB reliée à un réseau isolé, l'ID réseau de l'adresse IP et le masque de sous-réseau de votre CPU et de la carte adaptateur Ethernet à USB de la console de programmation doivent être exactement les mêmes. L'ID réseau est la première partie de l'adresse IP (trois premiers octets ; par exemple, **211.154.184.16**) qui détermine sur quel réseau IP vous vous trouvez. Le masque de sous-réseau a normalement la valeur **255.255.255.0**. Lorsqu'il est combiné avec l'adresse IP d'appareil par une opération mathématique ET, le masque de sous-réseau définit les limites d'un sous-réseau IP.

---

**Remarque**

Une carte adaptateur Ethernet à USB est utile lorsque vous ne voulez pas que votre CPU soit sur votre réseau local d'entreprise. Cet agencement est particulièrement utile pendant le test initial et les tests de mise en service.

---

Tableau 5- 8 Affectation d'adresses Ethernet

Carte adaptateur de la console de programmation	Type de réseau	Adresse IP (Internet Protocol)	Masque de sous-réseau
Carte adaptateur intégrée	Connexion à votre réseau d'usine (et éventuellement à Internet)	L'ID réseau de votre CPU et de la carte adaptateur intégrée de la console de programmation doivent être exactement identiques. L'ID réseau est la première partie de l'adresse IP (trois premiers octets ; par exemple, <b>211.154.184</b> .16) qui détermine sur quel réseau IP vous trouvez.	Le masque de sous-réseau de votre CPU et de la carte adaptateur intégrée doivent être exactement les mêmes. Le masque de sous-réseau a normalement la valeur <b>255.255.255.0</b> ; toutefois, comme votre ordinateur se trouve dans un réseau local d'usine, le masque de sous-réseau peut avoir diverses valeurs (par exemple, <b>255.255.254.0</b> ) afin de configurer des sous-réseaux uniques. Lorsqu'il est combiné avec l'adresse IP d'appareil par une opération mathématique ET, le masque de sous-réseau définit les limites d'un sous-réseau IP.
Carte adaptateur Ethernet à USB	Connexion à un réseau isolé	L'ID réseau de votre CPU et de la carte adaptateur Ethernet à USB de la console de programmation doivent être exactement identiques. L'ID réseau est la première partie de l'adresse IP (trois premiers octets ; par exemple, <b>211.154.184</b> .16) qui détermine sur quel réseau IP vous trouvez.	Le masque de sous-réseau de votre CPU et de la carte adaptateur Ethernet à USB doivent être exactement les mêmes. Le masque de sous-réseau a normalement la valeur <b>255.255.255.0</b> . Lorsqu'il est combiné avec l'adresse IP d'appareil par une opération mathématique ET, le masque de sous-réseau définit les limites d'un sous-réseau IP.

**Affectation ou vérification de l'adresse IP de votre console de programmation via "Favoris réseau" (sur votre bureau)**

Vous pouvez affecter ou vérifier l'adresse IP de votre console de programmation à l'aide des sélections de menu suivantes :

- (Cliquez avec le bouton droit de la souris sur) "Favoris réseau"
- "Propriétés"
- (Cliquez avec le bouton droit de la souris sur) "Connexion au réseau local"
- "Propriétés"

Dans le champ "Cette connexion utilise les éléments suivants" de la boîte de dialogue "Propriétés de Connexion au réseau local", défilez jusqu'à "Protocole Internet (TCP/IP)". Cliquez sur "Protocole Internet (TCP/IP)" puis sur le bouton "Propriétés". Sélectionnez "Obtenir une adresse IP automatiquement (DHCP)" ou "Utiliser l'adresse IP suivante" (pour entrer une adresse IP statique).

### Remarque

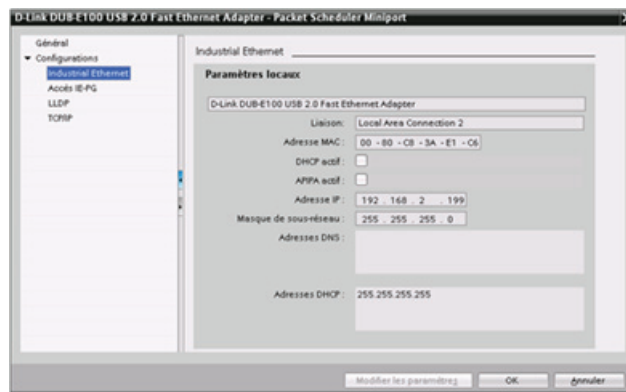
Le protocole DHCP (Dynamic Host Configuration Protocol) affecte automatiquement une adresse IP à votre console de programmation depuis le serveur DHCP à la mise sous tension.

## 5.6.4.2 Vérification de l'adresse IP de votre console de programmation

Vous pouvez vérifier les adresses MAC et IP de votre console de programmation à l'aide des sélections de menu suivantes :

1. Affichez le détail de "Accès en ligne" dans l'arborescence du projet.
2. Cliquez avec le bouton droit de la souris sur le réseau désiré et sélectionnez "Propriétés".
3. Dans la boîte de dialogue du réseau, affichez le détail de "Configurations" et sélectionnez "Industrial Ethernet".

Les adresses MAC et IP de la console de programmation s'affichent.

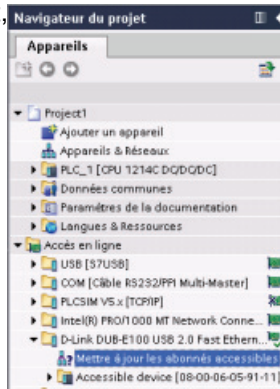


### 5.6.4.3 Affectation d'une adresse IP à une CPU en ligne

Vous pouvez affecter une adresse IP à un dispositif réseau en ligne. Cela s'avère particulièrement utile lors d'une configuration d'appareil initiale.

1. Dans l'arborescence du projet, vérifiez qu'aucune adresse IP n'est affectée à la CPU à l'aide des sélections de menu suivantes :

- "Accès en ligne"
- <Carte adaptateur pour le réseau dans lequel se trouve l'appareil>
- "Mettre à jour les abonnés accessibles"



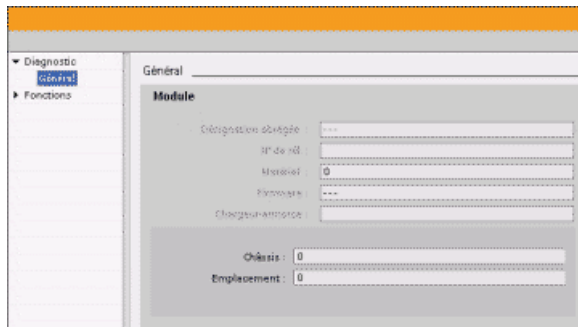
REMARQUE : Si une adresse MAC et non une adresse IP s'affiche, c'est qu'aucune adresse IP n'a été affectée.

2. Doublez-cliquez sur "En ligne & Diagnostic" sous l'abonné accessible concerné.



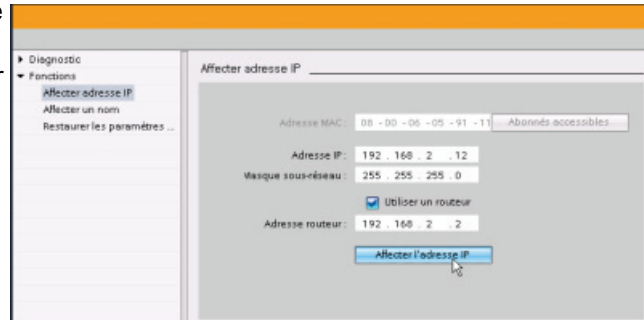
3. Faites les sélections de menu suivantes dans la boîte de dialogue "En ligne & Diagnostic" :

- "Fonctions"
- "Affecter adresse IP"



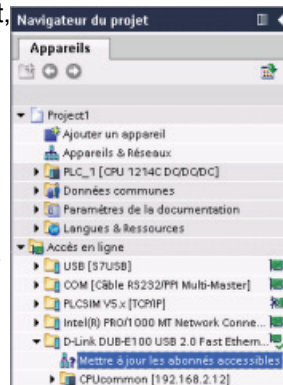


4. Entrez votre nouvelle adresse IP dans le champ "Adresse IP" et cliquez sur le bouton "Affecter adresse IP".



5. Dans l'arborescence du projet, vérifiez que votre nouvelle adresse IP a été affectée à la CPU à l'aide des sélections de menu suivantes :

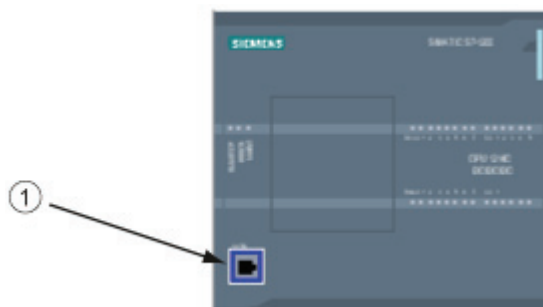
- "Accès en ligne"
- <Carte adaptateur pour le réseau dans lequel se trouve l'appareil>
- "Mettre à jour les abonnés accessibles"



#### 5.6.4.4 Configuration d'une adresse IP pour une CPU dans votre projet

##### Configuration de l'interface PROFINET

Pour configurer les paramètres pour l'interface PROFINET, sélectionnez le carré PROFINET vert sur la CPU. L'onglet "Propriétés" de la fenêtre d'inspection affiche le port PROFINET.



① Port PROFINET

## Configuration de l'adresse IP

**Adresse Ethernet (MAC) :** Dans un réseau PROFINET, chaque appareil reçoit une adresse MAC (Media Access Control : contrôle d'accès au support) du fabricant en vue de son identification. Une adresse MAC consiste en six groupes de deux chiffres hexadécimaux, séparés par un tiret (-) ou un deux-points (:), dans l'ordre de transmission (par exemple, 01-23-45-67-89-AB ou 01:23:45:67:89:AB).

**Adresse IP :** Chaque appareil doit également comporter une adresse IP (protocole Internet). Cette adresse permet à l'appareil de transmettre les données dans un réseau routé plus complexe.

Chaque adresse IP est divisée en quatre segments de 8 bits et est exprimée en format décimal avec séparation par des points (par exemple, 211.154.184.16). La première partie de l'adresse IP est utilisée pour l'ID de réseau (sur quel réseau est-on ?) et la deuxième partie de l'adresse correspond à l'ID d'hôte (unique pour chaque appareil dans le réseau). L'adresse IP 192.168.x.y est une désignation standard reconnue comme partie d'un réseau privé qui n'est pas routé sur Internet.

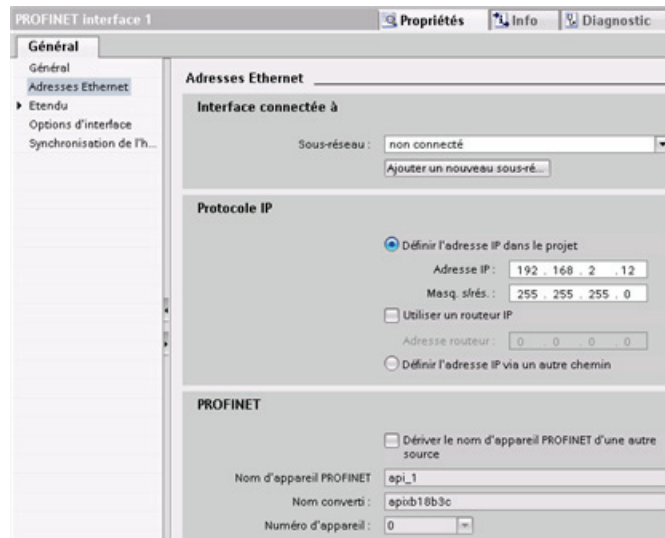
**Masque de sous-réseau :** Un sous-réseau est un groupement logique d'appareils réseau connectés. Les noeuds d'un sous-réseau tendent à être proches physiquement les uns des autres dans un réseau local. Un masque (appelé masque de sous-réseau ou masque de réseau) définit les frontières d'un sous-réseau IP.

Le masque de sous-réseau 255.255.255.0 convient généralement à un petit réseau local. Toutes les adresses IP dans ce réseau doivent avoir les trois mêmes premiers octets et les divers appareils dans ce réseau sont identifiés par le dernier octet (champ de 8 bits). Par exemple, on affectera le masque de sous-réseau 255.255.255.0 et les adresses IP 192.168.2.0 à 192.168.2.255 aux appareils dans un petit réseau local.

La seule liaison possible entre différents sous-réseaux se fait via un routeur. En présence de sous-réseaux, vous devez utiliser un routeur IP.

**Routeur IP :** Les routeurs constituent le lien entre réseaux locaux. Un ordinateur dans un réseau local peut envoyer par le biais d'un routeur des messages à d'autres réseaux qui peuvent abriter d'autres réseaux locaux. Si la destination des données ne se trouve pas à l'intérieur du réseau local, le routeur transmet les données à un autre réseau ou groupe de réseaux où les données peuvent être acheminées à destination.

Les routeurs se basent sur les adresses IP pour transmettre et recevoir des paquets de données.



**Propriétés des adresses IP** : Dans la fenêtre Propriétés, sélectionnez l'entrée de configuration "Adresses Ethernet". STEP 7 affiche la boîte de dialogue de configuration de l'adresse Ethernet qui associe le projet logiciel à l'adresse IP de la CPU qui recevra ce projet.

Tableau 5- 9 Paramètres pour l'adresse IP

Paramètre	Description	
Sous-réseau	Nom du sous-réseau auquel l'appareil est connecté. Cliquez sur le bouton "Ajouter nouveau sous-réseau" pour créer un nouveau sous-réseau. "non connecté" est la valeur par défaut. Deux types de connexion sont possibles : <ul style="list-style-type: none"> <li>• La valeur par défaut "non connecté" fournit une liaison locale.</li> <li>• Un sous-réseau est nécessaire lorsque votre réseau contient au moins deux appareils.</li> </ul>	
Protocole IP	Adresse IP	Adresse IP affectée à la CPU
	Masque de sous-réseau	Masque de sous-réseau affecté
	Utiliser un routeur IP	Cochez cette case pour indiquer qu'un routeur IP est utilisé.
	Adresse du routeur	Adresse IP affectée au routeur, le cas échéant

### Remarque

Toutes les adresses IP sont configurées lorsque vous chargez le projet dans la CPU. Si la CPU ne possède pas d'adresse IP préconfigurée, vous devez associer le projet à l'adresse MAC de l'appareil cible. Si votre CPU est connectée à un routeur sur un réseau, vous devez aussi entrer l'adresse IP du routeur.

Le bouton radio "Configurer l'adresse IP autrement" vous permet de modifier l'adresse IP en ligne ou à l'aide de l'instruction "T\_CONFIG" après le chargement du programme. La méthode d'affectation de l'adresse IP concerne la CPU uniquement.

 **ATTENTION**

Après chargement dans la CPU d'une configuration matérielle avec l'option "Configurer l'adresse IP autrement" activée, il n'est pas possible de faire passer la CPU de l'état de fonctionnement MARCHE à l'état ARRET ou de l'état ARRET à l'état MARCHE.

L'équipement utilisateur continuera à fonctionner dans ces conditions et pourra provoquer un fonctionnement inattendu des machines ou du processus, pouvant entraîner la mort, des blessures graves et des dommages matériels importants en l'absence de précautions appropriées.

Vous devez donc veiller à que l'adresse ou les adresses IP de votre CPU soient définies avant d'utiliser la CPU dans un environnement d'automatisation réel. Pour ce faire, vous pouvez utiliser le progiciel de programmation STEP 7, l'outil S7-1200 ou un appareil IHM raccordé en combinaison avec l'instruction T\_CONFIG.

 **ATTENTION**

La modification de l'adresse IP d'une CPU en ligne ou à partir du programme utilisateur peut créer une situation dans laquelle le réseau PROFINET s'arrête.

Si l'adresse IP d'une CPU est changée en une adresse IP en dehors du sous-réseau, le réseau PROFINET perdra la communication et l'échange de données sera interrompu. L'équipement utilisateur pourrait être configuré pour continuer à fonctionner dans ces conditions. La perte de la communication PROFINET peut provoquer un fonctionnement inattendu des machines ou du processus, pouvant entraîner la mort, des blessures graves et des dommages matériels importants en l'absence de précautions appropriées.

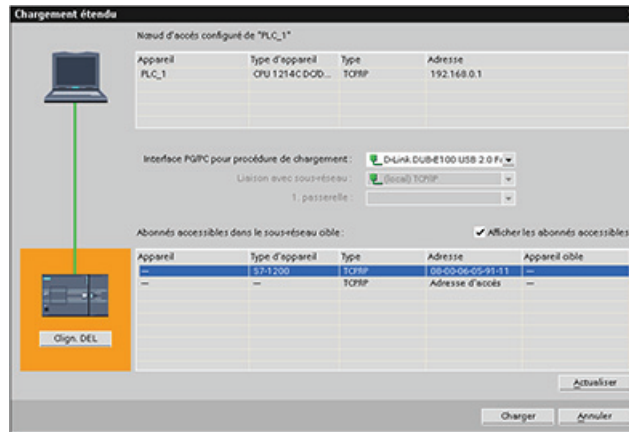
Si une adresse IP doit être modifiée à la main, veillez à ce que la nouvelle adresse IP se situe à l'intérieur du sous-réseau.

**Voir aussi**

T\_CONFIG (Page 480)

### 5.6.5 Test du réseau PROFINET

Une fois la configuration achevée, chargez le projet (Page 179) dans la CPU. Toutes les adresses IP sont configurées lorsque vous chargez le projet dans la CPU.



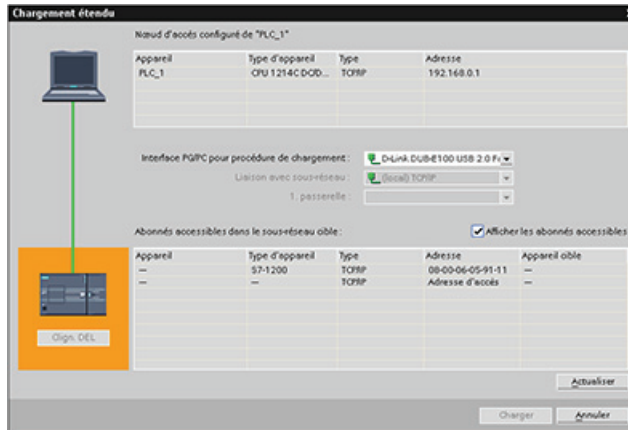
### Affectation d'une adresse IP à un appareil en ligne

La CPU S7-1200 ne dispose pas d'une adresse IP préconfigurée. Vous devez affecter manuellement une adresse IP à la CPU :

- Pour affecter une adresse IP à un appareil en ligne, reportez-vous à "Configuration des appareils, Affectation d'une adresse IP à une CPU en ligne" (Page 144) où vous trouverez la marche à suivre étape par étape.
- Pour affecter une adresse IP dans votre projet, vous devez configurer l'adresse IP dans la configuration des appareils, sauvegarder la configuration et la charger dans l'automate. Reportez-vous à "Configuration des appareils, Configuration d'une adresse IP pour une CPU dans votre projet" (Page 145) pour plus d'informations.

### Utilisation de la boîte de dialogue "Chargement élargi dans l'appareil" pour recherche les dispositifs réseau connectés

La boîte de dialogue "Chargement élargi dans l'appareil" de la fonction "Charger dans l'appareil" de la CPU S7-1200 peut montrer tous les dispositifs réseau accessibles en précisant si des adresses IP uniques ont été affectées ou non à tous les appareils. Cochez la case "Afficher les abonnés accessibles" pour afficher tous les appareils accessibles et disponibles avec les adresses MAC ou IP qui leur ont été affectées.



Si le dispositif réseau requis n'est pas dans la liste, c'est que la communication avec ce dispositif a été interrompue pour une certaine raison. Vous devez alors rechercher d'éventuelles erreurs de matériel et/ou de configuration dans cet appareil et ce réseau.

### 5.6.6 Localisation de l'adresse Ethernet (MAC) sur la CPU

Dans les réseaux PROFINET, une adresse MAC (Media Access Control) est un identificateur que le fabricant attribue à l'interface de réseau aux fins d'identification. Une adresse MAC code généralement le numéro d'identification déposé du fabricant.

Le format standard (IEEE 802.3) pour l'impression des adresses MAC sous forme conviviale consiste en six groupes de deux chiffres hexadécimaux, séparés par un tiret (-) ou un deux-points (:), dans l'ordre de transmission (par exemple, 01-23-45-67-89-ab ou 01:23:45:67:89:ab).

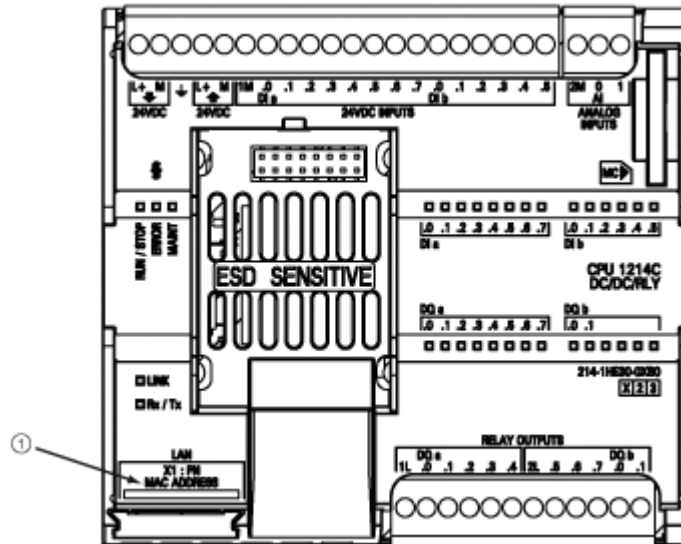
---

#### Remarque

Chaque CPU reçoit en usine une adresse MAC unique permanente. Vous ne pouvez pas modifier l'adresse MAC d'une CPU.

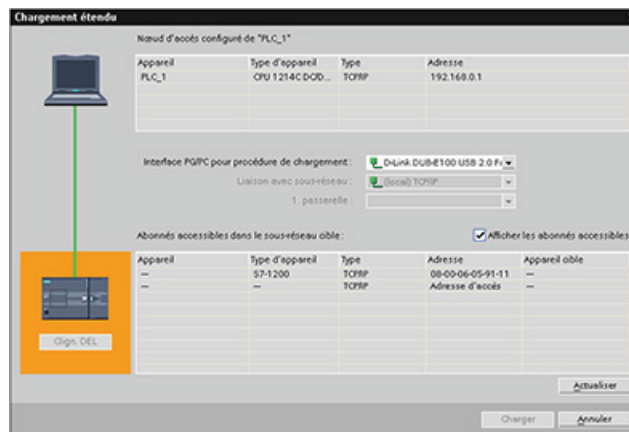
---

L'adresse MAC est imprimée sur le coin inférieur gauche à l'avant de la CPU. Vous devez soulever le volet inférieur pour voir cette adresse MAC.



① Adresse MAC

Au départ, la CPU n'a pas d'adresse IP mais uniquement une adresse MAC reçue en usine. La communication PROFINET exige qu'une adresse IP unique soit affectée à tous les appareils.



Utilisez la fonction "Charger dans l'appareil" de la CPU et la boîte de dialogue "Chargement élargi dans l'appareil" pour visualiser tous les appareils réseau accessibles et vous assurer que des adresses IP uniques ont été affectées à tous les appareils. Cette boîte de dialogue affiche tous les appareils accessibles et disponibles avec les adresses MAC ou IP qui leur ont été affectées. Les adresses MAC sont capitales pour identifier les appareils auxquels manque l'adresse IP unique exigée.

## 5.6.7 Configuration de la synchronisation via le protocole NTP

Le protocole NTP (Network Time Protocol) est largement utilisé pour synchroniser les horloges des systèmes informatiques avec des serveurs de temps Internet. En mode NTP, le CP envoie des requêtes d'heure à intervalles réguliers (en mode client) au serveur NTP dans le sous-réseau (local). L'heure la plus précise et la plus fiable possible est calculée en fonction des réponses du serveur et l'horloge de la station est synchronisée.

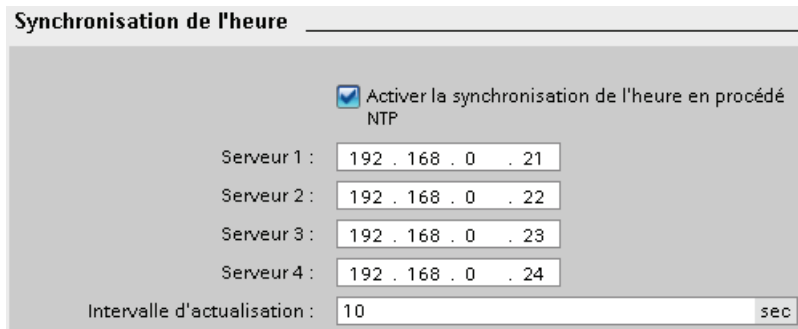
5.6 Configuration de la CPU pour la communication

L'avantage de ce mode est qu'il permet la synchronisation de l'heure au-delà des limites de sous-réseaux.

Il faut configurer les adresses IP de quatre serveurs NTP au maximum. L'intervalle de mise à jour définit l'intervalle (en secondes) entre les requêtes d'heure. La valeur de l'intervalle doit être comprise entre 10 secondes et un jour.

En mode NTP, c'est généralement le temps UTC (temps universel coordonné) - qui correspond au temps moyen de Greenwich (GMT) - qui est transmis.

Dans la fenêtre Propriétés, sélectionnez l'entrée de configuration "Synchronisation de l'heure". STEP 7 affiche la boîte de dialogue Synchronisation de l'heure :



**Remarque**

Toutes les adresses IP sont configurées lorsque vous chargez le projet dans la CPU.

Tableau 5- 10 Paramètres pour la synchronisation de l'heure

Paramètre	Définition
Activer la synchronisation de l'heure via le serveur NTP	Cochez la case pour activer la synchronisation de l'heure via des serveurs NTP.
Serveur 1	Adresse IP affectée au serveur de temps réseau 1
Serveur 2	Adresse IP affectée au serveur de temps réseau 2
Serveur 3	Adresse IP affectée au serveur de temps réseau 3
Serveur 4	Adresse IP affectée au serveur de temps réseau 4
Intervalle de mise à jour	Valeur de l'intervalle (en s)

**5.6.8 Temps de mise en route, affectation de nom et d'adresse pour un appareil PROFINET**

PROFINET IO peut allonger le temps de mise en route de votre système (délai d'attente configurable). Un plus grand nombre d'appareils et des appareils lents influent sur la quantité de temps nécessaire pour passer à l'état MARCHE.



Votre réseau PROFINET S7-1200 peut comporter les nombres maximum suivants de périphériques PROFINET IO :

- 16 périphériques IO au maximum dans V3.0.
- 8 périphériques IO au maximum dans V2.2.

Chaque station (ou périphérique IO) démarre de manière indépendante à la mise en route, ce qui influe sur le temps de mise en route total de la CPU. Si vous donnez une valeur trop faible au délai d'attente configurable, le temps de mise en route pourrait ne pas être suffisant pour que toutes les stations mènent à bien leur démarrage. Si cette situation survient, des erreurs de station erronées seront signalées.

Vous trouverez le "temps d'affectation des paramètres pour la périphérie décentralisée" (délai d'attente) dans les propriétés de la CPU, sous "Mise en route". Le délai d'attente configurable par défaut est de 60 000 ms (1 minute) ; il est configurable par l'utilisateur.

### Affectation de nom et d'adresse pour un appareil PROFINET dans STEP 7

Tous les appareils PROFINET **doivent** avoir un nom d'appareil et une adresse IP. Servez-vous de STEP 7 pour définir les noms d'appareils et configurer les adresses IP. Le téléchargement des noms d'appareils dans les périphériques IO se fait à l'aide de PROFINET DCP (Discovery and Configuration Protocol).

### Affectation de l'adresse PROFINET au démarrage du système

Le contrôleur diffuse les noms des appareils sur le réseau et les appareils répondent avec leur adresse MAC. Le contrôleur affecte alors une adresse IP à l'appareil à l'aide du protocole DCP PROFINET :

- Si l'adresse MAC a une adresse IP configurée, la station exécute un démarrage.
- Si l'adresse MAC n'a pas d'adresse IP configurée, STEP 7 affecte l'adresse qui est configurée dans le projet et la station exécute alors un démarrage.
- En cas de problème lors de cette opération, une erreur de station survient et il n'y a pas de démarrage. Cette situation provoque le dépassement de la valeur de délai d'attente configurable.



## Concepts de programmation

### 6.1 Principes de conception d'un système d'automatisation

Lorsque vous concevez un système d'automatisation, vous devez choisir parmi diverses méthodes et divers critères. Les principes généraux suivants sont valables pour de nombreux projets de conception. Il vous faudra, bien sûr, suivre les procédures en vigueur dans votre société ainsi que les usages découlant de votre formation et de votre lieu de travail.

Tableau 6- 1 Principes de conception d'un système d'automatisation

Étapes recommandées	Tâches
Subdivision de votre processus ou de votre installation en unités	Segmentez votre processus ou votre installation en parties indépendantes les unes des autres. Ces subdivisions déterminent les limites entre les automates et ont une influence sur les descriptions fonctionnelles et l'affectation des ressources.
Création des descriptions fonctionnelles	Décrivez le fonctionnement de chaque partie du processus ou de l'installation, par exemple entrées/sorties, description fonctionnelle de l'exploitation, états devant être atteints avant de permettre l'actionnement des actionneurs (tels que valves solénoïdes, moteurs et entraînements), description de l'interface opérateur et toute interface avec d'autres sections du processus ou de l'installation.
Conception des circuits de sécurité	<p>Identifiez tout équipement pouvant nécessiter de la logique câblée pour la sécurité. N'oubliez pas que des défaillances dangereuses peuvent se produire dans les appareils de commande, ce qui peut entraîner un démarrage ou un changement inattendu dans le fonctionnement d'une machine. Lorsqu'il existe alors des risques de blessures sur les personnes ou de dommages matériels importants, songez à implémenter des dispositifs de sécurité électromécaniques prioritaires (qui opèrent indépendamment de l'automate) afin d'éviter des dysfonctionnements dangereux. La conception des circuits de sécurité comprend les tâches suivantes :</p> <ul style="list-style-type: none"> <li>• Identifiez tout fonctionnement incorrect ou inattendu des actionneurs qui pourrait être dangereux.</li> <li>• Identifiez les conditions qui assurent que l'exploitation est sans danger et déterminez comment détecter ces conditions indépendamment de l'automate.</li> <li>• Identifiez comment l'automate affecte le processus lorsque la tension est appliquée puis coupée et identifiez également comment et quand des erreurs sont détectées. Servez-vous de ces informations uniquement pour la conception des fonctionnements normaux et anormaux prévisibles. Vous ne devez pas vous baser sur ce scénario du "meilleur des cas" pour les problèmes de sécurité.</li> <li>• Concevez les dispositifs de sécurité manuels ou électromécaniques prioritaires qui bloquent les fonctionnements dangereux indépendamment de l'automate.</li> <li>• Fournissez à l'automate les informations d'état appropriées en provenance des circuits indépendants afin que le programme et toute interface opérateur disposent des informations nécessaires.</li> <li>• Identifiez toute autre mesure de sécurité nécessaire à un déroulement sûr du processus.</li> </ul>
Planification de la sécurité du système	Déterminez le niveau de protection (Page 175) nécessaire pour accéder à votre processus. Vous pouvez protéger les CPU et les blocs de programme par un mot de passe afin d'empêcher les accès non autorisés.

Étapes recommandées	Tâches
Définition des postes d'opération	<p>Créez les schémas suivants des postes d'opération, basés sur les exigences des descriptions fonctionnelles :</p> <ul style="list-style-type: none"> <li>• Schéma d'ensemble indiquant l'emplacement de chaque poste d'opération par rapport au processus ou à l'installation</li> <li>• Schéma mécanique des dispositifs pour le poste d'opération, tels qu'afficheurs, commutateurs et lampes</li> <li>• Schémas électriques avec les E/S associées de l'automate et des modules d'entrées-sorties</li> </ul>
Création des schémas de configuration	<p>Créez des schémas de configuration de l'équipement de commande basés sur les exigences des descriptions fonctionnelles :</p> <ul style="list-style-type: none"> <li>• Schéma d'ensemble indiquant l'emplacement de chaque automate par rapport au processus ou à l'installation</li> <li>• Schéma mécanique de chaque automate et de tout module d'E/S incluant les armoires et autres équipements</li> <li>• Schémas électriques pour chaque automate et module d'E/S incluant les numéros de référence des appareils, les adresses de communication et les adresses d'E/S</li> </ul>
Création d'une liste de mnémoniques	<p>Créez une liste de mnémoniques avec les adresses absolues. N'indiquez pas seulement les signaux d'E/S physiques, mais également les autres éléments (tels que les noms de variables) qui seront utilisés dans votre programme.</p>

## 6.2 Organisation de votre programme utilisateur

Lorsque vous créez un programme utilisateur pour les tâches d'automatisation, vous insérez les instructions pour le programme dans des blocs de code :

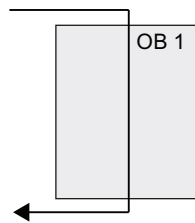
- Un bloc d'organisation (OB) réagit à un événement spécifique dans la CPU et peut interrompre l'exécution du programme utilisateur. Le bloc par défaut pour l'exécution cyclique du programme utilisateur (OB 1) fournit la structure de base pour votre programme utilisateur et est le seul bloc de code obligatoire pour un programme utilisateur. Si vous incluez d'autres OB dans votre programme, ces OB interrompent l'exécution de l'OB 1. Les autres OB exécutent des fonctions spécifiques, telles que des tâches de démarrage, la gestion des alarmes et des erreurs ou l'exécution d'un code spécifique à intervalles spécifiques.
- Un bloc fonctionnel (FB) est un sous-programme qui est exécuté lorsqu'il est appelé dans un autre bloc de code (OB, FB ou FC). Le bloc appelant transmet des paramètres au FB et identifie également un bloc de données spécifique (DB) qui contient les données pour l'appel spécifique ou instance de ce FB. Changer le DB d'instance permet à un FB générique de commander le fonctionnement d'un ensemble d'appareils. Ainsi, par exemple, un FB peut piloter plusieurs pompes ou vannes avec des DB d'instance différents contenant les paramètres de fonctionnement spécifiques de chaque pompe ou vanne.
- Une fonction (FC) est un sous-programme qui est exécuté lorsqu'il est appelé dans un autre bloc de code (OB, FB ou FC). Une FC ne comporte pas de DB d'instance associé. Le bloc appelant transmet des paramètres à la FC. Les valeurs de sortie de la FC doivent être écrites dans une adresse de mémoire ou dans un DB global.

## Choix du type de structure pour votre programme utilisateur

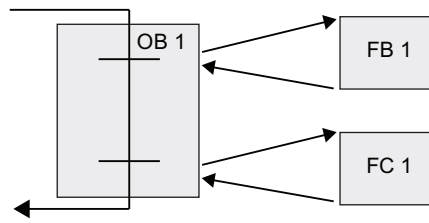
Selon les besoins de votre application, vous pouvez choisir soit une structure linéaire soit une structure modulaire pour votre programme utilisateur :

- Un programme linéaire exécute toutes les instructions pour vos tâches d'automatisation séquentiellement les unes après les autres. Avec un programme linéaire, vous placerez typiquement toutes les instructions dans l'OB d'exécution cyclique du programme (OB 1).
- Un programme modulaire appelle des blocs de code spécifiques qui exécutent des tâches spécifiques. Pour créer une structure modulaire, vous divisez la tâche d'automatisation complexe en petites tâches subordonnées qui correspondent aux fonctions technologiques du processus. Chaque bloc de code fournit le segment de programme pour une tâche subordonnée. Vous structurez votre programme en appelant l'un des blocs de code à partir d'un autre bloc.

Structure linéaire :



Structure modulaire :



En créant des blocs de code génériques pouvant être réutilisés à l'intérieur du programme utilisateur, vous simplifiez la conception et l'implémentation du programme utilisateur.

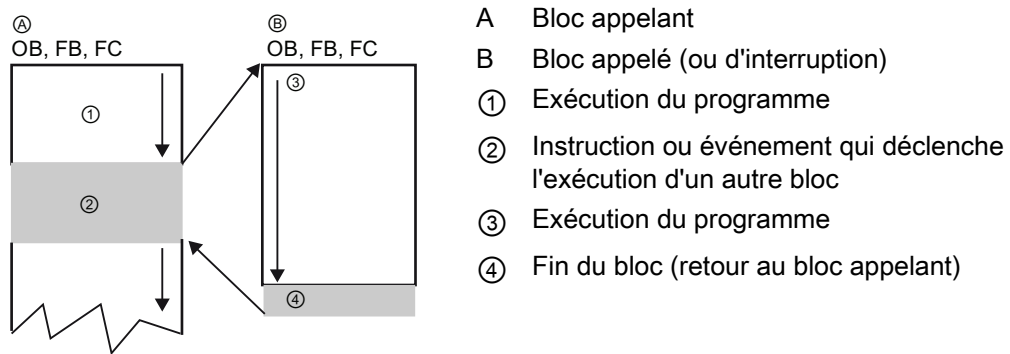
L'utilisation de blocs de code génériques présente un certain nombre d'avantages :

- Vous pouvez créer des blocs de code réutilisables pour des tâches standard, telles que la commande d'une pompe ou d'un moteur. Vous pouvez également stocker ces blocs de code génériques dans une bibliothèque qui peut être utilisée par d'autres applications ou solutions.
- Lorsque vous structurez le programme utilisateur en composants modulaires liés à des tâches fonctionnelles, votre programme devient plus facile à comprendre et à gérer. Non seulement les composants modulaires vous aident à standardiser la conception du programme, mais ils rendent également l'actualisation ou la modification du code plus rapide et plus facile.
- La création de composants modulaires simplifie le débogage de votre programme. En structurant le programme complet en tant qu'ensemble de segments de programme modulaires, vous pouvez tester la fonctionnalité de chaque bloc de code dès qu'il est élaboré.
- La création de composants modulaires liés à des fonctions technologiques spécifiques permet de simplifier et d'accélérer la mise en service de l'application achevée.

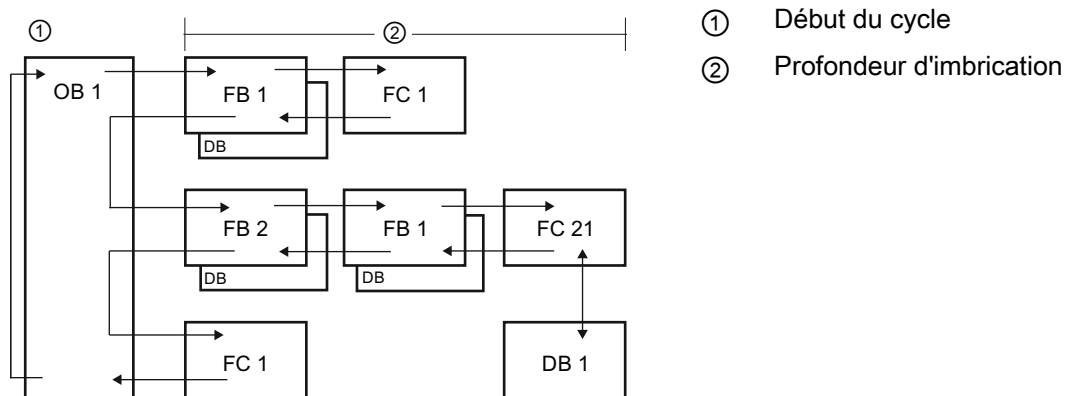
### 6.3 Utilisation de blocs pour structurer votre programme

En concevant des FB et des FC qui exécutent des tâches génériques, vous créez des blocs de code modulaires. Vous structurez ensuite votre programme en faisant appeler ces modules réutilisables par d'autres blocs de code. Le bloc appelant transmet des paramètres spécifiques de l'appareil au bloc appelé.

Lorsqu'un bloc de code appelle un autre bloc de code, la CPU exécute le code du programme dans le bloc appelé. Une fois l'exécution du bloc appelé achevée, la CPU reprend l'exécution du bloc appelant. Le traitement se poursuit par l'exécution de l'instruction qui suit l'appel de bloc.



Vous pouvez imbriquer les appels de bloc pour obtenir une structure plus modulaire. Dans l'exemple suivant, la profondeur d'imbrication est 4 : l'OB de cycle de programme plus 3 niveaux d'appels de blocs de code.



### 6.3.1 Bloc d'organisation (OB)

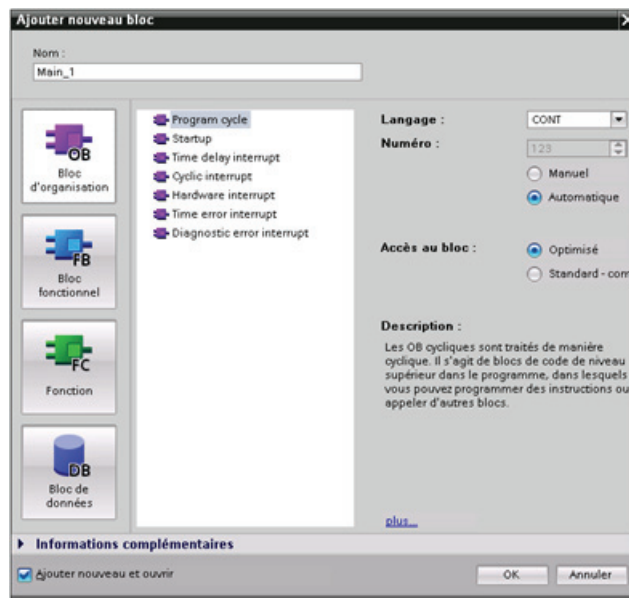
Les blocs d'organisation permettent de structurer votre programme. Ils servent d'interface entre le système d'exploitation et le programme utilisateur. Les OB sont déclenchés sur événement. La CPU exécute un OB suite à un événement, tel qu'une alarme de diagnostic ou un intervalle de temps. Certains OB ont des événements déclencheurs et un comportement prédéfinis.

L'OB du cycle de programme contient votre programme principal. Vous pouvez inclure plus d'un OB de cycle de programme dans votre programme utilisateur. A l'état MARCHE, les OB de cycle de programme s'exécutent au niveau de priorité le plus bas et peuvent être interrompus par tous les autres types de traitement de programme. L'OB de démarrage n'interrompt pas l'OB de cycle de programme, car la CPU exécute l'OB de démarrage avant de passer à l'état MARCHE.

Une fois le traitement des OB de cycle de programme achevé, la CPU réexécute immédiatement les OB de cycle de programme. Ce traitement cyclique est le type "normal" de traitement utilisé par les automates programmables. Pour de nombreuses applications, le programme utilisateur entier se trouve dans un OB de cycle de programme unique.

Vous pouvez créer d'autres OB qui exécuteront des fonctions spécifiques, telles que la gestion des alarmes et des erreurs ou l'exécution d'un code spécifique à intervalles spécifiques. Ces OB interrompent l'exécution des OB du cycle de programme.

Utilisez la boîte de dialogue "Ajouter nouveau bloc" pour créer de nouveaux OB dans votre programme utilisateur.



La gestion des alarmes se fait toujours sur événement. Lorsqu'un tel événement se produit, la CPU interrompt l'exécution du programme utilisateur et appelle l'OB qui a été configuré pour gérer cet événement. Une fois l'exécution de l'OB d'interruption achevée, la CPU reprend l'exécution du programme utilisateur à l'endroit où elle s'était interrompue.

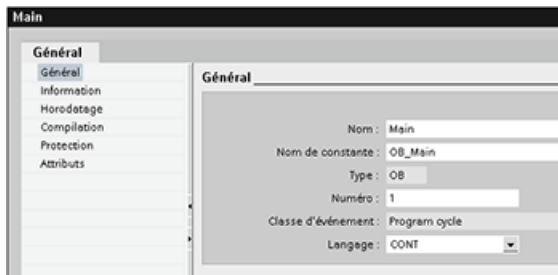
La CPU détermine l'ordre de traitement des événements d'alarme au moyen d'une priorité affectée à chaque OB. Chaque événement a une priorité de prise en charge particulière. Le niveau de priorité respectif à l'intérieur d'une classe de priorité détermine l'ordre dans lequel les OB sont exécutés. Plusieurs événements d'alarme peuvent être combinés dans des classes de priorité. Pour plus d'informations, reportez-vous à Concepts d'automatisation, Exécution du programme utilisateur (Page 71).

### Création d'un OB supplémentaire à l'intérieur d'une classe d'OB

Vous pouvez créer plusieurs OB pour votre programme utilisateur, et ce même pour les classes OB de cycle de programme et OB de démarrage. Utilisez la boîte de dialogue "Ajouter nouveau bloc" pour créer un OB. Entrez le nom de votre OB et indiquez un numéro d'OB supérieur ou égal à 200.

Si vous créez plusieurs OB de cycle de programme pour votre programme utilisateur, la CPU exécute chaque OB de cycle de programme par ordre numérique, en commençant par l'OB de cycle de programme de plus petit numéro (l'OB 1, par exemple). Par exemple, une fois le premier OB de cycle de programme (tel l'OB 1) achevé, la CPU exécute l'OB de cycle de programme de numéro immédiatement supérieur (tel l'OB 200).

### Configuration du fonctionnement d'un OB



Vous pouvez modifier les paramètres de fonctionnement pour un OB. Vous pouvez, par exemple, configurer le paramètre de temps pour un OB d'alarme temporisée ou pour un OB d'alarme cyclique.

### 6.3.2 Fonction (FC)

Une fonction (FC) est un bloc de code qui exécute typiquement une opération spécifique sur un ensemble de valeurs d'entrée. La FC stocke les résultats de cette opération dans des adresses de mémoire. Vous avez p. ex. recours à des FC pour effectuer des opérations standard et réutilisables (telles que des calculs mathématiques) ou des fonctions technologiques (telles que des contrôles individuels utilisant des opérations logiques sur bits) Une FC peut également être appelée plusieurs fois en différents points d'un programme. Cette réutilisation simplifie la programmation de tâches revenant souvent.

Une FC ne comporte pas de bloc de données (DB) d'instance associé. La FC se sert de la pile des données locales pour les données temporaires utilisées pour effectuer l'opération. Les données temporaires ne sont pas sauvegardées. Pour stocker les données de manière permanente, affectez la valeur de sortie à une adresse de mémoire globale, un memento M ou un DB global par exemple.



### 6.3.3 Bloc fonctionnel (FB)

Un bloc fonctionnel (FB) est un bloc de code qui utilise un bloc de données d'instance pour ses paramètres et ses données statiques. Les FB ont une mémoire de variables qui se situe dans un bloc de données (DB) appelé DB d'instance. Le DB d'instance fournit un bloc de mémoire qui est associé à cette instance (ou appel) du FB et qui contient les données une fois le FB achevé. Vous pouvez associer différents DB d'instance à différents appels du FB. Les DB d'instance vous permettent d'utiliser un FB générique pour commander plusieurs appareils. Vous structurez votre programme en insérant dans un bloc de code l'appel d'un FB et d'un DB d'instance. La CPU exécute alors le code dans ce FB et sauvegarde les paramètres du bloc et les données locales statiques dans le DB d'instance. Une fois l'exécution du FB achevée, la CPU revient au bloc de code qui a appelé le FB. Le DB d'instance conserve les valeurs pour cette instance du FB. Ces valeurs sont disponibles pour des appels ultérieurs du bloc fonctionnel soit dans le même cycle, soit dans d'autres cycles.

#### Blocs de code réutilisables à mémoire associée

Vous utiliserez typiquement un FB pour commander l'exécution de tâches ou le fonctionnement d'appareils qui durent plus d'un cycle. Pour stocker les paramètres de fonctionnement afin qu'ils soient rapidement accessibles d'un cycle au suivant, chaque FB dans votre programme utilisateur comporte un ou plusieurs DB d'instance. Lorsque vous appelez un FB, vous indiquez aussi un DB d'instance qui contient les paramètres du bloc et les données locales statiques pour cet appel ou "instance" du FB. Le DB d'instance conserve ces valeurs après que l'exécution du FB s'est achevée.

En concevant le FB pour des tâches de commande génériques, vous pouvez le réutiliser pour plusieurs appareils en sélectionnant des DB d'instance différents pour différents appels du FB.

Un FB sauvegarde les paramètres d'entrée, de sortie et d'entrée/sortie et les paramètres statiques dans un DB d'instance.

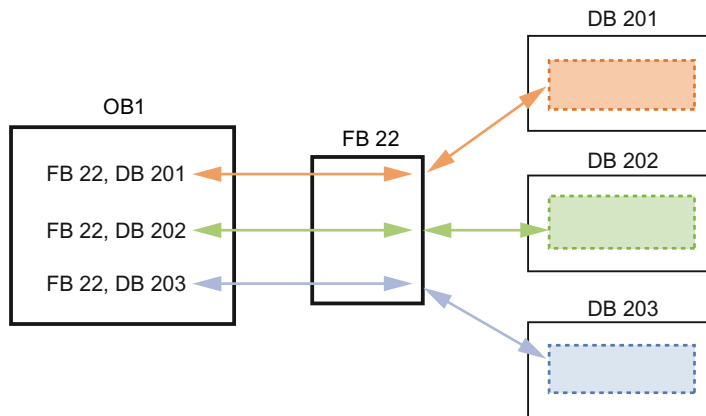
#### Définition de la valeur initiale dans le DB d'instance

Le DB d'instance comporte une valeur par défaut et une valeur initiale pour chaque paramètre. La valeur initiale fournit la valeur devant être utilisée lors de l'exécution du FB. La valeur initiale peut être modifiée lors de l'exécution de votre programme utilisateur.

L'interface de FB comporte également une colonne "Valeur par défaut" qui vous permet de définir une nouvelle valeur initiale pour le paramètre lors de l'écriture du code de programme. Cette valeur par défaut dans le FB est ensuite transférée dans la valeur initiale du DB d'instance associé. Si vous ne définissez pas de nouvelle valeur initiale pour un paramètre dans l'interface de FB, la valeur par défaut du DB d'instance est copiée dans la valeur initiale.

### Utilisation d'un FB unique avec plusieurs DB

La figure suivante montre un OB qui appelle un FB trois fois, avec un bloc de données différent à chaque appel. Cette structure permet à un FB générique de commander plusieurs appareils similaires, tels des moteurs, en affectant un bloc de données d'instance différent à chaque appel pour les différents appareils. Chaque DB d'instance mémorise les données (par exemple, la vitesse, le temps d'accélération et le temps de fonctionnement total) pour un appareil distinct.



Dans cet exemple, le FB 22 commande trois appareils distincts, le DB 201 contenant les données de fonctionnement du premier appareil, le DB 202 celles du deuxième appareil et le DB 203 celles du troisième appareil.

### 6.3.4 Bloc de données (DB)

Vous créez des blocs de données (DB) dans votre programme utilisateur pour sauvegarder les données des blocs de code. Tous les blocs de programme dans le programme utilisateur peuvent accéder aux données d'un DB global. En revanche, un DB d'instance contient des données pour un bloc fonctionnel spécifique (FB).

Les données sauvegardées dans un DB ne sont pas effacées à la fin de l'exécution du bloc de code associé. Il existe deux types de DB :

- Un DB global contient des données pour les blocs de code dans votre programme. Tous les OB, FB et FC peuvent accéder aux données dans un DB global.
- Un DB d'instance contient les données d'un FB spécifique. La structure des données dans un DB d'instance reflète les paramètres (Input, Output et InOut) et les données statiques du FB. La mémoire Temp pour le FB n'est pas stockée dans le DB d'instance.

---

#### Remarque

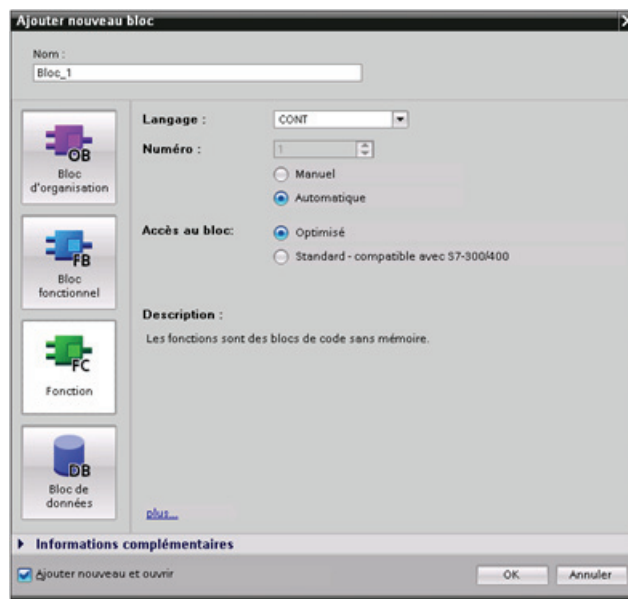
Bien que le DB d'instance reflète les données d'un FB spécifique, tout bloc de code peut accéder aux données à l'intérieur d'un DB d'instance.

---

Vous pouvez configurer un DB comme étant en lecture seule :

1. Cliquez avec le bouton droit de la souris sur le DB dans le navigateur du projet et sélectionnez "Propriétés" dans le menu contextuel.
2. Sélectionnez "Attributs" dans la boîte de dialogue "Propriétés".
3. Sélectionnez l'option "Bloc de données protégé en écriture dans l'appareil" et cliquez sur "OK".

## Création de blocs de code réutilisables



Utilisez la boîte de dialogue "Ajouter nouveau bloc" sous "Blocs de programme" dans le navigateur du projet pour créer des OB, des FB, des FC et des DB globaux.

Lors de la création d'un bloc de code, vous sélectionnez le langage de programmation pour ce bloc. Vous ne sélectionnez pas de langage pour les DB qui servent uniquement au stockage de données.

## 6.4 Comprendre le concept de cohérence des données

La CPU assure la cohérence des données pour tous les types de données simples (par exemple, Word et DWord) et pour toutes les structures système (par exemple, IEC\_TIMERS et DTL). La lecture ou l'écriture de la valeur ne peuvent pas être interrompues. Ainsi, la CPU protège l'accès à une valeur double mot jusqu'à ce que les quatre octets du double mot aient été lus ou écrits. Pour garantir que les OB de cycle de programme et les OB d'alarme n'écrivent pas en même temps dans la même adresse de mémoire, la CPU n'exécute pas d'OB d'alarme tant que l'opération de lecture ou d'écriture dans l'OB de cycle de programme n'est pas achevée.

Si votre programme utilisateur partage plusieurs valeurs en mémoire entre un OB de cycle de programme et un OB d'alarme, votre programme utilisateur doit également s'assurer que ces valeurs sont modifiées ou lues de manière cohérente. Vous pouvez utiliser les instructions DIS\_AIRT (Inhiber le traitement des alarmes) et EN\_AIRT (Valider le traitement des alarmes) dans votre OB de cycle de programme pour protéger les accès aux valeurs partagées.

- Insérez une instruction DIS\_AIRT dans le bloc de code pour garantir qu'aucun OB d'alarme ne peut être exécuté pendant l'opération de lecture ou d'écriture.
- Insérez les instructions qui lisent ou écrivent les valeurs qui pourraient être modifiées par un OB d'alarme.
- Insérez une instruction EN\_AIRT à la fin de la séquence pour annuler DIS\_AIRT et permettre l'exécution de l'OB d'alarme.

Une demande de communication émise par un appareil IHM ou une autre CPU peut également interrompre l'OB de cycle de programme. Les demandes de communication peuvent donc elles aussi constituer une menace pour la cohérence des données. La CPU s'assure que les types de données simples sont toujours lus et écrits de manière cohérente par les instructions du programme utilisateur. Comme le programme utilisateur est interrompu périodiquement par des opérations de communication, il n'est pas possible de garantir que plusieurs valeurs dans la CPU seront toutes mises à jour en même temps par l'IHM. Ainsi, les valeurs affichées sur l'écran d'une IHM donnée pourraient provenir de cycles différents de la CPU.

Les instructions point à point, les instructions PROFINET (telles que TSEND\_C et TRCV\_C), les instructions de périphérie décentralisée PROFINET et les instructions de périphérie décentralisée PROFIBUS (Page 291) transfèrent des tampons de données qui peuvent être interrompus. Garantissez la cohérence des données des tampons en évitant de lire ou d'écrire dans les tampons à la fois dans l'OB de cycle de programme et dans les OB d'alarme. S'il est nécessaire de modifier les valeurs des tampons pour ces instructions dans un OB d'alarme, utilisez une instruction DIS\_AIRT pour retarder toute alarme (OB d'alarme ou alarme de communication en provenance d'une IHM ou d'une autre CPU) jusqu'à l'exécution d'une instruction EN\_AIRT.

---

#### Remarque

L'utilisation de l'instruction DIS\_AIRT retarde le traitement des OB d'alarme jusqu'à exécution de l'instruction EN\_AIRT, affectant ainsi le temps d'attente d'alarme de votre programme utilisateur (durée s'écoulant entre l'apparition d'un événement et l'exécution de l'OB d'alarme correspondant).

---

## 6.5 Langage de programmation

STEP 7 fournit les langages de programmation standard suivants pour le S7-1200 :

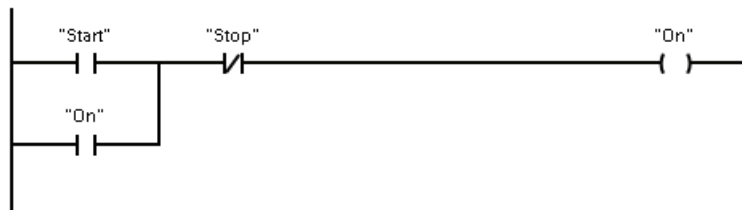
- CONT (schéma à contacts) est un langage de programmation graphique. Sa représentation se base sur des schémas (Page 165) de circuit.
- LOG (logigramme) est un langage de programmation se fondant sur les symboles logiques graphiques utilisés en algèbre (Page 166) booléenne.
- SCL (Structured Control Language) est un langage de programmation littéral évolué (Page 166).

Lorsque vous créez un bloc de code, vous sélectionnez le langage de programmation à utiliser par ce bloc.

Votre programme utilisateur peut utiliser des blocs de code créés dans n'importe lequel des langages de programmation disponibles.

### 6.5.1 Schéma à contacts (CONT)

Les éléments d'un schéma de circuit, contacts à ouverture et à fermeture et bobines par exemple, sont reliés pour former des réseaux.



Pour créer la logique pour des opérations complexes, vous pouvez insérer des branches formant des circuits parallèles. Les branches parallèles s'ouvrent vers le bas ou se connectent directement à la barre conductrice. Vous terminez les branches vers le haut.

CONT fournit des instructions sous forme de boîtes pour des fonctions variées, telles que les fonctions mathématiques, de temporisation, de comptage et de transfert.

STEP 7 ne limite pas le nombre d'instructions (lignes et colonnes) dans un réseau CONT.

---

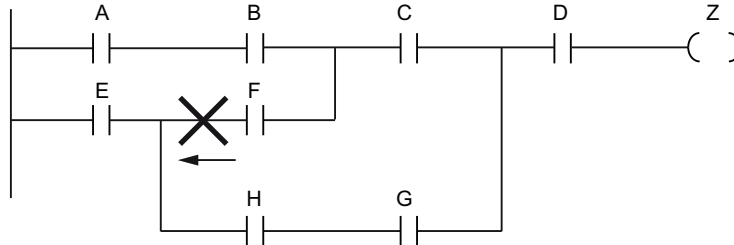
#### Remarque

Chaque réseau CONT doit se terminer par une bobine ou une boîte d'instruction.

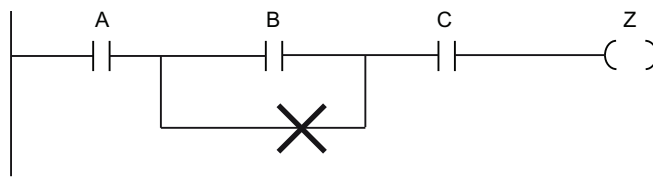
---

Vous devez tenir compte des règles suivantes lors de la création d'un réseau CONT :

- Vous ne pouvez pas créer de branche qui entraînerait un flux de courant en sens inverse.

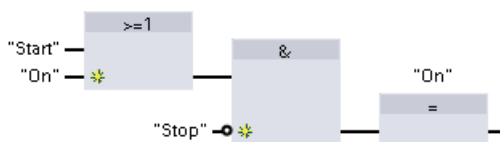


- Vous ne pouvez pas créer de branche qui provoquerait un court-circuit.



### 6.5.2 Logigramme (LOG)

Comme CONT, LOG est un langage de programmation graphique. La représentation de la logique repose sur les symboles logiques graphiques utilisés en algèbre booléenne.



Pour créer la logique pour des opérations complexes, insérez des branches parallèles entre les boîtes.

Les fonctions mathématiques et autres fonctions complexes peuvent être représentées directement avec des boîtes logiques.

STEP 7 ne limite pas le nombre d'instructions (lignes et colonnes) dans un réseau LOG.

### 6.5.3 SCL

SCL (Structured Control Language) est un langage de programmation évolué basé sur PASCAL pour les CPU SIMATIC S7. SCL prend en charge la structure de blocs de STEP 7 (Page 158). Vous pouvez en outre inclure des blocs de programme écrits en SCL avec des blocs de programme écrits en CONT et LOG.

Les instructions SCL utilisent des opérateurs de programmation standard, par exemple pour l'affectation (:=) et les fonctions mathématiques (+ pour l'addition, - pour la soustraction, \* pour la multiplication et / pour la division). SCL utilise également des opérations de gestion de programme PASCAL standard, telles que IF-THEN-ELSE, CASE, REPEAT-UNTIL, GOTO et RETURN. Vous pouvez utiliser n'importe quelle référence PASCAL pour les éléments syntaxiques du langage de programmation SCL. Un grand nombre des autres instructions pour SCL, comme les temporisations et les compteurs, correspondent aux instructions CONT et LOG. Pour plus d'informations sur des instructions spécifiques, reportez-vous aux descriptions correspondantes dans les chapitres sur les instructions de base (Page 185) et les instructions avancées (Page 263).

Vous pouvez spécifier, lors de la création de n'importe quel type de bloc (OB, FB ou FC), que ce bloc utilise le langage de programmation SCL. STEP 7 fournit un éditeur de programmes SCL qui inclut les éléments suivants :

- Section d'interface pour définir les paramètres du bloc de code
- Section de code pour le code de programme
- Arborescence d'instructions contenant les instructions SCL prises en charge par la CPU

Vous entrez le code SCL pour votre instruction directement dans la section de code. Pour des instructions plus complexes, faites simplement glisser les instructions SCL depuis l'arborescence d'instructions et déposez-les dans votre programme. Vous pouvez également utiliser un éditeur de texte quelconque pour créer un programme SCL et importer ensuite ce fichier dans STEP 7.

Interface			
	Nom	Type de données	Commentaire
1	Input		
2	StartStopSwitch	Bool	
3	Output		
4	RunYesNo	Bool	
5	InOut		
6	<ajouter nouvelle>		
7	Temp		
8	<ajouter nouvelle>		
9	Return		
10	Ret_Val	Void	

IF...	CASE... OF...	FOR... TO... DO...	WHILE... DO...
1	IF	condition	THEN
2		// Statement	section IF
3		;	
4	END_IF;		

Vous pouvez déclarer les types de paramètres suivants dans la section d'interface du bloc de code SCL :

- Input, Output, InOut et Ret\_Val : Ces paramètres définissent les variables d'entrée, les variables de sortie et la valeur en retour pour le bloc de code. Le nom de variable que vous entrez ici est utilisé localement lors de l'exécution du bloc de code. En règle générale, vous n'utilisez pas le nom de variable globale de la table de variables.
- Static (FB uniquement ; la figure ci-dessus concerne une FC) : Les variables statiques sont utilisées pour stocker des résultats intermédiaires statiques dans le bloc de données d'instance. Les données statiques sont conservées jusqu'à leur écrasement, qui peut avoir lieu après plusieurs cycles. Les noms des blocs qui sont appelés dans ce bloc de code comme multi-instance sont également stockés dans les données locales statiques.
- Temp : Ces paramètres sont les variables temporaires utilisées lors de l'exécution du bloc de code.

Si vous appelez le bloc de code SCL depuis un autre bloc de code, les paramètres du bloc de code SCL apparaissent comme entrées ou sorties.



Dans cet exemple, les variables pour "Start" et "On" (depuis la table de variables du projet) correspondent à "StartStopSwitch" et "RunYesNo" dans la table de déclaration du programme SCL.

### Construction d'une expression SCL

Une expression SCL est une formule permettant de calculer une valeur. Cette expression est composée d'opérandes et d'opérateurs (tels que \*, /, + ou -). Les opérandes peuvent être des variables, des constantes ou des expressions.

L'évaluation de l'expression se fait dans un certain ordre qui est défini par les facteurs suivants :

- Chaque opérateur a une priorité prédéfinie, l'opération de priorité la plus haute étant exécutée en premier.
- Si les opérateurs sont d'égale priorité, ils sont traités de gauche à droite.
- Vous utilisez des parenthèses pour désigner une série d'opérateurs devant être évalués ensemble.

Le résultat d'une expression peut être utilisé pour affecter une valeur à une variable utilisée par votre programme, comme condition devant être utilisée par une instruction de contrôle ou en tant que paramètres pour une autre instruction SCL ou pour appeler un bloc de code.

Tableau 6- 2 Opérateurs dans SCL

Type	Opération	Opérateur	Priorité
Parenthèses	( <i>expression</i> )	( , )	1
Mathématique	Puissance	**	2



Type	Opération	Opérateur	Priorité
	Signe (plus unaire)	+	3
	Signe (moins unaire)	-	3
	Multiplication	*	4
	Division	/	4
	Modulo	MOD	4
	Addition	+	5
	Soustraction	-	5
Comparaison	Inférieur à	<	6
	Inférieur ou égal à	<=	6
	Supérieur à	>	6
	Supérieur ou égal à	>=	6
	Egal à	=	7
	Différent de	<>	7
Logique sur bits	Négation (unaire)	NOT	3
	ET logique	AND ou &	8
	OU exclusif logique	XOR	9
	OU logique	OR	10
Affectation	Affectation	:=	11

En tant que langage de programmation évolué, SCL utilise des instructions standard pour les tâches de base :

- Instruction d'affectation : :=
- Fonctions mathématiques : +, -, \* et /
- Adressage de variables globales : "<nom variable>" (nom de la variable ou du bloc de données entre guillemets)
- Adressage de variables locales : #<nom variable> (nom de la variable précédé du symbole "#")

Les exemples suivants montrent différentes expressions pour divers usages :

<pre>"C" := #A+#B;</pre>	<p>Affecte la somme de deux variables locales à une variable globale.</p>
<pre>"Data_block_1".Tag := #A;</pre>	<p>Affectation à une variable de bloc de données</p>
<pre>IF #A &gt; #B THEN "C" := #A;</pre>	<p>Condition pour l'instruction IF-THEN</p>
<pre>"C" := SQRT (SQR (#A) + SQR (#B));</pre>	<p>Paramètres pour l'instruction SQRT</p>

Les opérateurs arithmétiques peuvent traiter divers types de données numériques. Le type de données du résultat est déterminé par le type de données des opérandes les plus significatifs. Par exemple, une multiplication qui utilise un opérande INT et un opérande REAL donne une valeur REAL en résultat.

### Instructions de contrôle

Une instruction de contrôle est un type spécifique d'expression SCL qui exécute les tâches suivantes :

- Aiguillage dans un programme
- Sections de répétition du code de programme SCL
- Saut à d'autres parties du programme SCL
- Exécution conditionnelle

Parmi les instructions de contrôle SCL, on trouve IF-THEN, CASE-OF, FOR-TO-DO, WHILE-DO, REPEAT-UNTIL, CONTINUE, GOTO et RETURN.

Une seule instruction occupe typiquement une ligne de code. Vous pouvez entrer plusieurs instructions sur une ligne ou vous pouvez subdiviser une instruction en plusieurs lignes de code pour la rendre plus facile à lire. Les séparateurs (tels que tabulations, sauts de ligne et espaces supplémentaires) ne sont pas pris en compte lors du contrôle de syntaxe. Une instruction END met fin à l'instruction de contrôle.

L'exemple suivant montre une instruction de contrôle FOR-TO-DO (les deux formes de codage sont syntaxiquement correctes).

```
FOR x := 0 TO max DO sum := sum + value(x) ; END_FOR;  
FOR x := 0 TO max DO  
    sum := sum + value(x) ;  
END_FOR;
```

Une instruction de contrôle peut également être munie d'un repère. Un repère est délimité par un deux-points au début de l'instruction :

**Repère** : <Instruction>;

L'aide en ligne de STEP 7 fournit des informations de référence complètes sur le langage de programmation SCL.

### Conditions

Une condition est une expression de comparaison ou une expression logique dont le résultat est de type BOOL (avec la valeur TRUE ou FALSE). L'exemple suivant montre différents types de conditions.

#Température > 50	Expression de relation
#Compteur <= 100	
#CHAR1 < 'S'	
(#Alpha <> 12) AND NOT #Beta	Expression de comparaison et expression logique
5 + #Alpha	Expression arithmétique

Une condition peut utiliser des expressions arithmétiques :

- La condition de l'expression est TRUE si le résultat est une valeur quelconque différente de zéro.
- La condition de l'expression est FALSE si le résultat est égal à zéro.

## Adressage

Comme CONT et LOG, SCL permet l'utilisation de variables (adressage symbolique) ou d'adresses absolues dans le programme utilisateur. SCL permet également d'utiliser une variable comme indice de tableau.

### Adressage absolu

```
I0.0  
MB100
```

### Adressage symbolique

"PLC_Tag_1"	Variable dans la table de variables API
"Data_block_1".Tag_1	Variable dans un bloc de données
"Data_block_1".MyArray[#i]	Élément de tableau dans un tableau bloc de données

## Adressage indexé avec les instructions PEEK et POKE

SCL fournit les instructions PEEK et POKE qui vous permettent de lire et d'écrire dans les blocs de données, les E/S ou la mémoire. Vous indiquez des paramètres pour les décalages d'octet ou de bit spécifiques pour l'opération.

---

### Remarque

Pour utiliser les instructions PEEK et POKE avec des blocs de données, vous devez utiliser des blocs de données standard (non optimisés). Notez également que les instructions PEEK et POKE ne font que transférer des données, elles n'ont pas connaissance des types de données aux diverses adresses.

---

```
PEEK(area:=_in_,  
      dbNumber:=_in_,  
      byteOffset:=_in_);
```

Lit l'octet désigné par byteOffset du bloc de données ou de la zone d'E/S ou de mémoire référencée.

Exemple avec un bloc de données :

```
%MB100 := PEEK(area:=16#84,  
              dbNumber:=1, byteOffset:=#i);
```

Exemple avec l'octet d'entrée IB3 :

```
%MB100 := PEEK(area:=16#81,  
              dbNumber:=0, byteOffset:=#i); // when  
#i = 3
```

```
PEEK_WORD(area:=_in_,  
          dbNumber:=_in_,  
          byteOffset:=_in_);
```

Lit le mot désigné par byteOffset du bloc de données ou de la zone d'E/S ou de mémoire référencée.

Exemple :

```
%MW200 := PEEK_WORD(area:=16#84,  
                   dbNumber:=1, byteOffset:=#i);
```

```
PEEK_DWORD(area:=_in_,
            dbNumber:=_in_,
            byteOffset:=_in_);
```

```
PEEK_BOOL(area:=_in_,
           dbNumber:=_in_,
           byteOffset:=_in_,
           bitOffset:=_in_);
```

```
POKE(area:=_in_,
      dbNumber:=_in_,
      byteOffset:=_in_,
      value:=_in_);
```

```
POKE_BOOL(area:=_in_,
           dbNumber:=_in_,
           byteOffset:=_in_,
           bitOffset:=_in_,
           value:=_in_);
```

```
POKE_BLK(area_src:=_in_,
          dbNumber_src:=_in_,
          byteOffset_src:=_in_,
          area_dest:=_in_,
          dbNumber_dest:=_in_,
          byteOffset_dest:=_in_,
          count:=_in_);
```

Lit le double mot désigné par byteOffset du bloc de données ou de la zone d'E/S ou de mémoire référencée.

Exemple :

```
%MD300 := PEEK_DWORD(area:=16#84,
                    dbNumber:=1, byteOffset:=#i);
```

Lit la valeur booléenne désignée par bitOffset et byteOffset du bloc de données ou de la zone d'E/S ou de mémoire référencée.

Exemple :

```
%MB100.0 := PEEK_BOOL(area:=16#84,
                     dbNumber:=1, byteOffset:=#ii,
                     bitOffset:=#j);
```

Ecrit la valeur (octet, mot ou double mot) au décalage byteOffset désigné du bloc de données ou de la zone d'E/S ou de mémoire référencée.

Exemple avec un bloc de données :

```
POKE(area:=16#84, dbNumber:=2,
      byteOffset:=3, value:="Tag_1");
```

Exemple avec l'octet de sortie QB3 :

```
POKE(area:=16#82, dbNumber:=0,
      byteOffset:=3, value:="Tag_1");
```

Ecrit la valeur booléenne aux décalages bitOffset et byteOffset désignés du bloc de données ou de la zone d'E/S ou de mémoire référencée.

Exemple :

```
POKE_BOOL(area:=16#84, dbNumber:=2,
           byteOffset:=3, bitOffset:=5,
           value:=0);
```

Ecrit un nombre ("count") d'octets en commençant au décalage byteOffset désigné du bloc de données ou de la zone d'E/S ou de mémoire source référencée au décalage byteOffset désigné du bloc de données ou de la zone d'E/S ou de mémoire de destination référencée.

Exemple :

```
POKE_BLK(area_src:=16#84,
          dbNumber_src:=#src_db,
          byteOffset_src:=#src_byte,
          area_dest:=16#84,
          dbNumber_dest:=#src_db,
          byteOffset_dest:=#src_byte,
          count:=10);
```

Pour les instructions PEEK et POKE, les valeurs suivantes sont possibles pour les paramètres "area", "area\_src" et "area\_dest". Pour les zones autres que les blocs de données, le paramètre dbNumber doit être égal à 0.

16#81	I
16#82	Q
16#83	M
16#84	DB

### Appel d'autres blocs de code depuis votre programme SCL

Pour appeler un autre bloc de code dans votre programme utilisateur, il vous suffit d'entrer le nom (ou l'adresse absolue) du FB ou de la FC concernée avec ses paramètres. Pour un FB, vous devez indiquer le DB d'instance à appeler avec le FB.

`<nom DB> (liste de paramètres)` Appel comme instance unique  
`<#nom instance> (liste de paramètres)` Appel comme multi-instance

```
"MyDB" (MyInput:=10, MyInOut:="Tag1");
```

`<nom FC> (liste de paramètres)` Appel standard  
`<opérande>:=<nom FC> (liste de paramètres)` Appel dans une expression

```
"MyFC" (MyInput:=10, MyInOut:="Tag1");
```

Vous pouvez aussi faire glisser des blocs du navigateur du projet dans l'éditeur de programme SCL, puis achever l'affectation des paramètres.

## 6.5.4 EN et ENO pour CONT, LOG et SCL

### Détermination du "flux de courant" (EN et ENO) pour une instruction

Certaines instructions (telles que les instructions mathématiques et de transfert) comportent des paramètres EN et ENO. Ces paramètres font référence au flux de courant en CONT et LOG et déterminent si l'instruction est ou non exécutée pendant ce cycle. SCL vous permet également d'activer le paramètre ENO pour un bloc de code.

- EN (pour Enable In) est une entrée booléenne. Un flux de courant (EN = 1) doit être présent au niveau de cette entrée pour que la boîte d'instruction soit exécutée. Si l'entrée EN d'une boîte CONT est connectée directement à la barre conductrice gauche, l'instruction sera toujours exécutée.
- ENO (pour Enable Out) est une sortie booléenne. Si un flux de courant est présent à l'entrée EN d'une boîte et que la fonction correspondante s'exécute sans erreur, la sortie ENO transmet le flux de courant (ENO = 1) à l'élément suivant. Si une erreur est détectée lors de l'exécution de la boîte d'instruction, le flux de courant s'arrête (ENO = 0) à la boîte qui a généré l'erreur.

Tableau 6- 3 Opérandes pour EN et ENO

Editeur de programmes	Entrées/sorties	Opérandes	Type de données
CONT	EN, ENO	Flux de courant	Bool
LOG	EN	I, I:P, Q, M, DB, Temp, flux de courant	Bool
	ENO	Flux de courant	Bool
SCL	EN <sup>1</sup>	TRUE, FALSE	Bool
	ENO <sup>2</sup>	TRUE, FALSE	Bool

<sup>1</sup> L'utilisation de EN est disponible uniquement pour les FB.

<sup>2</sup> L'utilisation de ENO avec le bloc de code SCL est facultative. Vous devez configurer le compilateur SCL afin qu'il active ENO à l'achèvement du bloc de code.

### Configuration de SCL pour l'activation de ENO

Procédez comme suit pour configurer le compilateur SCL pour l'activation de ENO :

1. Sélectionnez la commande "Paramètres" dans le menu "Outils".
2. Affichez le détail des propriétés "Programmation API" et sélectionnez "SCL (Structured Control Language)".
3. Sélectionnez l'option "Activer ENO automatiquement".

## Effet des paramètres Ret\_Val et Status sur ENO

Certaines instructions, telles que les instructions de communication ou les instructions de conversion de chaîne, fournissent un paramètre de sortie qui contient des informations sur le traitement de l'instruction. Par exemple, certaines instructions fournissent un paramètre Ret\_Val (valeur en retour) qui est typiquement un type de données Int contenant des informations d'état dans une plage allant de -32768 à +32767. D'autres instructions fournissent un paramètre Status qui est typiquement un type de données Word contenant des informations d'état dans une plage allant des valeurs hexadécimales 16#0000 à 16#FFFF. La valeur numérique contenue dans un paramètre Ret\_Val ou Status détermine l'état de ENO pour l'instruction correspondante.

- Ret\_Val : Une valeur comprise entre 0 et 32767 met typiquement ENO à 1 (ou VRAI). Une valeur comprise entre -32768 et -1 met typiquement ENO à 0 (ou FAUX). Pour évaluer Ret\_Val, passez en représentation hexadécimale.
- Status : Une valeur comprise entre 16#0000 et 16#7FFF met typiquement ENO à 1 (ou VRAI). Une valeur comprise entre 16#8000 et 16#FFFF met typiquement ENO à 0 (ou FAUX).

Les instructions s'exécutant sur plus d'un cycle comportent souvent un paramètre Busy (Bool) qui signale que l'instruction est active mais que l'exécution n'est pas encore achevée. Souvent, ces instructions fournissent aussi un paramètre Done (Bool) et un paramètre Error (Bool). Done signale que l'instruction s'est achevée sans erreur et Error que l'instruction s'est achevée avec une erreur.

- Lorsque Busy vaut 1 (ou VRAI), ENO est égal à 1 (ou VRAI).
- Lorsque Done vaut 1 (ou VRAI), ENO est égal à 1 (ou VRAI).
- Lorsque Error vaut 1 (ou VRAI), ENO est égal à 0 (ou FAUX).

### Voir aussi

Instructions Contrôler validité et Contrôler invalidité (Page 210)

## 6.6 Protection

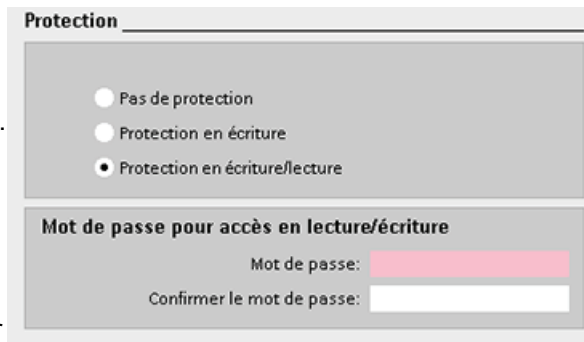
### 6.6.1 Protection d'accès pour la CPU

La CPU fournit trois niveaux de sécurité permettant de limiter l'accès à des fonctions spécifiques. Lorsque vous configurez le niveau de sécurité et le mot de passe pour une CPU, vous limitez les fonctions et les zones de mémoire qui sont accessibles sans saisie d'un mot de passe.

Il y a distinction entre majuscules et minuscules dans le mot de passe.

Procédez comme suit pour configurer le mot de passe :

1. Sélectionnez la CPU dans la "Configuration des appareils".
2. Dans la fenêtre d'inspection, sélectionnez l'onglet "Propriétés".
3. Sélectionnez la propriété "Protection" pour choisir le niveau de protection et entrer un mot de passe.



Chaque niveau permet d'accéder à certaines fonctions sans mot de passe. Par défaut, il n'y a pas de restriction d'accès ni de protection par mot de passe pour la CPU. Pour limiter l'accès à une CPU, vous configurez les propriétés de la CPU et vous entrez le mot de passe.

L'entrée du mot de passe via un réseau ne met pas en cause la protection par mot de passe de la CPU. La protection par mot de passe ne concerne pas l'exécution des instructions du programme utilisateur, fonctions de communication comprises. La saisie du mot de passe correct permet d'accéder à toutes les fonctions.

Les communications API-API (par le biais d'instructions de communication dans les blocs de code) ne sont pas réduites par le niveau de sécurité dans la CPU. La fonctionnalité IHM n'est pas non plus restreinte.

Tableau 6- 4 Niveaux de sécurité pour la CPU

Niveau de sécurité	Restrictions d'accès
Pas de protection	Autorise un accès intégral sans protection par mot de passe.
Protection en écriture	Permet l'accès IHM et toutes les formes de communication API-API sans protection par mot de passe. Un mot de passe est nécessaire pour modifier la CPU (écriture) et pour changer le mode de la CPU (MARCHE/ARRET).
Protection en lecture/écriture	Permet l'accès IHM et toutes les formes de communication API-API sans protection par mot de passe. Un mot de passe est nécessaire pour lire les données dans la CPU, modifier la CPU (écriture) et changer le mode de la CPU (MARCHE/ARRET).

### 6.6.2 Protection du savoir-faire

Une protection "know-how" vous permet d'empêcher l'accès non autorisé à un ou plusieurs blocs de code (OB, FB, FC ou DB) dans votre programme. Vous créez un mot de passe pour limiter l'accès au bloc de code. La protection par mot de passe empêche de lire ou de modifier le bloc de code sans autorisation. Sans mot de passe, vous pouvez lire uniquement les informations suivantes sur le bloc de code :

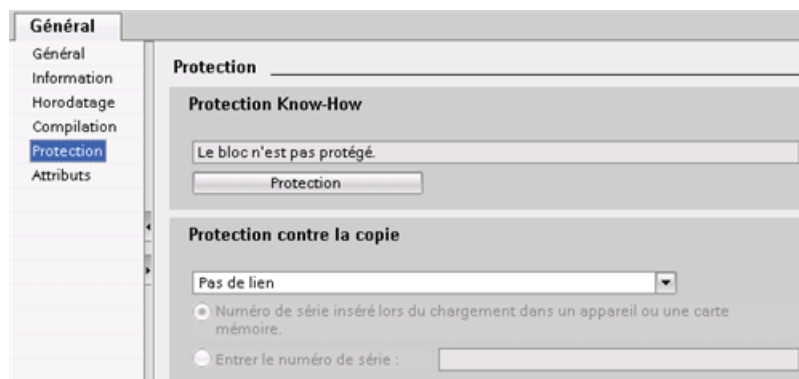
- Titre, commentaire et propriétés du bloc
- Paramètres transmis (IN, OUT, IN\_OUT, renvoi)



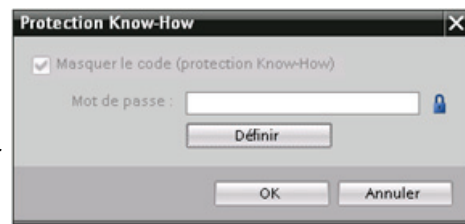
- Structure d'appel du programme
- Variables globales dans les références croisées (sans information sur l'endroit où elles sont utilisées) ; les variables locales sont masquées

Lorsque vous activez la protection "know-how" pour un bloc, le code à l'intérieur de ce bloc n'est accessible qu'après saisie du mot de passe.

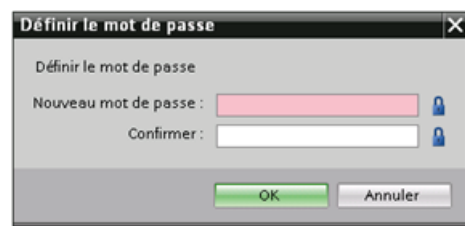
Utilisez la Task Card "Propriétés" du bloc de code pour configurer la protection "know-how" pour ce bloc. Après avoir ouvert le bloc de code, sélectionnez "Protection" dans les propriétés.



1. Dans les propriétés du bloc de code, cliquez sur le bouton "Protection" pour afficher la boîte de dialogue "Protection Know-How".
2. Cliquez sur le bouton "Définir" pour entrer le mot de passe.



Après avoir entré et confirmé le mot de passe, cliquez sur "OK".

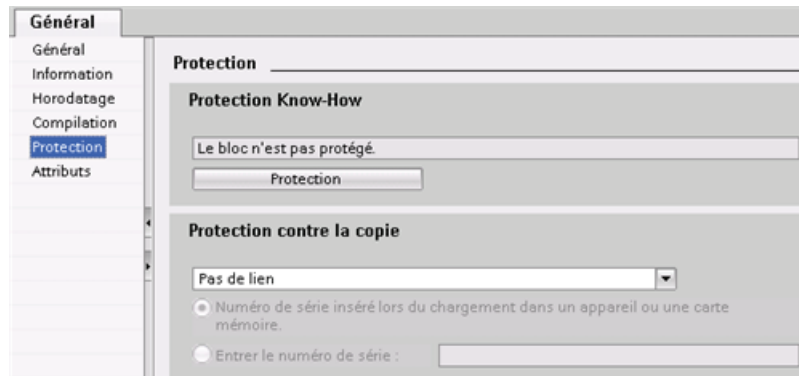


### 6.6.3 Protection contre la copie

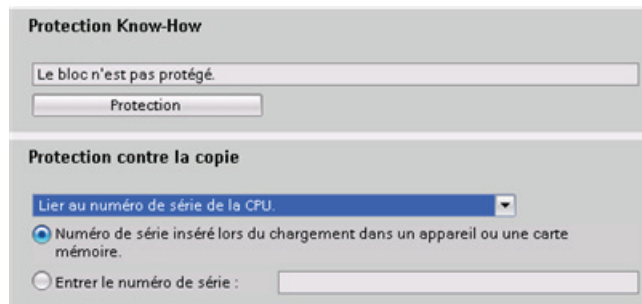
Une fonction de sécurité supplémentaire vous permet de lier le programme ou les blocs de code à une utilisation avec une carte mémoire ou une CPU spécifique. Cette fonction est particulièrement utile pour protéger votre propriété intellectuelle. Lorsque vous liez un programme ou un bloc à un appareil spécifique, vous restreignez le programme ou le bloc de code pour utilisation uniquement avec une carte mémoire ou une CPU spécifique. Cette fonction vous permet de distribuer un programme ou un bloc de code électroniquement (via Internet ou par email) ou en envoyant une cartouche mémoire.

Utilisez la Task Card "Propriétés" du bloc de code pour lier le bloc à une CPU ou une carte mémoire spécifique.

1. Après avoir ouvert le bloc de code, sélectionnez "Protection".



2. Dans la liste déroulante sous la tâche "Protection contre la copie", sélectionnez l'option de liaison du bloc de code à une carte mémoire ou à une CPU spécifique.



3. Sélectionnez le type de protection contre la copie et saisissez le numéro de série de la carte mémoire ou de la CPU.

---

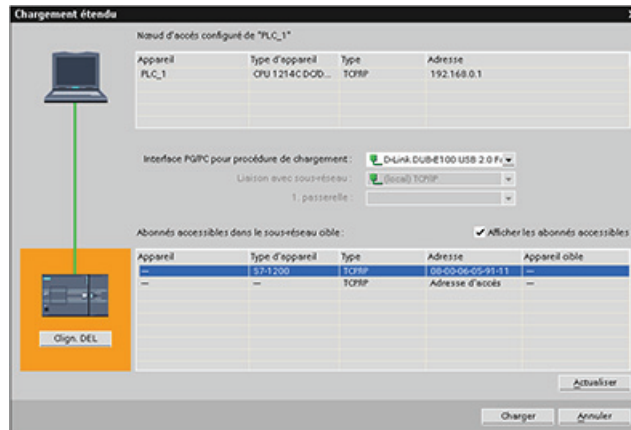
### Remarque

Il y a distinction entre majuscules et minuscules dans le numéro de série.

---

## 6.7 Chargement d'éléments de votre programme dans la CPU

Vous pouvez charger les éléments de votre projet de la console de programmation dans la CPU. Lors du chargement d'un projet, la CPU sauvegarde le programme utilisateur (OB, FC, FB et DB) en mémoire permanente.



Vous pouvez charger votre projet de la console de programmation dans votre CPU à partir de l'un des endroits suivants :

- "Arborescence du projet" : Cliquez avec le bouton droit de la souris sur l'élément de programme, puis cliquez sur la commande "Charger" du menu contextuel.
- Menu "En ligne" : Cliquez sur la commande "Charger dans l'appareil".
- Barre d'outils : Cliquez sur l'icône "Charger dans l'appareil".

## 6.8 Chargement à partir de la CPU

### 6.8.1 Copie des éléments du projet

Vous pouvez également copier les blocs de programme depuis une CPU en ligne ou une carte mémoire reliée à votre console de programmation.

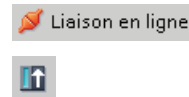
Préparez le projet hors ligne pour les blocs de programme copiés :

1. Ajoutez un appareil CPU qui correspond à la CPU en ligne.
2. Cliquez une fois sur le noeud de la CPU pour rendre le dossier "Blocs de programme" visible.



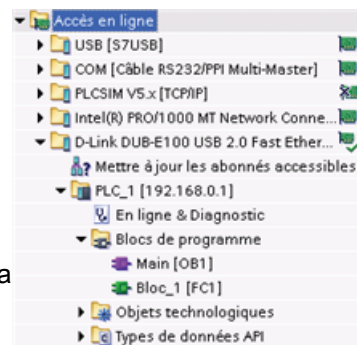
Procédez comme suit pour charger les blocs de programme de la CPU en ligne dans le projet hors ligne :

1. Cliquez sur le dossier "Blocs de programme" dans le projet hors ligne.
2. Cliquez sur le bouton "Liaison en ligne".
3. Cliquez sur le bouton "Charger depuis la CPU".
4. Confirmez votre décision dans la boîte de dialogue Charger depuis la CPU (Page 723).

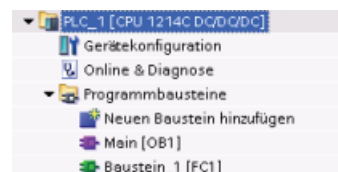


Vous pouvez également procéder comme suit :

1. Dans le navigateur du projet, affichez le détail du noeud pour "Accès en ligne" afin de sélectionner les blocs de programme dans la CPU en ligne :
2. Affichez le détail du noeud pour le réseau et double-cliquez sur "Mettre à jour les abonnés accessibles".
3. Affichez le détail du noeud pour la CPU.
4. Faites glisser le fichier "Blocs de programme" depuis la CPU en ligne et déplacez-le dans le fichier "Blocs de programme" de votre projet en ligne.
5. Dans la boîte de dialogue "Téléchargement prévisualisation", sélectionnez "Continuer" et cliquez ensuite sur le bouton "Téléchargement depuis l'appareil".



Lorsque le téléchargement est terminé, tous les blocs de programme, blocs technologiques et variables sont affichés dans la zone hors ligne.



### Remarque

Vous pouvez copier les blocs de programme depuis la CPU en ligne vers un programme existant. Le fichier "Blocs de programme" du projet hors ligne ne doit pas être vide. Cependant, le programme existant sera effacé et remplacé par le programme utilisateur de la CPU en ligne.

## 6.8.2 Utilisation de la fonction de comparaison

Vous pouvez utiliser l'éditeur de comparaison (Page 729) dans STEP 7 pour rechercher les différences entre les projets en ligne et hors ligne. Cela peut être utile avant d'effectuer un chargement depuis la CPU.

## 6.9 Débogage et test du programme

### 6.9.1 Visualisation et forçage de données dans la CPU

Comme indiqué dans le tableau suivant, vous pouvez visualiser et forcer des valeurs dans la CPU en ligne.

Tableau 6- 5 Visualisation et forçage de données avec STEP 7

Editeur	Visualisation	Forçage	Forçage permanent
Table de visualisation	Oui	Oui	Non
Table de forçage permanent	Oui	Non	Oui
Editeur de programmes	Oui	Oui	Non
Table des variables	Oui	Non	Non
Editeur de DB	Oui	Non	Non

Nom	Adresse	Format d'affichage	Valeur visualisati...	Valeur de forçage
*On*	%I0.0	Bool	<input type="checkbox"/> FALSE	
*Off*	%I0.1	Bool	<input type="checkbox"/> FALSE	
*Run*	%Q0.0	Bool	<input type="checkbox"/> FALSE	

Visualisation avec une table de visualisation



Visualisation avec l'éditeur CONT

Reportez-vous au chapitre "Outils en ligne et de diagnostic" pour plus d'informations sur la visualisation et le forçage de données dans la CPU (Page 730).

## 6.9.2 Tables de visualisation et tables de forçage permanent

Vous pouvez utiliser des "tables de visualisation" pour visualiser et modifier les valeurs d'un programme utilisateur exécuté par la CPU en ligne. Vous pouvez créer et sauvegarder différentes tables de visualisation dans votre projet afin de prendre en charge des environnements de test variés. Vous pouvez ainsi reproduire des tests pendant la mise en service ou à des fins de dépannage et de maintenance.

Avec une table de visualisation, vous pouvez surveiller la CPU et interagir avec elle alors qu'elle exécute le programme utilisateur. Vous pouvez afficher ou changer les valeurs non seulement pour les variables des blocs de code et de données, mais également pour les zones de mémoire de la CPU, à savoir les entrées et sorties (I et Q), la périphérie d'entrée (I:P), les mémentos (M) et les blocs de données (DB).

Avec la table de visualisation, vous pouvez débloquer les sorties physiques (Q:P) d'une CPU à l'état ARRET. Vous pouvez, par exemple, affecter des valeurs spécifiques aux sorties lorsque vous testez le câblage de la CPU.

STEP 7 fournit également une table de forçage permanent pour le forçage permanent d'une variable à une valeur spécifique. Pour plus d'informations sur le forçage, reportez-vous au paragraphe traitant du forçage de valeurs dans la CPU (Page 737) au chapitre "Outils en ligne et de diagnostic".

---

### Remarque

Les valeurs de forçage permanent sont stockées dans la CPU et non dans la table de visualisation.

Vous ne pouvez pas forcer une entrée (adresse I), mais vous pouvez forcer une entrée de périphérie. Pour forcer une entrée de périphérie, ajoutez ":P" à l'adresse (par exemple, "On:P").

---

## 6.9.3 Affichage de l'usage des références croisées

La fenêtre d'inspection affiche des informations sous forme de références croisées indiquant la manière dont un objet sélectionné est utilisé dans l'ensemble du projet, comme p. ex. le programme utilisateur, la CPU ou tout appareil IHM. La table des "Références croisées" affiche les instances où un objet sélectionné et utilisé, de même que les autres objets qui l'utilisent. La fenêtre d'inspection contient également les blocs qui sont exclusivement accessibles en ligne dans les références croisées. Pour afficher les références croisées, choisissez la commande "Afficher références croisées" (dans la vue du projet, les références croisées font partie du menu "Outils").

---

### Remarque

Il n'est pas nécessaire de fermer l'éditeur pour voir les informations sur les références croisées.

---

Vous avez la possibilité de trier les entrées des références croisées. La liste des références croisées fournit une vue d'ensemble de l'utilisation des adresses mémoire et des variables dans le programme utilisateur.

- Lorsque vous créez et modifiez un programme, vous obtenez une vue d'ensemble des opérandes, variables et appels de blocs utilisés.
- Vous pouvez, à partir des références croisées, sauter directement à l'occurrence d'utilisation des opérandes et variables.
- Durant un test du programme ou lors d'un dépannage, vous voyez quelle adresse de mémoire est traitée par quelle commande et dans quel bloc, quelle variable est utilisée dans quelle vue et quel bloc est appelé par quel autre bloc.

Tableau 6- 6 Éléments de la référence croisée

Colonne	Description
Objet	Nom de l'objet utilisant les objets de niveau inférieur ou qui est utilisé par les objets de niveau inférieur
Quantité	Nombre d'utilisations
Adresse	Toute adresse possible, p. ex. adresse de réseau.
Propriétés	Propriétés spécifiques des objets référencés, p. ex. noms de variables dans des déclarations multiinstances
En tant que	Affiche des informations supplémentaires sur l'objet, p. ex. si un DB d'instance est utilisé en tant que modèle ou instance multiple.
Accès	Type d'accès à l'opérande, à savoir en lecture (R) et/ou en écriture (W).
Adresse	Adresse de l'opérande
Type	Information sur le type et le langage utilisés pour créer l'objet.
Chemin	Chemin de l'objet dans l'arborescence

#### 6.9.4 Structure d'appel permettant de constater la hiérarchie d'appel

La structure d'appel décrit la hiérarchie d'appel des blocs dans votre programme utilisateur. Elle fournit une vue d'ensemble des blocs utilisés, des appels d'autres blocs, des relations entre blocs, des données requises pour chaque bloc et de l'état des blocs. Vous pouvez ouvrir l'éditeur de programmes et éditer les blocs de la structure d'appel.

L'affichage de la structure d'appel fournit une liste des blocs utilisés dans le programme utilisateur. STEP 7 met en évidence le premier niveau de la structure d'appel et affiche tout bloc n'étant pas appelé par un autre bloc dans le programme. Le premier niveau de la structure d'appel affiche les OB ainsi que tous les FC, FB et DB qui ne sont pas appelés par un OB. Si un bloc de code appelle un autre bloc, le bloc appelé est représenté en retrait sous le bloc appelant. La structure d'appel affiche uniquement les blocs qui sont appelés par un blocs de code.

Vous avez la possibilité d'afficher exclusivement les blocs entraînant des conflits dans la structure d'appel. Les conditions suivantes entraînent des conflits :

- Blocs exécutant des appels avec des horodatages plus anciens ou plus récents
- Blocs appelant un bloc avec une interface modifiée

- Blocs utilisant une variable avec une adresse et/ou un type de données modifiés
- Blocs n'étant pas appelés directement ou indirectement par un OB.
- Blocs appelant un bloc inexistant ou manquant

Vous pouvez regrouper plusieurs appels de bloc et blocs de données. Une liste déroulante indique les liens aux différentes adresses d'appel.

Vous pouvez également effectuer un contrôle de cohérence pour afficher les conflits d'horodatage. Un changement d'horodatage d'un bloc durant ou après la compilation du programme risque d'entraîner des conflits d'horodatage, qui à leur tour provoquent des incohérences parmi les blocs appelants et les blocs appelés.

- Une recompilation des blocs de code permet de corriger la plupart des conflits d'horodatage et d'interface.
- Si la compilation ne permet pas de remédier aux incohérences, servez-vous du lien dans la colonne "Détails" pour aller à la source du problème dans l'éditeur de programmes. Vous pouvez alors corriger manuellement toutes les incohérences.
- Tous les blocs marqués en rouge doivent être recompilés.



## Instructions de base

### 7.1 Instructions logiques sur bits

#### 7.1.1 Contacts et bobines

CONT et LOG sont très efficaces dans le traitement de la logique booléenne. Bien que SCL soit particulièrement efficace pour les calculs mathématiques complexes et pour les structures de commande de projet, vous pouvez aussi l'utiliser pour la logique booléenne.

#### Contacts CONT

Tableau 7- 1 Contacts à fermeture et contacts à ouverture



CONT	SCL	Description
"IN" 	<pre>IF in THEN     Statement; ELSE     Statement; END IF;</pre>	Contacts à fermeture et contacts à ouverture : Vous pouvez connecter des contacts à d'autres contacts et créer votre propre logique combinatoire. Si le bit d'entrée que vous indiquez utilise l'identificateur de mémoire I (entrée) ou Q (sortie), la valeur du bit est lue dans la mémoire image du processus. Les signaux physiques dans votre processus de commande sont câblés aux bornes I sur l'automate. La CPU lit les signaux d'entrée câblés et actualise en continu les valeurs d'état correspondantes dans la mémoire image des entrées.
"IN" 	<pre>IF NOT (in) THEN     Statement; ELSE     Statement; END_IF;</pre>	Vous pouvez demander la lecture directe d'une entrée physique en indiquant ":P" après le décalage I (par exemple, "%I3.4:P"). En cas de lecture directe, les valeurs de données binaires sont lues directement dans l'entrée physique et non dans la mémoire image. Une lecture directe n'actualise pas la mémoire image.

Tableau 7- 2 Types de données pour les paramètres

Paramètre	Type de données	Description
IN	Bool	Bit affecté

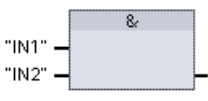
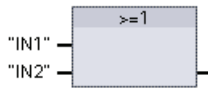
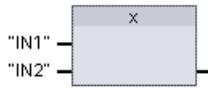
- Le contact à fermeture est fermé (activé) lorsque la valeur du bit affecté est égale à 1.
- Le contact à ouverture est fermé (activé) lorsque la valeur du bit affecté est égale à 0.
- Des contacts connectés en série créent des réseaux de logique ET.
- Des contacts connectés en parallèle créent des réseaux de logique OU.

**Boîtes ET, OU et OU EXCLUSIF en LOG**

En programmation LOG, les réseaux de contacts CONT sont transformés en réseaux de boîtes ET (&), OU (>=1) et OU EXCLUSIF (x) dans lesquels vous pouvez indiquer des valeurs binaires pour les entrées et sorties des boîtes. Vous pouvez aussi établir des connexions à d'autres boîtes logiques et créer vos propres combinaisons logiques. Une fois que vous avez placé la boîte dans votre réseau, vous pouvez faire glisser l'outil "Insérer entrée" de la barre d'outils "Favoris" ou de l'arborescence d'instructions vers le côté des entrées de la boîte afin d'ajouter des entrées. Vous pouvez aussi cliquer avec le bouton droit de la souris sur le connecteur d'entrée de la boîte et sélectionner "Insérer entrée".

Les entrées et les sorties de la boîte peuvent être connectées à une autre boîte logique ou vous pouvez entrer une adresse binaire ou un mnémonique de bit pour une entrée non connectée. Lors de l'exécution de la boîte d'instruction, les états des entrées en cours sont appliqués à la logique binaire et, si elle est vraie, la sortie de la boîte sera vraie.

Tableau 7- 3 Boîtes ET, OU et OU EXCLUSIF

LOG	SCL <sup>1</sup>	Description
	<pre>out := in1 AND in2;</pre>	Toutes les entrées d'une boîte ET doivent être vraies pour que la sortie soit vraie.
	<pre>out := in1 OR in2;</pre>	Il suffit qu'une entrée d'une boîte OU soit vraie pour que la sortie soit vraie.
	<pre>out := in1 XOR in2;</pre>	Un nombre impair d'entrées d'une boîte OU EXCLUSIF doivent être vraies pour que la sortie soit vraie.

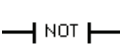
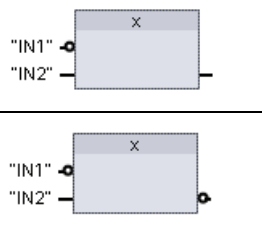
<sup>1</sup> Pour SCL : Vous devez affecter le résultat de l'opération à une variable destinée à être utilisée par une autre instruction.

Tableau 7- 4 Types de données pour les paramètres

Paramètre	Type de données	Description
IN1, IN2	Bool	Bit d'entrée

## Inverseur logique NOT

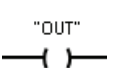
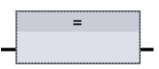
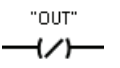
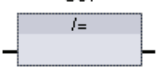
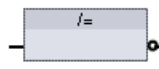
Tableau 7- 5 Inverseur logique NOT

CONT	LOG	SCL	Description
		NOT	<p>En programmation LOG, vous pouvez faire glisser l'outil "Inverser l'entrée binaire" de la barre d'outils "Favoris" ou de l'arborescence d'instructions vers une entrée ou une sortie afin de créer un inverseur logique sur ce connecteur de boîte.</p> <p>Le contact NOT en CONT inverse l'état logique de l'entrée de flux de courant.</p> <ul style="list-style-type: none"> <li>• S'il n'y a pas de flux de courant entrant dans le contact NOT, il y a un flux de courant sortant.</li> <li>• S'il y a un flux de courant entrant dans le contact NOT, il n'y a pas de flux de courant sortant.</li> </ul>

## Bobine et boîte d'affectation de sortie

L'instruction Bobine de sortie écrit une valeur pour un bit de sortie. Si le bit de sortie que vous indiquez utilise l'identificateur de mémoire Q, la CPU met le bit de sortie dans la mémoire image du processus à 1 ou à 0 conformément à l'état de flux de courant. Les signaux de sortie pour vos actionneurs de commande sont câblés aux bornes Q de la CPU. A l'état MARCHE, la CPU lit en continu vos signaux d'entrée, traite les états des entrées en fonction de la logique de votre programme, puis réagit en donnant aux sorties de nouvelles valeurs dans la mémoire image des sorties. Après chaque cycle d'exécution du programme, la CPU transfère le nouvel état des sorties mémorisé dans la mémoire image aux bornes de sortie câblées.

Tableau 7- 6 Bobine de sortie (CONT) et boîte d'affectation de sortie (LOG)

CONT	LOG	SCL	Description
		<code>out := &lt;expression booléenne&gt;;</code>	<p>En programmation LOG, les bobines CONT sont transformées en boîte d'affectation (= et /=) dans lesquelles vous indiquez une adresse de bit pour la sortie de la boîte. Les entrées et sorties de la boîte peuvent être connectées à une autre boîte logique ou vous pouvez entrer une adresse de bit.</p> <p>Vous pouvez demander l'écriture directe dans une sortie physique en indiquant ":P" après le décalage Q (par exemple, "%Q3.4:P"). En cas d'écriture directe, les valeurs de données binaires sont écrites directement dans la mémoire image des sorties et directement dans la sortie physique.</p>
		<code>out := NOT &lt;expression booléenne&gt;;</code>	
			

7.1 Instructions logiques sur bits

Tableau 7- 7 Types de données pour les paramètres

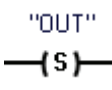
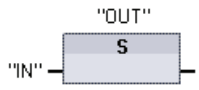
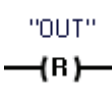
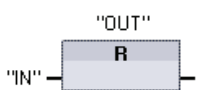
Paramètre	Type de données	Description
OUT	Bool	Bit affecté

- S'il y a flux de courant à travers une bobine de sortie ou qu'une boîte d'affectation "=" LOG est activée, le bit de sortie est mis à 1.
- S'il n'y a pas de flux de courant à travers une bobine de sortie ou qu'une boîte d'affectation "=" LOG n'est pas activée, le bit de sortie est mis à 0.
- S'il y a flux de courant à travers une bobine de sortie inversée ou qu'une boîte "/=" LOG est activée, le bit de sortie est mis à 0.
- S'il n'y a pas de flux de courant à travers une bobine de sortie inversée ou qu'une boîte "/=" LOG n'est pas activée, le bit de sortie est mis à 1.

7.1.2 Instructions Mise à 1 et Mise à 0

Mise à 1 et Mise à 0 d'1 bit

Tableau 7- 8 Instructions S et R

CONT	LOG	SCL	Description
		Non disponible	Lorsque S (Mise à 1) est activé, la valeur de données à l'adresse OUT est mise à 1. Lorsque S n'est pas activé, OUT n'est pas modifié.
		Non disponible	Lorsque R (Mise à 0) est activé, la valeur de données à l'adresse OUT est mise à 0. Lorsque R n'est pas activé, OUT n'est pas modifié.


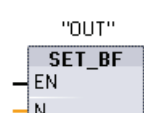

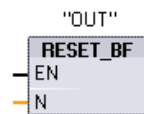
- 1 Pour CONT et LOG : Ces instructions peuvent être placées n'importe où dans le réseau.
- 2 Pour SCL : Vous devez écrire du code pour reproduire cette fonction dans votre application.

Tableau 7- 9 Types de données pour les paramètres

Paramètre	Type de données	Description
IN (ou connexion à contact/logique de porte)	Bool	Adresse de bit à visualiser
OUT	Bool	Adresse de bit à mettre à 1 ou à 0

### Mise à 1 et Mise à 0 champ de bits

Tableau 7- 10 Instructions SET\_BF et RESET\_BF :

CONT <sup>1</sup>	LOG	SCL	Description
<p>"OUT"</p>  <p>"n"</p>	<p>"OUT"</p> 	Non disponible	Lorsque SET_BF est activé, la valeur 1 est affectée à "n" bits en commençant à l'adresse OUT. Lorsque SET_BF n'est pas activé, OUT n'est pas modifié.
<p>"OUT"</p>  <p>"n"</p>	<p>"OUT"</p> 	Non disponible	RESET_BF écrit la valeur 0 dans "n" bits en commençant à l'adresse OUT. Lorsque RESET_BF n'est pas activé, OUT n'est pas modifié.

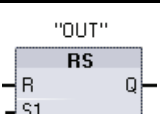

- 1 Pour CONT et LOG : Ces instructions doivent être à l'extrême droite dans une branche.
- 2 Pour SCL : Vous devez écrire du code pour reproduire cette fonction dans votre application.

Tableau 7- 11 Types de données pour les paramètres

Paramètre	Type de données	Description
OUT	Bool	Elément de départ d'un champ de bits à mettre à 1 ou à 0 (exemple : #MyArray[3])
n	Constante (UInt)	Nombre de bits à écrire

### Bascule "mise à 0, mise à 1" et Bascule "mise à 1, mise à 0"

Tableau 7- 12 Instructions RS et SR :

CONT/LOG	SCL	Description
<p>"OUT"</p> 	Non disponible	L'opération RS est une bascule avec mise à 1 prioritaire où la mise à 1 domine. Si les signaux de mise à 1 (S1) et de mise à 0 (R) sont tous deux vrais, l'adresse de sortie OUT sera à 1.
<p>"OUT"</p> 	Non disponible	L'opération SR est une bascule avec mise à 0 prioritaire où la mise à 0 domine. Si les signaux de mise à 1 (S) et de mise à 0 (R1) sont tous deux vrais, l'adresse de sortie OUT sera à 0.

- 1 Pour CONT et LOG : Ces instructions doivent être à l'extrême droite dans une branche.
- 2 Pour SCL : Vous devez écrire du code pour reproduire cette fonction dans votre application.

7.1 Instructions logiques sur bits

Tableau 7- 13 Types de données pour les paramètres


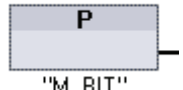

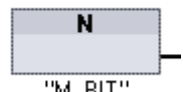
Paramètre	Type de données	Description
S, S1	Bool	Entrée de mise à 1 ; 1 indique la prédominance.
R, R1	Bool	Entrée de mise à 0 ; 1 indique la prédominance.
OUT	Bool	Sortie binaire affectée "OUT"
Q	Bool	Suit l'état du bit "OUT".

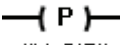
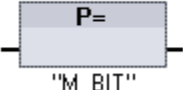
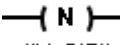
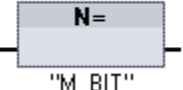
Le paramètre OUT correspond à l'adresse de bit qui est mise à 1 ou à 0. La sortie Q facultative reflète l'état de signal de l'adresse "OUT".

Instruction	S1	R	bit "OUT"
RS	0	0	Etat précédent
	0	1	0
	1	0	1
	1	1	1
SR	<b>S</b>	<b>R1</b>	
	0	0	Etat précédent
	0	1	0
	1	0	1
	1	1	0

7.1.3 Instructions Front montant et Front descendant



Tableau 7- 14 Détection de front montant et de front descendant

CONT	LOG	SCL	Description
 <p>"IN" "M_BIT"</p>	 <p>"IN" "M_BIT"</p>	Non disponible	<p>CONT : L'état de ce contact est VRAI lorsqu'un front montant (0 à 1) est détecté sur le bit "IN" affecté. L'état logique du contact est alors combiné à l'état du flux de courant entrant pour définir l'état du flux de courant sortant. Le contact P peut être situé n'importe où dans le réseau, excepté à la fin d'une branche.</p> <p>LOG : L'état logique de la sortie est VRAI lorsqu'un front montant (0 à 1) est détecté sur le bit d'entrée affecté. La boîte P ne peut être située qu'au début d'une branche.</p>
 <p>"IN" "M_BIT"</p>	 <p>"IN" "M_BIT"</p>	Non disponible	<p>CONT : L'état de ce contact est VRAI lorsqu'un front descendant (1 à 0) est détecté sur le bit d'entrée affecté. L'état logique du contact est alors combiné à l'état du flux de courant entrant pour définir l'état du flux de courant sortant. Le contact N peut être situé n'importe où dans le réseau, excepté à la fin d'une branche.</p> <p>LOG : L'état logique de la sortie est VRAI lorsqu'un front descendant (1 à 0) est détecté sur le bit d'entrée affecté. La boîte N ne peut être située qu'au début d'une branche.</p>

CONT	LOG	SCL	Description
<p>"OUT"</p>  <p>"M_BIT"</p>	<p>"OUT"</p>  <p>"M_BIT"</p>	Non disponible	<p>CONT : Le bit affecté "OUT" est VRAI lorsqu'un front montant (0 à 1) est détecté sur le flux de courant entrant dans la bobine. L'état du flux de courant entrant traverse toujours la bobine en tant qu'état du flux de courant sortant. La bobine P peut être située n'importe où dans le réseau.</p> <p>LOG : Le bit affecté "OUT" est VRAI lorsqu'un front montant (0 à 1) est détecté sur l'état logique au niveau de la connexion d'entrée de la boîte ou sur l'affectation de bit d'entrée si la boîte est située au début d'une branche. L'état logique d'entrée traverse toujours la boîte en tant qu'état logique de sortie. La boîte P= peut être située n'importe où dans la branche.</p>
<p>"OUT"</p>  <p>"M_BIT"</p>	<p>"OUT"</p>  <p>"M_BIT"</p>	Non disponible	<p>CONT : Le bit affecté "OUT" est VRAI lorsqu'un front descendant (1 à 0) est détecté sur le flux de courant entrant dans la bobine. L'état du flux de courant entrant traverse toujours la bobine en tant qu'état du flux de courant sortant. La bobine N peut être située n'importe où dans le réseau.</p> <p>LOG : Le bit affecté "OUT" est VRAI lorsqu'un front descendant (1 à 0) est détecté sur l'état logique au niveau de la connexion d'entrée de la boîte ou sur l'affectation de bit d'entrée si la boîte est située au début d'une branche. L'état logique d'entrée traverse toujours la boîte en tant qu'état logique de sortie. La boîte N= peut être située n'importe où dans la branche.</p>

1 Pour SCL : Vous devez écrire du code pour reproduire cette fonction dans votre application.

Tableau 7- 15 Instructions P\_TRIG et N\_TRIG

CONT/LOG	SCL	Description
 <p>"M_BIT"</p>	Non disponible	<p>Le flux de courant ou l'état logique de la sortie Q est VRAI lorsqu'un front montant (0 à 1) est détecté sur l'état de l'entrée CLK (LOG) ou sur le flux de courant entrant CLK (CONT).</p> <p>En CONT, l'instruction P_TRIG ne peut pas être située au début ou à la fin d'un réseau. En LOG, l'instruction P_TRIG peut être située n'importe où, excepté à la fin d'une branche.</p>
 <p>"M_BIT"</p>	Non disponible	<p>Le flux de courant ou l'état logique de la sortie Q est VRAI lorsqu'un front descendant (1 à 0) est détecté sur l'état de l'entrée CLK (LOG) ou sur le flux de courant entrant CLK (CONT).</p> <p>En CONT, l'instruction N_TRIG ne peut pas être située au début ou à la fin d'un réseau. En LOG, l'instruction N_TRIG peut être située n'importe où, excepté à la fin d'une branche.</p>

1 Pour SCL : Vous devez écrire du code pour reproduire cette fonction dans votre application.

Tableau 7- 16 Types de données pour les paramètres (contacts/bobines P et N, P=, N=, P\_TRIG et N\_TRIG)

Paramètre	Type de données	Description
M_BIT	Bool	Bit de memento dans lequel l'état précédent de l'entrée est sauvegardé
IN	Bool	Bit d'entrée dont le front de transition doit être détecté
OUT	Bool	Bit de sortie qui indique qu'un front de transition a été détecté
CLK	Bool	Flux de courant ou bit d'entrée dont le front de transition doit être détecté
Q	Bool	Sortie qui indique qu'un front a été détecté

Toutes les instructions sur front utilisent un bit de memento (M\_BIT) pour stocker l'état précédent du signal d'entrée surveillé. Un front est détecté par comparaison de l'état de l'entrée avec l'état du bit de memento. Si les états indiquent un changement de l'entrée dans la direction concernée, un front est signalé par écriture de la valeur VRAI dans la sortie. Sinon, FAUX est écrit dans la sortie.

---

**Remarque**

Les instructions sur front évaluent les valeurs de l'entrée et du bit de memento à chaque fois qu'elles sont exécutées, première exécution comprise. Vous devez prendre en compte l'état initial de l'entrée et du bit de memento dans la conception de votre programme et soit autoriser soit empêcher la détection de front lors du premier cycle.

Comme le bit de memento doit être conservé d'une exécution à la suivante, vous devez utiliser pour chaque instruction sur front un bit unique qui ne doit pas servir à un autre endroit dans le programme. Evitez également d'utiliser de la mémoire temporaire ou de la mémoire pouvant être affectée par d'autres fonctions système, telles que l'actualisation des E/S. Utilisez uniquement de la mémoire M, DB global ou statique (dans un DB d'instance) pour les affectations de M\_BIT.

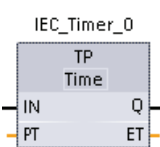
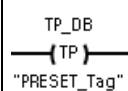
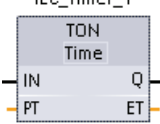
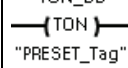
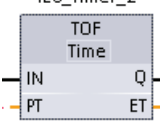
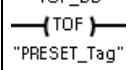
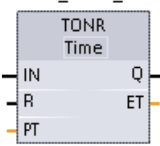
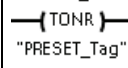
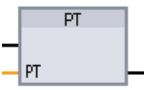
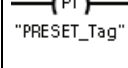
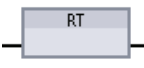

---



## 7.2 Temporisations

Vous utilisez les temporisations pour créer des retards programmés. Le nombre de temporisations que vous pouvez utiliser dans votre programme utilisateur est limité uniquement par la quantité de mémoire dans la CPU. Chaque temporisation utilise une structure de DB de 16 octets de type de données IEC\_Timer pour conserver les données de la temporisation, cette structure étant indiquée au-dessus de la boîte ou de la bobine. STEP 7 crée automatiquement le DB lorsque vous insérez l'instruction.

Tableau 7- 17 Temporisations

Boîtes CONT/LOG	Bobines CONT	SCL	Description
 <p>IEC_Timer_0 TP Time IN Q PT ET</p>	 <p>TP_DB —{TP}— "PRESET_Tag"</p>	<pre>"IEC_Timer_0_DB".TP (   IN:=_bool_in_,   PT:=_time_in_,   Q=&gt;_bool_out_,   ET=&gt;_time_out_);</pre>	La temporisation TP génère une impulsion de durée prédéfinie.
 <p>IEC_Timer_1 TON Time IN Q PT ET</p>	 <p>TON_DB —{TON}— "PRESET_Tag"</p>	<pre>"IEC_Timer_0_DB".TON (   IN:=_bool_in_,   PT:=_time_in_,   Q=&gt;_bool_out_,   ET=&gt;_time_out_);</pre>	La temporisation TON met la sortie Q à 1 après un temps de retard prédéfini.
 <p>IEC_Timer_2 TOF Time IN Q PT ET</p>	 <p>TOF_DB —{TOF}— "PRESET_Tag"</p>	<pre>"IEC_Timer_0_DB".TOF (   IN:=_bool_in_,   PT:=_time_in_,   Q=&gt;_bool_out_,   ET=&gt;_time_out_);</pre>	La temporisation TOF met la sortie Q à 0 après un temps de retard prédéfini.
 <p>IEC_Timer_3 TONR Time IN Q R ET PT</p>	 <p>TONR_DB —{TONR}— "PRESET_Tag"</p>	<pre>"IEC_Timer_0_DB".TONR (   IN:=_bool_in_,   R:=_bool_in_,   PT:=_time_in_,   Q=&gt;_bool_out_,   ET=&gt;_time_out_);</pre>	La temporisation TONR met la sortie Q à 1 après un temps de retard prédéfini. Le temps écoulé est accumulé sur plusieurs périodes de temps jusqu'à ce que l'entrée R soit utilisée pour réinitialiser le temps écoulé.
<p>LOG uniquement :</p>  <p>PT PT</p>	 <p>TON_DB —{PT}— "PRESET_Tag"</p>	Pas d'équivalent SCL	La bobine PT (Charger durée) charge une nouvelle valeur de temps prédéfinie dans la temporisation IEC_Timer spécifiée.
<p>LOG uniquement :</p>  <p>RT</p>	 <p>TON_DB —[RT]—</p>	Pas d'équivalent SCL	La bobine RT (Réinitialiser temporisation) réinitialise la temporisation IEC_Timer spécifiée.

- STEP 7 crée automatiquement le DB lorsque vous insérez l'instruction.
- Dans les exemples SCL, "IEC\_Timer\_0\_DB" est le nom du DB d'instance.

Tableau 7- 18 Types de données pour les paramètres

Paramètre	Type de données	Description
Boîte : IN Bobine : Flux de courant	Bool	TP, TON et TONR : Boîte : 0=désactiver la temporisation, 1=activer la temporisation Bobine : pas de flux de courant=désactiver la temporisation, flux de courant=activer la temporisation TOF : Boîte : 0=activer la temporisation, 1=désactiver la temporisation Bobine : pas de flux de courant=activer la temporisation, flux de courant=désactiver la temporisation
R	Bool	Boîte TONR uniquement : 0=pas de réinitialisation 1=remettre le temps écoulé et le bit Q à 0
Boîte : PT Bobine : "PRESET_Tag"	Time	Boîte ou bobine de temporisation : Entrée valeur de temps prédéfinie
Boîte : Q Bobine : DBdata.Q	Bool	Boîte de temporisation : sortie de boîte Q ou bit Q dans les données du DB de temporisation Bobine de temporisation : vous pouvez accéder au bit Q uniquement dans les données du DB de temporisation
Boîte : ET Bobine : DBdata.ET	Time	Boîte de temporisation : sortie de boîte ET (temps écoulé) ou valeur de temps ET dans les données du DB de temporisation Bobine de temporisation : vous pouvez accéder à la valeur de temps ET uniquement dans les données du DB de temporisation

Tableau 7- 19 Effet des modifications de valeur dans les paramètres PT et IN

Temporisation	Modifications des paramètres de boîte PT et IN et des paramètres de bobine correspondants
TP	<ul style="list-style-type: none"> <li>La modification de PT n'a pas d'effet pendant que la temporisation s'exécute.</li> <li>La modification de IN n'a pas d'effet pendant que la temporisation s'exécute.</li> </ul>
TON	<ul style="list-style-type: none"> <li>La modification de PT n'a pas d'effet pendant que la temporisation s'exécute.</li> <li>Le passage de IN à FAUX pendant que la temporisation s'exécute réinitialise et arrête la temporisation.</li> </ul>
TOF	<ul style="list-style-type: none"> <li>La modification de PT n'a pas d'effet pendant que la temporisation s'exécute.</li> <li>Le passage de IN à VRAI pendant que la temporisation s'exécute réinitialise et arrête la temporisation.</li> </ul>
TONR	<ul style="list-style-type: none"> <li>La modification de PT n'a pas d'effet pendant que la temporisation s'exécute mais a un effet lorsque la temporisation reprend.</li> <li>Le passage de IN à FAUX pendant que la temporisation s'exécute arrête la temporisation mais ne la réinitialise pas. Si IN repasse à VRAI, la temporisation recommence à s'écouler à partir de la valeur de temps accumulée.</li> </ul>

Les valeurs PT (temps prédéfini) et ET (temps écoulé) sont sauvegardées dans les données DB IEC\_TIMER spécifiées en tant qu'entiers doubles signés représentant des millisecondes. Les données TIME utilisent l'identificateur T# et peuvent être entrées en tant qu'unités de temps simple (T#200ms ou 200) ou unités de temps composées (T#2s\_200ms).

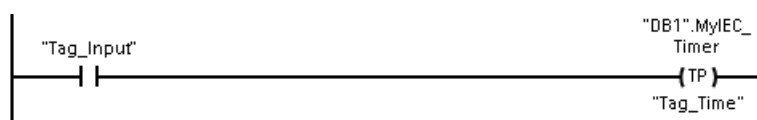
Tableau 7- 20 Taille et plage du type de données de TIME

Type de données	Taille	Plages valides <sup>1</sup>
TIME	32 bits, stockés comme données DInt	T#-24d_20h_31m_23s_648ms à T#24d_20h_31m_23s_647ms Stockés comme -2 147 483 648 ms à +2 147 483 647 ms

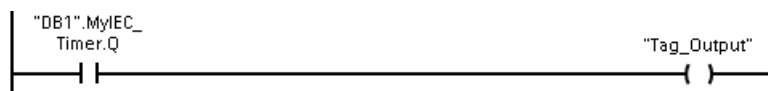
<sup>1</sup> La plage négative du type de données TIME indiquée ci-dessus ne peut pas être utilisée avec les temporisations. Les valeurs négatives de PT (temps prédéfini) sont mises à zéro lorsque l'instruction de temporisation est exécutée. ET (temps écoulé) est toujours une valeur positive.

### Exemple de bobine de temporisation

Les bobines de temporisation -(TP)-, -(TON)-, -(TOF)- et -(TONR)- doivent être la dernière instruction dans un réseau CONT. Comme illustré dans l'exemple de temporisation, un contact dans un réseau suivant évalue le bit Q dans les données DB IEC\_Timer de la bobine de temporisation. De même, vous devez accéder à l'élément ELAPSED dans les données DB IEC\_timer si vous souhaitez utiliser la valeur de temps écoulé dans votre programme.



La temporisation d'impulsion démarre lors d'un front montant de la valeur de bit Tag\_Input. La temporisation s'écoule pendant la durée indiquée par la valeur de temps Tag\_Time.



Tant que la temporisation s'exécute, l'état de DB1.MyIEC\_Timer.Q est égal à 1 et la valeur de Tag\_Output est égale à 1. Lorsque la valeur Tag\_Time est écoulée, DB1.MyIEC\_Timer.Q est égal à 0 et la valeur de Tag\_Output est égale à 0.

**Bobines Réinitialiser temporisation -(RT)- et Charger durée -(PT)-**

Ces bobines peuvent être utilisées avec des boîtes ou des bobines de temporisation et peuvent être placées en position médiane. L'état du flux de courant sortant de la bobine est toujours le même que l'état à l'entrée de la bobine. Lorsque la bobine -(RT)- est activée, l'élément de temps ELAPSED des données DB IEC\_Timer indiquées est remis à 0. Lorsque la bobine -(PT)- est activée, l'élément de temps PRESET des données DB IEC\_Timer indiquées est remis à 0.

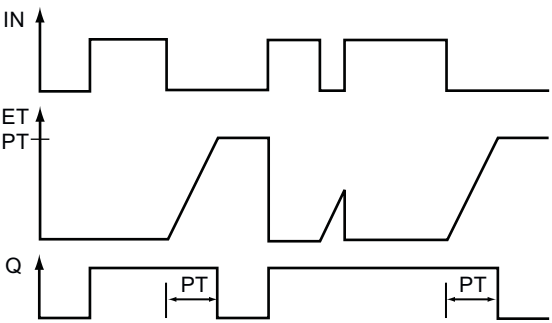
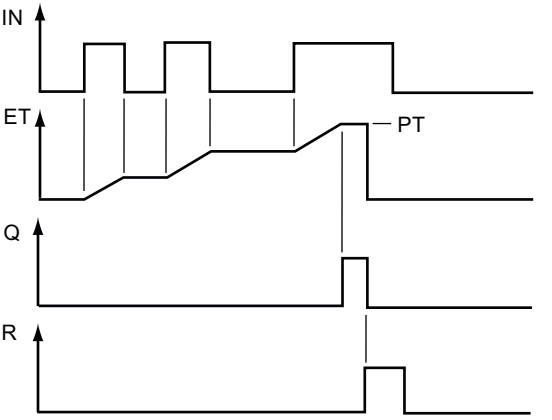
**Remarque**

Lorsque vous placez des temporisations dans un FB, vous pouvez sélectionner l'option "Bloc de données multi-instance". Les noms des structures de temporisation peuvent être différents avec des structures de données distinctes, mais les données de temporisation sont stockées dans un bloc de données unique et ne nécessitent pas de bloc de données séparé pour chaque temporisation. Cela réduit le temps de traitement et l'espace mémoire nécessaire pour la gestion des temporisations. Il n'y a pas d'interaction entre les structures des données de temporisation dans le DB multi-instance partagé.

**Fonctionnement des temporisations**

Tableau 7- 21 Types de temporisations CEI

Temporisation	Chronogramme
<p><b>TP</b> : temporisation sous forme d'impulsion                      La temporisation TP génère une impulsion de durée prédéfinie.</p>	<p>The diagram shows three input pulses (IN). For each pulse, the timer value (ET/PT) ramps up linearly from 0 to the preset time (PT). The output (Q) is a rectangular pulse that starts at the beginning of the input pulse and ends exactly at the time PT. The PT intervals are marked with arrows at the bottom of the Q signal.</p>
<p><b>TON</b> : temporisation sous forme de retard à la montée                      La temporisation TON met la sortie Q à 1 après un temps de retard prédéfini.</p>	<p>The diagram shows three input pulses (IN). For each pulse, the timer value (ET/PT) ramps up linearly from 0 to the preset time (PT). The output (Q) remains at 0 until the timer reaches PT, at which point it transitions to 1 and stays there for the duration of the input pulse. The PT intervals are marked with arrows at the bottom of the Q signal.</p>

Temporisation	Chronogramme
<p><b>TOF</b> : temporisation sous forme de retard à la retombée</p> <p>La temporisation TOF met la sortie Q à 0 après un temps de retard prédéfini.</p>	
<p><b>TONR</b> : temporisation sous forme de retard à la montée mémorisé</p> <p>La temporisation TONR met la sortie Q à 1 après un temps de retard prédéfini. Le temps écoulé est accumulé sur plusieurs périodes de temps jusqu'à ce que l'entrée R soit utilisée pour réinitialiser le temps écoulé.</p>	

### Remarque

Dans la CPU, aucune ressource réservée n'est allouée à une temporisation spécifique. Au lieu de cela, chaque temporisation utilise sa propre structure de temporisation dans la mémoire DB et une temporisation CPU interne s'exécutant en continu pour effectuer la synchronisation.

Lorsqu'une temporisation est démarrée en raison d'un front à l'entrée d'une instruction TP, TON, TOF ou TONR, la valeur de la temporisation CPU interne s'exécutant en continu est copiée dans l'élément START de la structure de DB allouée à cette instruction de temporisation. Cette valeur de démarrage reste inchangée tant que la temporisation continue à s'exécuter et est utilisée plus tard à chaque fois que la temporisation est actualisée. A chaque fois que la temporisation est démarrée, une nouvelle valeur de démarrage est chargée de la temporisation CPU interne dans la structure de la temporisation.

Lorsqu'une temporisation est actualisée, la valeur de démarrage décrite ci-dessus est soustraite de la valeur en cours de la temporisation CPU interne pour déterminer le temps écoulé. Le temps écoulé est alors comparé à la valeur prédéfinie pour déterminer l'état du bit Q de la temporisation. Les éléments ELAPSED et Q sont alors actualisés dans la structure de DB allouée à cette temporisation. Notez que le temps écoulé est limité à la valeur prédéfinie (la temporisation ne continue pas à cumuler le temps écoulé une fois la valeur prédéfinie atteinte).

Une temporisation est actualisée uniquement lorsque :

- une instruction de temporisation (TP, TON, TOF ou TONR) est exécutée,
- l'élément ELAPSED de la structure de temporisation dans le DB est référencé directement par une instruction,
- l'élément Q de la structure de temporisation dans le DB est référencé directement par une instruction.

### Programmation des temporisations

Vous devez tenir compte des conséquences suivantes du fonctionnement des temporisations lors de la planification et de la création de votre programme utilisateur :

- Plusieurs actualisations d'une temporisation peuvent avoir lieu dans le même cycle. La temporisation est actualisée à chaque fois que l'instruction de temporisation (TP, TON, TOF, TONR) est exécutée et à chaque fois que l'élément ELAPSED ou Q de la structure de temporisation est utilisé comme paramètre d'une autre instruction exécutée. Cela constitue un avantage si vous voulez disposer des dernières données de temps (sensiblement une lecture directe de la temporisation). Toutefois, si vous désirez avoir des valeurs cohérentes durant tout un cycle de programme, vous devez placer votre instruction de temporisation avant toutes les autres instructions ayant besoin de ces valeurs et utiliser des variables connectées aux sorties Q et ET de l'instruction de temporisation plutôt que les éléments ELAPSED et Q de la structure de DB de temporisation.
- Il peut y avoir des cycles pendant lesquels une temporisation n'est pas actualisée. Il est possible de démarrer une temporisation dans une fonction et de ne plus appeler cette fonction pendant un ou plusieurs cycles. Si aucune autre instruction référençant les éléments ELAPSED ou Q de la structure de temporisation n'est exécutée, la temporisation n'est pas actualisée. Il n'y aura pas d'actualisation tant que l'instruction de temporisation n'est pas réexécutée ou qu'une autre instruction utilisant l'élément ELAPSED ou Q de la structure de temporisation comme paramètre n'est pas exécutée.

- Bien que cela ne soit pas typique, vous pouvez affecter la même structure de temporisation à plusieurs instructions de temporisation. Mais en général, pour éviter des interactions imprévisibles, vous n'utiliserez qu'une instruction de temporisation (TP, TON, TOF, TONR) par structure de temporisation.
- Les temporisations à auto-réinitialisation sont utiles pour déclencher des actions qui doivent se produire périodiquement. Typiquement, vous créez des temporisations à auto-réinitialisation en plaçant un contact à ouverture qui référence le bit de temporisation devant l'instruction de temporisation. Ce réseau de temporisation est typiquement situé au-dessus d'un ou de plusieurs réseaux dépendants qui utilisent le bit de temporisation pour déclencher des actions. Lorsque la temporisation expire (le temps écoulé atteint la valeur prédéfinie), le bit de temporisation est à 1 pour un cycle, permettant ainsi à la logique de réseau dépendante pilotée par le bit de temporisation de s'exécuter. Lors de l'exécution suivante du réseau de temporisation, le contact à ouverture est désactivé, ce qui réinitialise la temporisation et efface le bit de temporisation. Au cycle suivant, le contact à ouverture est activé ce qui redémarre la temporisation. Lors de la création d'une telle temporisation à auto-réinitialisation, n'utilisez pas l'élément Q de la structure de DB de temporisation comme paramètre pour le contact à ouverture devant l'instruction de temporisation, mais utilisez la variable connectée à la sortie Q de l'instruction de temporisation. En effet, il ne faut pas accéder à l'élément Q de la structure de DB de temporisation, car cela provoque une actualisation de la temporisation et que, si la temporisation est actualisée en raison du contact à ouverture, le contact réinitialisera l'instruction de temporisation immédiatement. La sortie Q de l'instruction de temporisation ne sera pas à 1 pour un cycle et les réseaux dépendants ne s'exécuteront pas.

### Conservation des données de temporisation après une transition MARCHE-ARRET-MARCHE ou une mise hors tension puis sous tension de la CPU

Si une session MARCHE se termine par passage à l'état ARRET ou mise hors tension puis sous tension de la CPU et qu'une nouvelle session MARCHE est démarrée, les données de temporisation sauvegardées à l'état MARCHE précédent sont perdues à moins que la structure de données de temporisation n'ait été définie comme rémanente (temporisations TP, TON, TOF et TONR).

Lorsque vous acceptez les valeurs par défaut dans la boîte de dialogue d'options d'appel après avoir inséré une temporisation dans l'éditeur de programmes, un DB d'instance qui **ne peut pas être défini comme rémanent** vous est automatiquement affecté. Pour assurer la rémanence de vos données de temporisation, vous devez donc utiliser soit un DB global, soit un DB multi-instance.

### Affectation d'un DB global pour sauvegarder les données de temporisation comme données rémanentes

Cette option fonctionne quel que soit l'endroit où se trouve la temporisation (OB, FC ou FB).

1. Créez un DB global :

- Double-cliquez sur "Ajouter nouveau bloc" dans l'arborescence de projet.
  - Cliquez sur l'icône de bloc de données (DB).
  - Choisissez DB global comme type.
  - Vérifiez que la case "Optimisé" est cochée pour le type de DB si vous désirez pouvoir définir des éléments individuels de ce DB comme rémanents. L'autre option de type de DB "Standard - compatible avec S7-300/400" permet uniquement de définir tous les éléments de DB soit comme rémanents, soit comme non rémanents.
  - Cliquez sur OK.
2. Ajoutez une ou des structures de temporisation au DB :
- Dans le nouveau DB global, ajoutez une nouvelle variable statique en utilisant le type de données IEC\_Timer.
  - Cochez la case dans la colonne "Rémanence" afin que cette structure soit rémanente.
  - Répétez cette procédure pour créer des structures pour toutes les temporisations que vous voulez sauvegarder dans ce DB. Vous pouvez soit placer chaque structure de temporisation dans un DB global unique, soit placer plusieurs structures de temporisation dans le même DB global. Vous pouvez également placer d'autres variables statiques en plus des temporisations dans ce DB global. Insérer plusieurs structures de temporisation dans le même DB global permet de réduire le nombre total de vos blocs.
  - Renommez les structures de temporisation si vous le souhaitez.
3. Ouvrez dans l'éditeur le bloc de programme dans lequel vous voulez placer une temporisation rémanente (OB, FC ou FB).
4. Placez la temporisation à l'emplacement désiré.
5. Lorsque la boîte de dialogue des options d'appel s'affiche, cliquez sur le bouton Annuler.
6. Au-dessus de la nouvelle temporisation, tapez le nom (sans utiliser l'icône d'aide à la navigation) du DB global et de la structure de temporisation que vous avez créés auparavant (exemple : "Data\_block\_3.Static\_1").

### Affectation d'un DB multi-instance pour sauvegarder les données de temporisation comme données rémanentes

Cette option ne fonctionne que si vous placez la temporisation dans un FB.

Cette option dépend du fait que le FB a ou non été créé avec accès "optimisé" (accès symbolique uniquement). Une fois le FB créé, vous ne pouvez plus modifier la case à cocher pour "Optimisé". Vous devez faire le choix correct au moment où vous créez le FB, sur le premier écran après avoir sélectionné "Ajouter nouveau bloc" dans l'arborescence. Pour vérifier comment l'attribut d'accès est configuré pour un FB existant, cliquez avec le bouton droit de la souris sur le FB dans l'arborescence de projet, sélectionnez "Propriétés", puis sélectionnez "Attributs".

Si le FB a été créé avec la case "Optimisé" cochée (accès symbolique uniquement) :

1. Ouvrez le FB dans l'éditeur.
2. Placez la temporisation à l'emplacement désiré dans le FB.



3. Lorsque la boîte de dialogue des options d'appel s'affiche, cliquez sur l'icône "Multiinstance". L'option Multiinstance n'est disponible que si l'instruction est insérée dans un FB.
4. Dans la boîte de dialogue des options d'appel, renommez la temporisation si vous le souhaitez.
5. Cliquez sur OK. L'instruction de temporisation apparaît dans l'éditeur et la structure IEC\_TIMER apparaît dans l'interface de FB sous Statique.
6. Si nécessaire, ouvrez l'éditeur d'interface de FB (vous devrez peut-être cliquer sur la petite flèche pour agrandir la vue).
7. Sous Statique, localisez la structure de temporisation qui vient d'être créée pour vous.
8. Dans la colonne Rémanence de cette structure de temporisation, changez la sélection à "Rémanence". Plus tard, à chaque fois que ce FB sera appelé dans un autre bloc de programme, un DB d'instance sera créé avec cette définition d'interface qui contient la structure de temporisation définie comme rémanente.

Si le FB a été créé avec la case "Standard - compatible avec S7-300/400" cochée (accès symbolique et accès direct) :

1. Ouvrez le FB dans l'éditeur.
2. Placez la temporisation à l'emplacement désiré dans le FB.
3. Lorsque la boîte de dialogue des options d'appel s'affiche, cliquez sur l'icône "Multiinstance". L'option Multiinstance n'est disponible que si l'instruction est insérée dans un FB.
4. Dans la boîte de dialogue des options d'appel, renommez la temporisation si vous le souhaitez.
5. Cliquez sur OK. L'instruction de temporisation apparaît dans l'éditeur et la structure IEC\_TIMER apparaît dans l'interface de FB sous Statique.
6. Ouvrez le bloc qui utilisera ce FB.
7. Placez ce FB à l'emplacement désiré. Cela entraîne la création d'un bloc de données d'instance pour ce FB.
8. Ouvrez le bloc de données d'instance qui a été créé lorsque vous avez placé le FB dans l'éditeur.
9. Sous Statique, localisez la structure de temporisation qui vous intéresse. Dans la colonne Rémanence de cette structure de temporisation, cochez la case pour rendre cette structure rémanente.

## 7.3 Compteurs

Tableau 7- 22 Compteurs

CONT/LOG	SCL	Description
	<pre>"IEC_Counter_0_DB".CTU(   CU:=_bool_in,   R:=_bool_in,   PV:=_int_in,   Q=&gt;_bool_out,   CV=&gt;_int_out);</pre>	<p>Utilisez les compteurs pour compter des événements de programme internes et des événements de processus externes. Chaque compteur utilise une structure sauvegardée dans un bloc de données afin de conserver les données du compteur. Vous affectez le bloc de données lors du placement du compteur dans l'éditeur.</p> <ul style="list-style-type: none"> <li>• CTU est un compteur de comptage.</li> <li>• CTD est un compteur de décomptage.</li> <li>• CTUD est un compteur de comptage et de décomptage.</li> </ul>
	<pre>"IEC_Counter_0_DB".CTD(   CD:=_bool_in,   LD:=_bool_in,   PV:=_int_in,   Q=&gt;_bool_out,   CV=&gt;_int_out);</pre>	
	<pre>"IEC_Counter_0_DB".CTUD(   CU:=_bool_in,   CD:=_bool_in,   R:=_bool_in,   LD:=_bool_in,   PV:=_int_in,   QU=&gt;_bool_out,   QD=&gt;_bool_out,   CV=&gt;_int_out);</pre>	

- 1 Pour CONT et LOG : Sélectionnez le type de données de la valeur de comptage dans la liste déroulante sous le nom de l'instruction.
- 2 STEP 7 crée automatiquement le DB lorsque vous insérez l'instruction.
- 3 Dans les exemples SCL, "IEC\_Counter\_0\_DB" est le nom du DB d'instance.

Tableau 7- 23 Types de données pour les paramètres

Paramètre	Type de données <sup>1</sup>	Description
CU, CD	Bool	Compter ou décompter, d'une valeur de 1
R (CTU, CTUD)	Bool	Remise à zéro de la valeur de comptage
LD (CTD, CTUD)	Bool	Commande de chargement pour la valeur prédéfinie
PV	SInt, Int, DInt, USInt, UInt, UDIInt	Valeur de comptage prédéfinie
Q, QU	Bool	Vrai si CV >= PV
QD	Bool	Vrai si CV <= 0
CV	SInt, Int, DInt, USInt, UInt, UDIInt	Valeur de comptage en cours

<sup>1</sup> La plage numérique des valeurs de comptage dépend du type de données que vous sélectionnez. Si la valeur de comptage est un entier non signé, vous pouvez décompter jusqu'à zéro ou compter jusqu'à la limite de plage. Si la valeur de comptage est un entier signé, vous pouvez décompter jusqu'à la limite entière négative et compter jusqu'à la limite entière positive.

Le nombre de compteurs que vous pouvez utiliser dans votre programme utilisateur est limité uniquement par la quantité de mémoire dans la CPU. Les compteurs utilisent la quantité de mémoire suivante :

- Un compteur utilise 3 octets pour les types de données SInt ou USInt.
- Un compteur utilise 6 octets pour les types de données Int ou UInt.
- Un compteur utilise 12 octets pour les types de données DInt ou UInt.

Ces instructions utilisent des compteurs logiciels dont la vitesse de comptage maximale est limitée par la vitesse d'exécution de l'OB dans lequel ils se trouvent. Il faut exécuter l'OB dans lequel se trouvent les instructions suffisamment souvent pour détecter toutes les transitions des entrées CU ou CD. Utilisez l'instruction CTRL\_HSC (Page 359) pour disposer d'opérations de comptage plus rapides.

### Remarque

Lorsque vous placez des compteurs dans un FB, vous pouvez sélectionner l'option "DB multi-instance". Les noms des structures de compteur peuvent alors être différents avec des structures de données distinctes, mais les données de compteur sont contenues dans un seul DB et vous n'avez pas besoin d'un DB distinct pour chaque compteur. Cela réduit le temps de traitement et l'espace mémoire nécessaire pour les compteurs. Il n'y a pas d'interaction entre les structures de données de compteur dans le DB multi-instance partagé.

## Fonctionnement des compteurs

Tableau 7- 24 Fonctionnement du compteur CTU

Compteur	Fonctionnement
<p>Le compteur CTU incrémente de 1 lorsque la valeur du paramètre CU passe de 0 à 1. Le chronogramme illustre le fonctionnement d'un compteur CTU avec une valeur de comptage entière non signée (PV = 3).</p> <ul style="list-style-type: none"> <li>• Si la valeur du paramètre CV (valeur de comptage en cours) est supérieure ou égale à la valeur du paramètre PV (valeur prédéfinie), le paramètre de sortie Q du compteur est égal à 1.</li> <li>• Si la valeur du paramètre de réinitialisation R passe de 0 à 1, la valeur de comptage en cours est remise à 0.</li> </ul>	<p>The diagram shows the relationship between the counter's inputs and outputs. The CU signal (clock) has four rising edges. The R signal (reset) is 0 until the fourth rising edge of CU, then pulses high. The CV signal (current count) starts at 0 and increments to 1, 2, 3, and 4 at each rising edge of CU. The Q signal (output) is 0 until CV reaches 3, then becomes 1. When R pulses high, CV resets to 0 and Q returns to 0.</p>

7.3 Compteurs

Tableau 7- 25 Fonctionnement du compteur CTD

Compteur	Fonctionnement
<p>Le compteur CTD décrémente de 1 lorsque la valeur du paramètre CD passe de 0 à 1. Le chronogramme illustre le fonctionnement d'un compteur CTD avec une valeur de comptage entière non signée (PV = 3).</p> <ul style="list-style-type: none"> <li>• Si la valeur du paramètre CV (valeur de comptage en cours) est inférieure ou égale à 0, le paramètre de sortie Q du compteur est égal à 1.</li> <li>• Si la valeur du paramètre LOAD passe de 0 à 1, la valeur dans le paramètre PV (valeur prédéfinie) est chargée dans le compteur en tant que nouvelle valeur de comptage en cours CV.</li> </ul>	

Tableau 7- 26 Fonctionnement du compteur CTUD

Compteur	Fonctionnement
<p>Le compteur CTUD incrémente ou décrémente de 1 en cas de passage de 0 à 1 de l'entrée de comptage (CU) ou de décomptage (CD). Le chronogramme illustre le fonctionnement d'un compteur CTUD avec une valeur de comptage entière non signée (PV = 4).</p> <ul style="list-style-type: none"> <li>• Si la valeur du paramètre CV est supérieure ou égale à la valeur du paramètre PV (valeur prédéfinie), le paramètre de sortie QU du compteur est égal à 1.</li> <li>• Si la valeur du paramètre CV est inférieure ou égale à zéro, le paramètre de sortie QD du compteur est égal à 1.</li> <li>• Si la valeur du paramètre LOAD passe de 0 à 1, la valeur dans le paramètre PV est chargée dans le compteur en tant que nouvelle valeur de comptage en cours CV.</li> <li>• Si la valeur du paramètre de réinitialisation R passe de 0 à 1, la valeur de comptage en cours est remise à 0.</li> </ul>	

### Conservation des données de compteur après une transition MARCHE-ARRET-MARCHE ou une mise hors tension puis sous tension de la CPU

Si une session MARCHE se termine par passage à l'état ARRET ou mise hors tension puis sous tension de la CPU et qu'une nouvelle session MARCHE est démarrée, les données de compteur sauvegardées à l'état MARCHE précédent sont perdues à moins que la structure de données de compteur n'ait été définie comme rémanente (compteurs CTU, CTD et CTUD).

Lorsque vous acceptez les valeurs par défaut dans la boîte de dialogue d'options d'appel après avoir inséré un compteur dans l'éditeur de programmes, un DB d'instance qui **ne peut pas être défini comme rémanent** vous est automatiquement affecté. Pour assurer la rémanence de vos données de compteur, vous devez donc utiliser soit un DB global, soit un DB multi-instance.

### Affectation d'un DB global pour sauvegarder les données de compteur comme données rémanentes

Cette option fonctionne quel que soit l'endroit où se trouve le compteur (OB, FC ou FB).

1. Créez un DB global :

- Double-cliquez sur "Ajouter nouveau bloc" dans l'arborescence de projet.
- Cliquez sur l'icône de bloc de données (DB).
- Choisissez DB global comme type.
- Vérifiez que la case "Adressage symbolique uniquement" est cochée si vous désirez pouvoir définir des éléments individuels de ce DB comme rémanents.
- Cliquez sur OK.

2. Ajoutez une ou des structures de compteur au DB :

- Dans le nouveau DB global, ajoutez une nouvelle variable statique en utilisant l'un des types de données de compteur. Veillez à tenir compte du type que vous voulez utiliser pour la valeur prédéfinie et la valeur de comptage en cours.

Type de données compteur	Type correspondant pour les valeurs prédéfinie et de comptage en cours
IEC_Counter	INT
IEC_SCounter	SINT
IEC_DCounter	DINT
IEC_UCounter	UINT
IEC_USCounter	USINT
IEC_UDCounter	UDINT

1. Cochez la case dans la colonne "Rémanence" afin que cette structure soit rémanente.
  - Répétez cette procédure pour créer des structures pour tous les compteurs que vous voulez sauvegarder dans ce DB. Vous pouvez soit placer chaque structure de compteur dans un DB global unique, soit placer plusieurs structures de compteur dans le même DB global. Vous pouvez également placer d'autres variables statiques en plus des compteurs dans ce DB global. Insérer plusieurs structures de compteur dans le même DB global permet de réduire le nombre total de vos blocs.
  - Renommez les structures de compteur si vous le souhaitez.
2. Ouvrez dans l'éditeur le bloc de programme dans lequel vous voulez placer un compteur rémanent (OB, FC ou FB).
3. Placez le compteur à l'emplacement désiré.
4. Lorsque la boîte de dialogue des options d'appel s'affiche, cliquez sur le bouton Annuler. Vous devriez maintenant voir un nouveau compteur avec "???" juste au-dessus et en dessous du nom d'instruction.
5. Au-dessus du nouveau compteur, tapez le nom (sans utiliser l'icône d'aide à la navigation) du DB global et de la structure de compteur que vous avez créés auparavant (exemple : "Data\_block\_3.Static\_1"). Le type correspondant de la valeur prédéfinie et de la valeur en cours est alors complété (exemple : UInt pour une structure IEC\_UCounter).

### Affectation d'un DB multi-instance pour sauvegarder les données de compteur comme données rémanentes

Cette option ne fonctionne que si vous placez le compteur dans un FB.

Cette option dépend du fait que le FB a ou non été créé avec accès symbolique uniquement. Une fois le FB créé, vous ne pouvez plus modifier la case à cocher pour "Adressage symbolique uniquement". Vous devez faire le choix correct au moment où vous créez le FB, sur le premier écran après avoir sélectionné "Ajouter nouveau bloc" dans l'arborescence. Pour vérifier la configuration de cette case pour un FB existant, cliquez avec le bouton droit de la souris sur le FB dans l'arborescence de projet, sélectionnez "Propriétés", puis sélectionnez "Attributs".

Si le FB a été créé avec la case "Adressage symbolique uniquement" cochée :

1. Ouvrez le FB dans l'éditeur.
2. Placez le compteur à l'emplacement désiré dans le FB.
3. Lorsque la boîte de dialogue des options d'appel s'affiche, cliquez sur l'icône "Multiinstance". L'option Multiinstance n'est disponible que si l'instruction est insérée dans un FB.
4. Dans la boîte de dialogue des options d'appel, renommez le compteur si vous le souhaitez.
5. Cliquez sur OK. L'instruction de compteur apparaît dans l'éditeur avec le type INT pour la valeur prédéfinie et la valeur de comptage et la structure IEC\_COUNTER apparaît dans l'interface de FB sous Statique.
6. Si désiré, changez le type INT dans le compteur en l'un des autres types proposés. La structure de compteur changera de manière correspondante.

Type indiqué dans l'instruction de compteur (pour la valeur prédéfinie et la valeur de comptage)	Type de structure correspondante montré dans l'interface de FB
--------------------------------------------------------------------------------------------------------	-------------------------------------------------------------------

INT	IEC_Counter
SINT	IEC_SCounter
DINT	IEC_DCounter
UINT	IEC_UCounter
USINT	IEC_USCounter
UDINT	IEC_UDCounter

1. Si nécessaire, ouvrez l'éditeur d'interface de FB (vous devrez peut-être cliquer sur la petite flèche pour agrandir la vue).
2. Sous Statique, localisez la structure de compteur qui vient d'être créée pour vous.
3. Dans la colonne Rémanence de cette structure de compteur, changez la sélection à "Rémanence". Plus tard, à chaque fois que ce FB sera appelé dans un autre bloc de programme, un DB d'instance sera créé avec cette définition d'interface qui contient la structure de compteur définie comme rémanente.

Si le FB a été créé avec la case "Adressage symbolique uniquement" *non* cochée :

1. Ouvrez le FB dans l'éditeur.
2. Placez le compteur à l'emplacement désiré dans le FB.
3. Lorsque la boîte de dialogue des options d'appel s'affiche, cliquez sur l'icône "Multiinstance". L'option Multiinstance n'est disponible que si l'instruction est insérée dans un FB.
4. Dans la boîte de dialogue des options d'appel, renommez le compteur si vous le souhaitez.

7.4 Comparaison

5. Cliquez sur OK. L'instruction de compteur apparaît dans l'éditeur avec le type INT pour la valeur prédéfinie et la valeur de comptage et la structure IEC\_COUNTER apparaît dans l'interface de FB sous Statique.
6. Si désiré, changez le type INT dans le compteur en l'un des autres types proposés. La structure de compteur changera de manière correspondante.

Type indiqué dans l'instruction de compteur (pour la valeur prédéfinie et la valeur de comptage)      Type de structure correspondante montré dans l'interface de FB

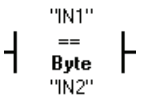
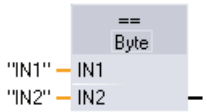
INT	IEC_Counter
SINT	IEC_SCounter
DINT	IEC_DCounter
UINT	IEC_UCounter
USINT	IEC_USCounter
UDINT	IEC_UDCounter

1. Ouvrez le bloc qui utilisera ce FB.
2. Placez ce FB à l'emplacement désiré. Cela entraîne la création d'un bloc de données d'instance pour ce FB.
3. Ouvrez le bloc de données d'instance qui a été créé lorsque vous avez placé le FB dans l'éditeur.
4. Sous Statique, localisez la structure de compteur qui vous intéresse. Dans la colonne Rémanence de cette structure de compteur, cochez la case pour rendre cette structure rémanente.

## 7.4 Comparaison

### 7.4.1 Comparaison

Tableau 7- 27 Instructions de comparaison

CONT	LOG	SCL	Description
		<pre> out := in1 = in2; or IF in1 = in2   THEN out := 1;   ELSE out := 0; END IF;                     </pre>	<p>Compare deux valeurs ayant le même type de données. Lorsque la comparaison est vraie, le contact de comparaison CONT est activé. Lorsque la comparaison est vraie, la sortie de la boîte de comparaison LOG est VRAIE.</p>

<sup>1</sup> Pour CONT et LOG : Cliquez sur le nom de l'instruction ("==" par exemple) pour modifier le type de comparaison dans la liste déroulante. Cliquez sur "???" et sélectionnez le type de données dans la liste déroulante.



Tableau 7- 28 Types de données pour les paramètres

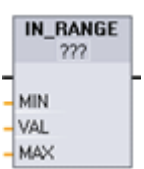
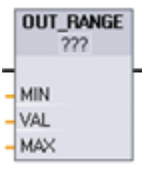
Paramètre	Type de données	Description
IN1, IN2	SInt, Int, DInt, USInt, UInt, UDIInt, Real, LReal, String, Char, Time, DTL, constante	Valeurs à comparer

Tableau 7- 29 Types de comparaison

Type de relation	La comparaison est vraie si ...
=	IN1 est égal à IN2
<>	IN1 est différent de IN2
>=	IN1 est supérieur ou égal à IN2
<=	IN1 est inférieur ou égal à IN2
>	IN1 est supérieur à IN2
<	IN1 est inférieur à IN2

## 7.4.2 Instructions Valeur dans la plage et Valeur en dehors de la plage

Tableau 7- 30 Instructions Valeur dans la plage et Valeur en dehors de la plage

CONT/LOG	SCL	Description
	<pre>out := IN_RANGE (min, val, max) ;</pre>	<p>Teste si une valeur d'entrée se situe ou non à l'intérieur d'une plage de valeurs indiquée.</p> <p>Lorsque la comparaison est vraie, la sortie de la boîte est VRAIE.</p>
	<pre>out := OUT_RANGE (min, val, max) ;</pre>	

<sup>1</sup> Pour CONT et LOG : Cliquez sur "???" et sélectionnez le type de données dans la liste déroulante.

Tableau 7- 31 Types de données pour les paramètres

Paramètre	Type de données <sup>1</sup>	Description
MIN, VAL, MAX	SInt, Int, DInt, USInt, UInt, UDIInt, Real, LReal, constante	Entrées du comparateur

<sup>1</sup> Les paramètres d'entrée MIN, VAL et MAX doivent avoir le même type de données.

- La comparaison IN\_RANGE est vraie si : MIN <= VAL <= MAX
- La comparaison OUT\_RANGE est vraie si : VAL < MIN ou VAL > MAX

### 7.4.3 Instructions Contrôler validité et Contrôler invalidité

Tableau 7- 32 Instructions Contrôler validité et Contrôler invalidité

CONT	LOG	SCL	Description
"IN" — OK —	"IN" OK	Non disponible	Teste si une référence de donnée d'entrée est un nombre réel valide selon la spécification IEEE 754.
"IN" — NOT_OK —	"IN" NOT_OK	Non disponible	

- <sup>1</sup> Pour CONT et LOG : Lorsque le résultat du test est vrai, le contact CONT est activé et transmet le flux de courant. Lorsque le résultat du test est vrai, la sortie de la boîte LOG est VRAIE.

Tableau 7- 33 Types de données pour le paramètre

Paramètre	Type de données	Description
IN	Real, LReal	Données d'entrée

Tableau 7- 34 Fonctionnement

Instruction	Le test concernant le nombre réel est vrai si :
OK	La valeur d'entrée est un nombre réel valide <sup>1</sup> .
NOT_OK	La valeur d'entrée n'est pas un nombre réel valide <sup>1</sup> .

- <sup>1</sup> Une valeur Real ou LReal est invalide s'il s'agit de +/- INF (l'infini), NaN (pas un nombre) ou d'une valeur dénormalisée. Une valeur dénormalisée est un nombre très proche de zéro. La CPU remplace la valeur dénormalisée par un zéro dans les calculs.

#### Voir aussi

EN et ENO pour CONT, LOG et SCL (Page 174)

## 7.5 Fonctions mathématiques

### 7.5.1 Instruction Calculer

Tableau 7- 35 Instruction CALCULATE

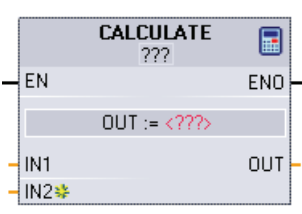
CONT/LOG	SCL	Description
	<p>Utilisez les expressions mathématiques SCL standard pour créer l'équation.</p>	<p>L'instruction CALCULATE permet de créer une fonction mathématique qui opère sur les entrées (IN1, IN2, .. INn) et fournit le résultat dans OUT, conformément à l'équation que vous définissez.</p> <ul style="list-style-type: none"> <li>Sélectionnez d'abord un type de données. Toutes les entrées et la sortie doivent avoir le même type de données.</li> <li>Pour ajouter une autre entrée, cliquez sur l'icône à côté de la dernière entrée.</li> </ul>

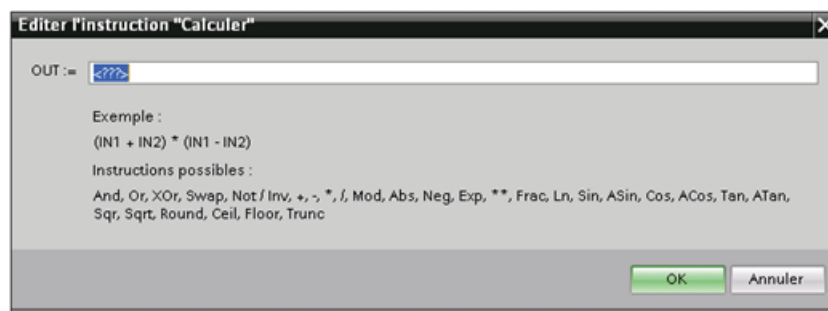
Tableau 7- 36 Types de données pour les paramètres

Paramètre	Type de données <sup>1</sup>
IN1, IN2, ..INn	SInt, Int, DInt, USInt, UInt, UDIInt, Real, LReal, Byte, Word, DWord
OUT	SInt, Int, DInt, USInt, UInt, UDIInt, Real, LReal, Byte, Word, DWord

<sup>1</sup> Les paramètres IN et OUT doivent avoir le même type de données (avec conversion implicite des paramètres d'entrée). Par exemple, une valeur SINT pour une entrée sera convertie en valeur INT ou REAL si OUT est un entier (INT) ou un réel (REAL).

Cliquez sur l'icône du calculateur pour ouvrir la boîte de dialogue et définir votre fonction mathématique. Vous entrez votre équation sous forme d'entrées (telles que IN1 et IN2) et d'opérations. Lorsque vous cliquez sur "OK" pour sauvegarder la fonction, le dialogue crée automatiquement les entrées pour l'instruction CALCULATE.

Un exemple et une liste des opérations mathématiques que vous pouvez inclure sont présentés au bas de l'éditeur.



**Remarque**

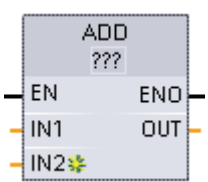
Vous devez également créer une entrée pour chaque constante dans votre fonction. La valeur constante sera alors entrée dans l'entrée associée pour l'instruction CALCULATE.

Le fait d'entrer des constantes sous forme d'entrées vous permet de copier l'instruction CALCULATE à d'autres endroits dans votre programme utilisateur sans avoir à changer la fonction. Vous pouvez alors modifier les valeurs ou les variables des entrées pour l'instruction sans modifier la fonction.

Lorsque CALCULATE est exécuté et que toutes les opérations individuelles dans le calcul s'achèvent avec succès, ENO est égal à 1. Sinon, ENO est égal à 0.

### 7.5.2 Instructions Addition, Soustraction, Multiplication et Division

Tableau 7- 37 Instructions Addition, Soustraction, Multiplication et Division

CONT/LOG	SCL	Description
	<pre> out := in1 + in2; out := in1 - in2; out := in1 * in2; out := in1 / in2;                     </pre>	<ul style="list-style-type: none"> <li>• ADD : Addition (IN1 + IN2 = OUT)</li> <li>• SUB : Soustraction (IN1 - IN2 = OUT)</li> <li>• MUL : Multiplication (IN1 * IN2 = OUT)</li> <li>• DIV : Division (IN1 / IN2 = OUT)</li> </ul> <p>Une division sur des entiers tronque la partie fractionnaire du quotient afin de produire une sortie entière.</p>

<sup>1</sup> Pour CONT et LOG : Cliquez sur "???" et sélectionnez un type de données dans le menu déroulant.

Tableau 7- 38 Types de données pour les paramètres (CONT et LOG)

Paramètre	Type de données <sup>1</sup>	Description
IN1, IN2	SInt, Int, DInt, USInt, UInt, UDIInt, Real, LReal, constante	Entrées de l'opération mathématique
OUT	SInt, Int, DInt, USInt, UInt, UDIInt, Real, LReal	Sortie de l'opération mathématique

<sup>1</sup> Les paramètres IN1, IN2 et OUT doivent avoir le même type de données.



Pour ajouter une entrée ADD ou MUL, cliquez sur l'icône de création ou cliquez avec le bouton droit de la souris sur la ligne d'entrée de l'un des paramètres IN existants et sélectionnez la commande "Insérer entrée".

Pour supprimer une entrée, cliquez avec le bouton droit de la souris sur la ligne d'entrée de l'un des paramètres IN existants (lorsqu'il y a plus d'entrées que les deux entrées d'origine) et sélectionnez la commande "Supprimer".

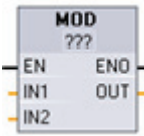
Lorsqu'elle est validée (EN = 1), l'instruction mathématique effectue l'opération spécifiée sur les valeurs d'entrée (IN1 et IN2) et mémorise le résultat dans l'adresse de mémoire indiquée par le paramètre de sortie (OUT). Une fois l'opération achevée avec succès, l'instruction met ENO à 1.

Tableau 7- 39 Etat de ENO

ENO	Description
1	Pas d'erreur
0	Le résultat de l'opération mathématique se situerait hors de la plage de nombres valide du type de données sélectionné. La partie la moins significative du résultat qui tient dans la taille de destination est renvoyée.
0	Division par 0 (IN2 = 0) : Le résultat est indéfini et zéro est renvoyé.
0	Real/LReal : Si l'une des valeurs d'entrée est NaN (pas un nombre), NaN est renvoyé.
0	ADD Real/LReal : Si les deux valeurs d'entrée sont l'infini avec des signes différents, il s'agit d'une opération interdite et NaN est renvoyé.
0	SUB Real/LReal : Si les deux valeurs d'entrée sont l'infini avec le même signe, il s'agit d'une opération interdite et NaN est renvoyé.
0	MUL Real/LReal : Si une valeur d'entrée est zéro et l'autre est l'infini, il s'agit d'une opération interdite et NaN est renvoyé.
0	DIV Real/LReal : Si les deux valeurs d'entrée sont zéro ou l'infini, il s'agit d'une opération interdite et NaN est renvoyé.

### 7.5.3 Instruction Modulo

Tableau 7- 40 Instruction MOD

CONT/LOG	SCL	Description
	<pre>out := in1 MOD in2;</pre>	<p>Vous pouvez utiliser l'instruction MOD pour renvoyer le reste d'une division entière. La valeur dans l'entrée IN1 est divisée par la valeur dans l'entrée IN2 et le reste est renvoyé dans la sortie OUT.</p>

<sup>1</sup> Pour CONT et LOG : Cliquez sur "???" et sélectionnez un type de données dans le menu déroulant.

Tableau 7- 41 Types de données pour les paramètres

Paramètre	Type de données <sup>1</sup>	Description
IN1 et IN2	SInt, Int, DInt, USInt, UInt, UDIInt, constante	Entrées de l'opération modulo
OUT	SInt, Int, DInt, USInt, UInt, UDIInt	Sortie de l'opération modulo

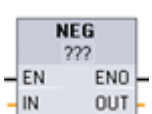
<sup>1</sup> Les paramètres IN1, IN2 et OUT doivent avoir le même type de données.

Tableau 7- 42 Valeurs de ENO

ENO	Description
1	Pas d'erreur
0	Valeur IN2 = 0 ; la valeur zéro est affectée à OUT.

### 7.5.4 Instruction Négation

Tableau 7- 43 Instruction NEG

CONT/LOG	SCL	Description
	<code>-(in) ;</code>	L'instruction NEG inverse le signe arithmétique de la valeur dans le paramètre IN et mémorise le résultat dans le paramètre OUT.

<sup>1</sup> Pour CONT et LOG : Cliquez sur "???" et sélectionnez un type de données dans le menu déroulant.

Tableau 7- 44 Types de données pour les paramètres

Paramètre	Type de données <sup>1</sup>	Description
IN	SInt, Int, DInt, Real, LReal, constante	Entrée de l'opération mathématique
OUT	SInt, Int, DInt, Real, LReal	Sortie de l'opération mathématique


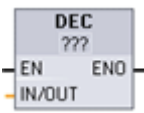
<sup>1</sup> Les paramètres IN et OUT doivent avoir le même type de données.

Tableau 7- 45 Etat de ENO

ENO	Description
1	Pas d'erreur
0	La valeur résultante se situe hors de la plage de nombres valide du type de données sélectionné. Exemple pour SInt : NEG (-128) donne comme résultat +128 qui dépasse le maximum du type de données.

## 7.5.5 Instructions Incrémenter et Décrémenter

Tableau 7- 46 Instructions INC et DEC

CONT/LOG	SCL	Description
	<code>in_out := in_out + 1;</code>	Incrémente une valeur entière signée ou non signée : valeur IN_OUT +1 = valeur IN_OUT
	<code>in_out := in_out - 1;</code>	Décrémente une valeur entière signée ou non signée : valeur IN_OUT - 1 = valeur IN_OUT

1 Pour CONT et LOG : Cliquez sur "???" et sélectionnez un type de données dans le menu déroulant.

Tableau 7- 47 Types de données pour les paramètres

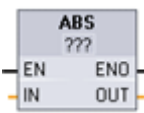
Paramètre	Type de données	Description
IN/OUT	SInt, Int, DInt, USInt, UInt, UDInt	Entrée et sortie de l'opération mathématique

Tableau 7- 48 Etat de ENO

ENO	Description
1	Pas d'erreur
0	La valeur résultante se situe hors de la plage de nombres valide du type de données sélectionné. Exemple pour SInt : INC (+127) donne comme résultat +128 qui dépasse le maximum du type de données.

## 7.5.6 Instruction Valeur absolue

Tableau 7- 49 Instruction ABS

CONT/LOG	SCL	Description
	<code>out := ABS(in);</code>	Calcule la valeur absolue d'un nombre réel ou entier signé dans le paramètre IN et mémorise le résultat dans le paramètre OUT.

1 Pour CONT et LOG : Cliquez sur "???" et sélectionnez un type de données dans le menu déroulant.

7.5 Fonctions mathématiques

Tableau 7- 50 Types de données pour les paramètres

Paramètre	Type de données <sup>1</sup>	Description
IN	SIInt, Int, DInt, Real, LReal	Entrée de l'opération mathématique
OUT	SIInt, Int, DInt, Real, LReal	Sortie de l'opération mathématique

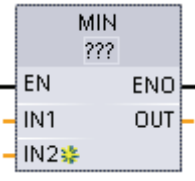
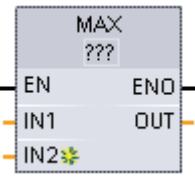
<sup>1</sup> Les paramètres IN et OUT doivent avoir le même type de données.

Tableau 7- 51 Etat de ENO

ENO	Description
1	Pas d'erreur
0	Le résultat de l'opération mathématique se situe hors de la plage de nombres valide du type de données sélectionné. Exemple pour SIInt : ABS (-128) donne comme résultat +128 qui dépasse le maximum du type de données.

7.5.7 Instructions Calculer le minimum et Calculer le maximum

Tableau 7- 52 Instructions MIN et MAX

CONT/LOG	SCL	Description
	<pre>out:= MIN(     in1:=_variant_in_,     in2:=_variant_in_     [...in32]);</pre>	L'instruction MIN compare la valeur de deux paramètres IN1 et IN2 et affecte la valeur minimale (inférieure) au paramètre OUT.
	<pre>out:= MAX(     in1:=_variant_in_,     in2:=_variant_in_     [...in32]);</pre>	L'instruction MAX compare la valeur de deux paramètres IN1 et IN2 et affecte la valeur maximale (supérieure) au paramètre OUT.

<sup>1</sup> Pour CONT et LOG : Cliquez sur "???" et sélectionnez un type de données dans le menu déroulant.

Tableau 7- 53 Types de données pour les paramètres

Paramètre	Type de données <sup>1</sup>	Description
IN1, IN2 [...IN32]	SIInt, Int, DInt, USInt, UInt, UDInt, Real, LReal, constante	Entrées de l'opération mathématique (jusqu'à 32 entrées)
OUT	SIInt, Int, DInt, USInt, UInt, UDInt, Real, LReal	Sortie de l'opération mathématique

<sup>1</sup> Les paramètres IN1, IN2 et OUT doivent avoir le même type de données.





Pour ajouter une entrée, cliquez sur l'icône de création ou cliquez avec le bouton droit de la souris sur la ligne d'entrée de l'un des paramètres IN existants et sélectionnez la commande "Insérer entrée".

Pour supprimer une entrée, cliquez avec le bouton droit de la souris sur la ligne d'entrée de l'un des paramètres IN existants (lorsqu'il y a plus d'entrées que les deux entrées d'origine) et sélectionnez la commande "Supprimer".

Tableau 7- 54 Etat de ENO

ENO	Description
1	Pas d'erreur
0	Pour le type de données Real uniquement : <ul style="list-style-type: none"> <li>• Une ou plusieurs entrées ne sont pas des nombres réels (NaN).</li> <li>• La valeur OUT résultante est +/- INF (l'infini).</li> </ul>

## 7.5.8 Instruction Définir une limite

Tableau 7- 55 Instruction LIMIT

CONT/LOG	SCL	Description
	<b>LIMIT</b> (MN:= <i>variant_in</i> , IN:= <i>variant_in</i> , MX:= <i>variant_in</i> , OUT:= <i>variant_out</i> ) ;	L'instruction Limit teste si la valeur du paramètre IN se situe à l'intérieur de la plage de valeurs indiquée par les paramètres MIN et MAX and if not, clamps the value at MIN or MAX. et, si ce n'est pas le cas, limite la valeur à MIN ou MAX.

<sup>1</sup> Pour CONT et LOG : Cliquez sur "???" et sélectionnez un type de données dans le menu déroulant.

Tableau 7- 56 Types de données pour les paramètres

Paramètre	Type de données <sup>1</sup>	Description
MN, INet MX	SInt, Int, DInt, USInt, UInt, UDInt, Real, LReal, constante	Entrées de l'opération mathématique
OUT	SInt, Int, DInt, USInt, UInt, UDInt, Real, LReal	Sortie de l'opération mathématique

<sup>1</sup> Les paramètres MN, IN, MXet OUTdoivent avoir le même type de données.

Si la valeur du paramètre IN se situe à l'intérieur de la plage indiquée, la valeur de IN est mémorisée dans le paramètre OUT. Si la valeur du paramètre IN se situe hors de la plage indiquée, OUT prend la valeur du paramètre MIN (si la valeur de IN est inférieure à la valeur MIN) ou la valeur du paramètre MAX (si la valeur de IN est supérieure à la valeur MAX).

Tableau 7- 57 Etat de ENO

ENO	Description
1	Pas d'erreur
0	Real : Si une ou plusieurs des valeurs de MIN, IN et MAX sont NaN (pas des nombres), NaN est renvoyé.
0	Si MIN est supérieur à MAX, la valeur IN est affectée à OUT.

Exemples SCL :

- MyVal := LIMIT(MN:=10,IN:=53, MX:=40); //Résultat : MyVal = 40
- MyVal := LIMIT(MN:=10,IN:=37, MX:=40); //Résultat : MyVal = 37
- MyVal := LIMIT(MN:=10,IN:=8, MX:=40); //Résultat : MyVal = 10

### 7.5.9 Instructions mathématiques à virgule flottante



Vous utilisez les instructions à virgule flottante pour programmer des opérations mathématiques avec le type de données Real ou LReal :

- SQR : Carré ( $IN^2 = OUT$ )
- SQRT : Racine carrée ( $\sqrt{IN} = OUT$ )
- LN : Logarithme népérien ( $LN(IN) = OUT$ )
- EXP : Fonction exponentielle ( $e^{IN} = OUT$ ), avec base  $e = 2,71828182845904523536$
- EXPT : Elever à la puissance ( $IN1^{IN2} = OUT$ )

Les paramètres IN1 et OUT de EXPT ont toujours le même type de données, à savoir Real ou LReal. Vous pouvez sélectionner le type de données pour l'exposant IN2 parmi de nombreux types de données.

- FRAC : Calculer le nombre de décimales (partie fractionnaire du nombre à virgule flottante  $IN = OUT$ )
- SIN : Sinus ( $\sin(IN \text{ radians}) = OUT$ )  
ASIN : Arc sinus ( $\arcsin(IN) = OUT \text{ radians}$ ), avec  $\sin(OUT \text{ radians}) = IN$
- COS : Cosinus ( $\cos(IN \text{ radians}) = OUT$ )  
ACOS : Arc cosinus ( $\arccos(IN) = OUT \text{ radians}$ ), avec  $\cos(OUT \text{ radians}) = IN$
- TAN : Tangente ( $\tan(IN \text{ radians}) = OUT$ )  
ATAN : Arc tangente ( $\arctan(IN) = OUT \text{ radians}$ ), avec  $\tan(OUT \text{ radians}) = IN$

Tableau 7- 58 Exemples d'instructions mathématiques à virgule flottante

CONT/LOG	SCL	Description
	<pre>out := SQR(in);</pre> <p>ou</p> <pre>out := in * in;</pre>	<p>Carré : <math>IN^2 = OUT</math></p> <p>Par exemple : Si IN est égal à 9, alors OUT est égal à 81.</p>
	<pre>out := in1 ** in2;</pre>	<p>Elever à la puissance : <math>IN1^{IN2} = OUT</math></p> <p>Par exemple : Si IN1 est égal à 3 et IN2 est égal à 2, alors OUT est égal à 9.</p>

- 1 Pour CONT et LOG : Cliquez sur "???" (près du nom de l'instruction) et sélectionnez un type de données dans le menu déroulant.
- 2 Pour SCL : Vous pouvez aussi utiliser les opérateurs mathématiques SCL de base pour créer les expressions mathématiques

Tableau 7- 59 Types de données pour les paramètres

Paramètre	Type de données	Description
IN, IN1	Real, LReal, constante	Entrées
IN2	SInt, Int, DInt, USInt, UInt, UDInt, Real, LReal, constante	Entrée exposant EXPT
OUT	Real, LReal	Sorties

Tableau 7- 60 Etat de ENO

ENO	Instruction	Situation d'erreur	Résultat (OUT)
1	Toutes	Pas d'erreur	Résultat valide
0	SQR	Le résultat dépasse la plage Real/LReal valide.	+INF
		IN est +/- NaN (pas un nombre)	+NaN
	SQRT	IN est négatif.	-NaN
		IN est +/- INF (l'infini) ou +/- NaN.	+/- INF ou +/- NaN
	LN	IN est 0.0, négatif, -INF ou -NaN.	-NaN
		IN est +INF ou +NaN.	+INF ou +NaN
	EXP	Le résultat dépasse la plage Real/LReal valide.	+INF
		IN est +/- NaN.	+/- NaN
	SIN, COS, TAN	IN est +/- INF ou +/- NaN.	+/- INF ou +/- NaN
	ASIN, ACOS	IN se situe en dehors de la plage valide de -1.0 à +1.0.	+NaN
		IN est +/- NaN.	+/- NaN
	ATAN	IN est +/- NaN.	+/- NaN
	FRAC	IN est +/- INF ou +/- NaN.	+NaN
EXPT	IN1 est +INF et IN2 n'est pas -INF.	+INF	

7.6 Transfert

ENO	Instruction	Situation d'erreur	Résultat (OUT)
		IN1 est négatif ou -INF.	+NaN si IN2 est Real/LReal, -INF sinon
		IN1 ou IN2 est +/- NaN.	+NaN
		IN1 est 0.0 et IN2 est Real/LReal (seulement).	+NaN

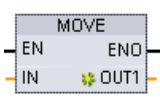
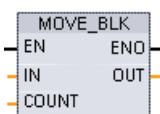
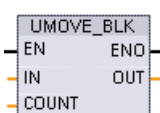
## 7.6 Transfert

### 7.6.1 Instructions Copier valeur et Copier zone

Utilisez les instructions de transfert pour copier des éléments de données à une nouvelle adresse mémoire et les convertir en un autre type de données. Les données source ne sont pas modifiées par le processus de transfert.

- L'instruction MOVE copie un élément de données unique de l'adresse source indiquée par le paramètre IN dans l'adresse de destination indiquée par le paramètre OUT.
- Les instructions MOVE\_BLK et UMOVE\_BLK disposent d'un paramètre COUNT supplémentaire. COUNT précise combien d'éléments de données doivent être copiés. Le nombre d'octets par élément copié dépend du type de données affecté aux noms de variable des paramètres IN et OUT dans la table des variables API.

Tableau 7- 61 Instructions MOVE, MOVE\_BLK et UMOVE\_BLK

CONT/LOG	SCL	Description
	<pre>out1 := in;</pre>	Copie un élément de données stocké à l'adresse indiquée dans une ou plusieurs nouvelles adresses. <sup>1</sup>
	<pre>MOVE_BLK (     in:=_variant_in,     count:=_uint_in,     out=&gt;_variant_out);</pre>	Copie de zone interruptible qui copie une zone d'éléments de données à une nouvelle adresse.
	<pre>UMOVE_BLK (     in:=_variant_in,     count:=_uint_in,     out=&gt;_variant_out);</pre>	Copie de zone contiguë qui copie une zone d'éléments de données à une nouvelle adresse.

<sup>1</sup> Instruction MOVE : Pour ajouter une autre sortie en CONT ou LOG, cliquez sur l'icône de création à côté du paramètre de sortie. Pour SCL, utilisez des instructions d'affectation multiple. Vous pouvez également utiliser l'une des structures de boucle.

Tableau 7- 62 Types de données pour l'instruction MOVE

Paramètre	Type de données	Description
IN	SInt, Int, DInt, USInt, UInt, UDInt, Real, LReal, Byte, Word, DWord, Char, Array, Struct, DTL, Time	Adresse de source
OUT	SInt, Int, DInt, USInt, UInt, UDInt, Real, LReal, Byte, Word, DWord, Char, Array, Struct, DTL, Time	Adresse de destination



Pour ajouter des sorties MOVE, cliquez sur l'icône de création ou cliquez avec le bouton droit de la souris sur la ligne de sortie de l'un des paramètres OUT existants et sélectionnez la commande "Insérer sortie".

Pour supprimer une sortie, cliquez avec le bouton droit de la souris sur la ligne de sortie de l'un des paramètres OUT existants (lorsqu'il y a plus de sorties que les deux sorties d'origine) et sélectionnez la commande "Supprimer".

Tableau 7- 63 Types de données pour les instructions MOVE\_BLK et UMOVE\_BLK

Paramètre	Type de données	Description
IN	SInt, Int, DInt, USInt, UInt, UDInt, Real, LReal, Byte, Word, DWord	Adresse de début de la source
COUNT	UInt	Nombre d'éléments de données à copier
OUT	SInt, Int, DInt, USInt, UInt, UDInt, Real, LReal, Byte, Word, DWord	Adresse de début de la destination

### Remarque

#### Règles pour les opérations de copie de données

- Pour copier le type de données Bool, utilisez SET\_BF, RESET\_BF, R, S ou la bobine de sortie (CONT) (Page 188).
- Pour copier un seul type de données simple, utilisez MOVE.
- Pour copier un tableau de type de données simple, utilisez MOVE\_BLK ou UMOVE\_BLK.
- Pour copier une structure, utilisez MOVE.
- Pour copier une chaîne, utilisez S\_MOVE (Page 271).
- Pour copier un seul caractère dans une chaîne, utilisez MOVE.
- Vous ne pouvez pas utiliser les instructions MOVE\_BLK et UMOVE\_BLK pour copier des tableaux ou des structures dans les zones de mémoire I, Q ou M.

Les instructions MOVE\_BLK et UMOVE\_BLK diffèrent dans la façon dont les alarmes sont traitées :

- Les événements d'alarme sont **mis en file d'attente et traités** pendant l'exécution de MOVE\_BLK. Utilisez l'instruction MOVE\_BLK lorsque les données à l'adresse de destination de la copie ne sont pas utilisées dans un sous-programme d'OB d'alarme ou lorsque les données de destination n'ont pas besoin d'être cohérentes si elles sont utilisées. Si le fonctionnement de MOVE\_BLK est interrompu, le dernier élément de données copié est complet et cohérent à l'adresse de destination. Le fonctionnement de MOVE\_BLK reprend une fois l'OB d'alarme exécuté.
- Les événements d'alarme sont **mis en file d'attente mais ne sont pas traités** tant que l'exécution de UMOVE\_BLK n'est pas achevée. Utilisez l'instruction UMOVE\_BLK lorsque l'opération de copie doit être achevée et que les données de destination doivent être cohérentes avant l'exécution d'un sous-programme d'OB d'alarme. Pour plus d'informations, reportez-vous au paragraphe sur la cohérence des données (Page 163).

ENO est toujours vrai après l'exécution de l'instruction MOVE.

Tableau 7- 64 Etat de ENO

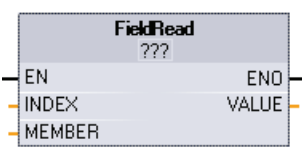

ENO	Situation d'erreur	Résultat
1	Pas d'erreur	Tous les COUNT éléments ont été copiés avec succès.
0	La plage source (IN) ou la plage de destination (OUT) dépasse la zone de mémoire disponible.	Les éléments qui entrent sont copiés. Il n'y a pas copie d'éléments partiels.

## 7.6.2 Instructions FieldRead et FieldWrite

### Remarque

STEP 7 V10.5 **n'acceptait pas** de variable comme indice de tableau, ni les tableaux multidimensionnels. Les instructions FieldRead et FieldWrite rendaient possible l'utilisation d'un indice variable pour les tableaux à une dimension. STEP 7 V11 **prend en charge** les variables comme indices de tableau et les tableaux multidimensionnels. Les instructions FieldRead et FieldWrite sont incluses dans STEP 7 V11 pour assurer la compatibilité amont avec les programmes qui utilisaient ces instructions.

Tableau 7- 65 Instructions FieldRead et FieldWrite

CONT/LOG	SCL	Description
	<pre>value := member[index];</pre>	L'instruction FieldRead lit l'élément de valeur d'indice INDEX dans le tableau dont le premier élément est indiqué par le paramètre MEMBER. La valeur de l'élément de tableau est transférée à l'adresse indiquée par le paramètre VALUE.
	<pre>member[index] := value;</pre>	L'instruction FieldWrite transfère la valeur à l'adresse indiquée par le paramètre VALUE dans le tableau dont le premier élément est indiqué par le paramètre MEMBER. La valeur est transférée dans l'élément de tableau dont l'indice est indiqué par le paramètre INDEX.

<sup>1</sup> Pour CONT et LOG : Cliquez sur "???" et sélectionnez un type de données dans le menu déroulant.

Tableau 7- 66 Types de données pour les paramètres

Paramètre et type		Type de données	Description
Index	Entrée	DInt	Numéro d'indice de l'élément de tableau à lire ou dans lequel écrire
Member <sup>1</sup>	Entrée	Types d'élément de tableau : Bool, Byte, Word, DWord, Char, SInt, Int, Dint, USInt, UInt, UDInt, Real, LReal	Emplacement du premier élément dans un tableau à une dimension défini dans un bloc de données global ou une interface de bloc. Par exemple : Si l'indice du tableau est donné sous la forme [-2..4], l'indice du premier élément est -2 et non 0.
Value <sup>1</sup>	Sortie	Bool, Byte, Word, DWord, Char, SInt, Int, Dint, USInt, UInt, UDInt, Real, LReal	Pour FieldRead, emplacement dans lequel l'élément de tableau indiqué est copié. Pour FieldWrite, emplacement de la valeur qui est copiée dans l'élément de tableau indiqué.

<sup>1</sup> Le type de données de l'élément de tableau indiqué par le paramètre MEMBER et celui du paramètre VALUE doivent être identiques.

La sortie de validation ENO est égale à 0 si l'une des conditions suivantes est vraie :

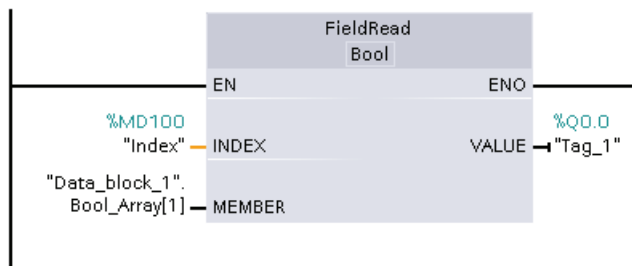
- L'entrée EN a l'état de signal "0".
- L'élément de tableau indiqué par le paramètre INDEX n'est pas défini dans le tableau désigné par le paramètre MEMBER.
- Des erreurs, telles qu'un débordement, se sont produites pendant le traitement.

### Accès à des données par indexation de tableau

Pour accéder aux éléments d'un tableau au moyen d'une variable, il suffit d'utiliser la variable comme indice de tableau dans la logique du programme. Le réseau ci-dessous, par exemple, définit une sortie en fonction de la valeur booléenne, désignée par la variable API "Index", d'un tableau de booléens dans "Data\_block\_1".



La logique avec l'indice de tableau variable est équivalente à l'ancienne méthode utilisant l'instruction FieldRead :



Les instructions FieldWrite et FieldRead peuvent être remplacées par la logique d'indexation de tableau par une variable.

SCL ne dispose pas d'instructions FieldRead ou FieldWrite, mais prend en charge l'adressage indirect d'un tableau avec une variable :

```
#Tag_1 := "Data_block_1".Bool_Array[#Index];
```

### 7.6.3 Instructions Compléter

Tableau 7- 67 Instructions FILL\_BLK et UFILL\_BLK

CONT/LOG	SCL	Description
	<pre>FILL_BLK (     in:=_variant_in,     count:=int,     out=&gt;_variant_out);</pre>	L'instruction Compléter zone remplit une zone d'adresses avec des copies d'un élément de données indiqué.
	<pre>UFILL_BLK (     in:=_variant_in,     count:=int     out=&gt;_variant_out);</pre>	L'instruction Compléter zone contiguë remplit une zone d'adresses avec des copies d'un élément de données indiqué.



Tableau 7- 68 Types de données pour les paramètres

Paramètre	Type de données	Description
IN	SInt, Int, DIntT, USInt, UInt, UDIInt, Real, LReal, Byte, Word, DWord	Adresse source des données
COUNT	USInt, UInt	Nombre d'éléments de données à copier
OUT	SInt, Int, DIntT, USInt, UInt, UDIInt, Real, LReal, Byte, Word, DWord	Adresse de destination des données

### Remarque

#### Règles pour les opérations de remplissage de zone

- Pour compléter avec le type de données BOOL, utilisez SET\_BF, RESET\_BF, R, S ou la bobine de sortie (CONT).
- Pour compléter avec un seul type de données simple, utilisez MOVE.
- Pour compléter un tableau avec un type de données simple, utilisez FILL\_BLK ou UFILL\_BLK.
- Pour compléter un seul caractère dans une chaîne, utilisez MOVE.
- Vous ne pouvez pas utiliser les instructions FILL\_BLK et UFILL\_BLK pour compléter des tableaux dans les zones de mémoire I, Q ou M.

Les instructions FILL\_BLK et UFILL\_BLK copient l'élément de données source IN dans la destination dont l'adresse initiale est indiquée par le paramètre OUT. Le processus de copie est répété de manière à compléter un bloc d'adresses adjacentes jusqu'à ce que le nombre de copies soit égal au paramètre COUNT.

Les instructions FILL\_BLK et UFILL\_BLK diffèrent dans la façon dont les alarmes sont traitées :

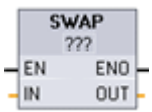
- Les événements d'alarme sont **mis en file d'attente et traités** pendant l'exécution de FILL\_BLK. Utilisez l'instruction FILL\_BLK lorsque les données à l'adresse de destination de la copie ne sont pas utilisées dans un sous-programme d'OB d'alarme ou lorsque les données de destination n'ont pas besoin d'être cohérentes si elles sont utilisées.
- Les événements d'alarme sont **mis en file d'attente mais ne sont pas traités** tant que l'exécution de UFILL\_BLK n'est pas achevée. Utilisez l'instruction UFILL\_BLK lorsque l'opération de copie doit être achevée et que les données de destination doivent être cohérentes avant l'exécution d'un sous-programme d'OB d'alarme.

Tableau 7- 69 Etat de ENO

ENO	Situation d'erreur	Résultat
1	Pas d'erreur	L'élément IN a été copié avec succès dans les COUNT destinations.
0	La plage de destination (OUT) dépasse la zone de mémoire disponible.	Les éléments qui entrent sont copiés. Il n'y a pas copie d'éléments partiels.

### 7.6.4 Instruction Permutation

Tableau 7- 70 Instruction SWAP

CONT/LOG	SCL	Description
	<pre>out := SWAP(in);</pre>	Inverse l'ordre des octets dans des éléments de données à deux et quatre octets. L'ordre des bits à l'intérieur de chaque octet n'est pas modifié. ENO est toujours VRAI après l'exécution de l'instruction SWAP.

<sup>1</sup> Pour CONT et LOG : Cliquez sur "???" et sélectionnez un type de données dans le menu déroulant.

Tableau 7- 71 Types de données pour les paramètres

Paramètre	Type de données	Description
IN	Word, DWord	Octets de données dans l'ordre
OUT	Word, DWord	Octets de données dans l'ordre inverse

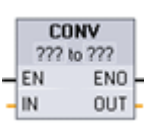
Exemple 1	Paramètre IN = MB0 (avant exécution)		Paramètre OUT = MB4 (après exécution)	
Adresse	MB0	MB1	MB4	MB5
W#16#1234	12	34	34	12
WORD	OPFo	OPFa	OPFo	OPFa

Exemple 2	Paramètre IN = MB0 (avant exécution)				Paramètre OUT = MB4 (après exécution)			
Adresse	MB0	MB1	MB2	MB3	MB4	MB5	MB6	MB7
DW#16# 12345678	12	34	56	78	78	56	34	12
DWORD	OPFo			OPFa	OPFo			OPFa

## 7.7 Convertir

### 7.7.1 Instruction CONV

Tableau 7- 72 Instruction Convertir (CONV)

CONT/LOG	SCL	Description
	<pre>out := &lt;data type in&gt;_TO_&lt;data type out&gt;(in);</pre>	Convertit un élément de données d'un type de données dans un autre type de données.

- 1 Pour CONT et LOG : Cliquez sur "???" et sélectionnez les types de données dans le menu déroulant.
- 2 Pour SCL : Construisez l'instruction de conversion en identifiant le type de données pour le paramètre d'entrée et le paramètre de sortie. Par exemple, DWORD\_TO\_REAL convertit une valeur double mot en une valeur réelle.

Tableau 7- 73 Types de données pour les paramètres

Paramètre	Type de données	Description
IN	Chaîne de bits <sup>1</sup> , SInt, USInt, Int, UInt, DInt, UDInt, Real, LReal, BCD16, BCD32	Valeur d'entrée
OUT	Chaîne de bits <sup>1</sup> , SInt, USInt, Int, UInt, DInt, UDInt, Real, LReal, BCD16, BCE32	Valeur d'entrée convertie au nouveau type de données

- 1 L'instruction ne vous permet pas de sélectionner des chaînes de bits (octet, mot, double mot). Pour entrer un opérande de type Byte, Word ou DWord comme paramètre de l'instruction, sélectionnez un entier non signé ayant la même longueur en bits. Sélectionnez, par exemple, USInt pour un octet, UInt pour un mot ou UDInt pour un double mot.

Une fois que vous avez sélectionné le type de données (convertir de), une liste des conversions possibles s'affiche dans la liste déroulante (convertir en). Les conversions de et en BCD16 sont limitées au type de données Int. Les conversions de et en BCD32 sont limitées au type de données DInt.

Tableau 7- 74 Etat de ENO

ENO	Description	Résultat OUT
1	Pas d'erreur	Résultat valide
0	IN est +/- INF ou +/- NaN.	+/- INF ou +/- NaN
0	Le résultat dépasse la plage valide pour le type de données OUT.	OUT prend la valeur des octets de poids faible de IN

## 7.7.2 Instructions de conversion pour SCL

### Instructions de conversion pour SCL

Tableau 7- 75 Conversion à partir de Bool, Byte, Word ou DWord

Type de données	Instruction	Résultat
Bool	BOOL_TO_BYTE, BOOL_TO_WORD, BOOL_TO_DWORD, BOOL_TO_INT, BOOL_TO_DINT	La valeur est transférée dans le bit de poids faible du type de données cible.
Byte	BYTE_TO_BOOL	Le bit de poids faible est transféré dans le type de données cible.
	BYTE_TO_WORD, BYTE_TO_DWORD	La valeur est transférée dans l'octet de poids faible du type de données cible.
	BYTE_TO_SINT, BYTE_TO_USINT	La valeur est transférée dans le type de données cible.
	BYTE_TO_INT, BYTE_TO_UINT, BYTE_TO_DINT, BYTE_TO_UDINT	La valeur est transférée dans l'octet de poids faible du type de données cible.
Word	WORD_TO_BOOL	Le bit de poids faible est transféré dans le type de données cible.
	WORD_TO_BYTE	L'octet de poids faible de la valeur source est transféré dans le type de données cible.
	WORD_TO_DWORD	La valeur est transférée dans l'octet de poids faible du type de données cible.
	WORD_TO_SINT, WORD_TO_USINT	L'octet de poids faible de la valeur source est transféré dans le type de données cible.
	WORD_TO_INT, WORD_TO_UINT	La valeur est transférée dans le type de données cible.
	WORD_TO_DINT, WORD_TO_UDINT	La valeur est transférée dans l'octet de poids faible du type de données cible.
DWord	DWORD_TO_BOOL	Le bit de poids faible est transféré dans le type de données cible.
	DWORD_TO_BYTE, DWORD_TO_WORD, DWORD_TO_SINT, DWORD_TO_USINT, DWORD_TO_INT, DWORD_TO_UINT	L'octet de poids faible de la valeur source est transféré dans le type de données cible.
	DWORD_TO_DINT, DWORD_TO_UDINT, DWORD_TO_REAL	La valeur est transférée dans le type de données cible.

Tableau 7- 76 Conversion à partir d'un entier court (SInt ou USInt)

Type de données	Instruction	Résultat
SInt	SINT_TO_BOOL	Le bit de poids faible est transféré dans le type de données cible.
	SINT_TO_BYTE	La valeur est transférée dans le type de données cible.
	SINT_TO_WORD, SINT_TO_DWORD, SINT_TO_INT, SINT_TO_DINT	La valeur est transférée dans l'octet de poids faible du type de données cible.

Type de données	Instruction	Résultat
	SINT_TO_USINT, SINT_TO_UINT, SINT_TO_UDINT, SINT_TO_REAL, SINT_TO_LREAL, SINT_TO_CHAR, SINT_TO_STRING	La valeur est convertie.
USInt	USINT_TO_BOOL	Le bit de poids faible est transféré dans le type de données cible.
	USINT_TO_BYTE	La valeur est transférée dans le type de données cible.
	USINT_TO_WORD, USINT_TO_DWORD, USINT_TO_INT, USINT_TO_UINT, USINT_TO_DINT, USINT_TO_UDINT	La valeur est transférée dans l'octet de poids faible du type de données cible.
	USINT_TO_SINT, USINT_TO_REAL, USINT_TO_LREAL, USINT_TO_CHAR, USINT_TO_STRING	La valeur est convertie.

Tableau 7- 77 Conversion à partir d'un entier (Int ou UInt)

Type de données	Instruction	Résultat
Int	INT_TO_BOOL	Le bit de poids faible est transféré dans le type de données cible.
	INT_TO_BYTE, INT_TO_DWORD, INT_TO_SINT, INT_TO_USINT, INT_TO_UINT, INT_TO_UDINT, INT_TO_REAL, INT_TO_LREAL, INT_TO_CHAR, INT_TO_STRING	La valeur est convertie.
	INT_TO_WORD	La valeur est transférée dans le type de données cible.
	INT_TO_DINT	La valeur est transférée dans l'octet de poids faible du type de données cible.
UInt	UINT_TO_BOOL	Le bit de poids faible est transféré dans le type de données cible.
	UINT_TO_BYTE, UINT_TO_SINT, UINT_TO_USINT, UINT_TO_INT, UINT_TO_REAL, UINT_TO_LREAL, UINT_TO_CHAR, UINT_TO_STRING	La valeur est convertie.
	UINT_TO_WORD, UINT_TO_DATE	La valeur est transférée dans le type de données cible.
	UINT_TO_DWORD, UINT_TO_DINT, UINT_TO_UDINT	La valeur est transférée dans l'octet de poids faible du type de données cible.

7.7 Convertir

Tableau 7- 78 Conversion à partir d'un entier double (Dint ou UDInt)

Type de données	Instruction	Résultat
Dint	DINT_TO_BOOL	Le bit de poids faible est transféré dans le type de données cible.
	DINT_TO_BYTE, DINT_TO_WORD, DINT_TO_SINT, DINT_TO_USINT, DINT_TO_INT, DINT_TO_UINT, DINT_TO_UDINT, DINT_TO_REAL, DINT_TO_LREAL, DINT_TO_CHAR, DINT_TO_STRING	La valeur est convertie.
	DINT_TO_DWORD, DINT_TO_TIME	La valeur est transférée dans le type de données cible.
UDInt	UDINT_TO_BOOL	Le bit de poids faible est transféré dans le type de données cible.
	UDINT_TO_BYTE, UDINT_TO_WORD, UDINT_TO_SINT, UDINT_TO_USINT, UDINT_TO_INT, UDINT_TO_UINT, UDINT_TO_DINT, UDINT_TO_REAL, UDINT_TO_LREAL, UDINT_TO_CHAR, UDINT_TO_STRING	La valeur est convertie.
	UDINT_TO_DWORD, UDINT_TO_TOD	La valeur est transférée dans le type de données cible.

Tableau 7- 79 Conversion à partir d'un réel (Real ou LReal)

Type de données	Instruction	Résultat
Real	REAL_TO_DWORD, REAL_TO_LREAL	La valeur est transférée dans le type de données cible.
	REAL_TO_SINT, REAL_TO_USINT, REAL_TO_INT, REAL_TO_UINT, REAL_TO_DINT, REAL_TO_UDINT, REAL_TO_STRING	La valeur est convertie.
LReal	LREAL_TO_SINT, LREAL_TO_USINT, LREAL_TO_INT, LREAL_TO_UINT, LREAL_TO_DINT, LREAL_TO_UDINT, LREAL_TO_REAL, LREAL_TO_STRING	La valeur est convertie.

Tableau 7- 80 Conversion à partir de Time, DTL, TOD ou Date



Type de données	Instruction	Résultat
Time	TIME_TO_DINT	La valeur est transférée dans le type de données cible.
DTL	DTL_TO_DATE, DTL_TO_TOD	La valeur est convertie.
TOD	TOD_TO_UDINT	La valeur est convertie.
Date	DATE_TO_UINT	La valeur est convertie.

Tableau 7- 81 Conversion à partir de Char ou String

Type de données	Instruction	Résultat
Char	CHAR_TO_SINT, CHAR_TO_USINT, CHAR_TO_INT, CHAR_TO_UINT, CHAR_TO_DINT, CHAR_TO_UDINT	La valeur est convertie.
	CHAR_TO_STRING	La valeur est transférée dans le premier caractère de la chaîne.
String	STRING_TO_SINT, STRING_TO_USINT, STRING_TO_INT, STRING_TO_UINT, STRING_TO_DINT, STRING_TO_UDINT, STRING_TO_REAL, STRING_TO_LREAL	La valeur est convertie.
	STRING_TO_CHAR	Le premier caractère de la chaîne est copié dans Char.

### 7.7.3 Instructions Arrondir et Former un nombre entier

Tableau 7- 82 Instructions ROUND et TRUNC

CONT/LOG	SCL	Description
	<pre>out := ROUND (in);</pre>	<p>Convertit un nombre réel en entier. Le type de données par défaut est DINT. Lorsque la sortie est un type de données valide autre que DINT, il faut la déclarer explicitement, par exemple ROUND_REAL ou ROUND_LREAL.</p> <p>La partie fractionnaire du nombre réel est arrondie à la valeur entière la plus proche (IEEE, arrondir au plus proche). Si le nombre se situe exactement à mi-chemin entre deux entiers (par exemple 10,5), le nombre est arrondi à l'entier pair. Par exemple :</p> <ul style="list-style-type: none"> <li>• ROUND (10.5) = 10</li> <li>• ROUND (11.5) = 12</li> </ul>
	<pre>out := TRUNC (in);</pre>	<p>TRUNC convertit un nombre réel en entier. La partie fractionnaire du nombre réel est tronquée à zéro (IEEE, arrondir à zéro).</p>

<sup>1</sup> Pour CONT et LOG : Cliquez sur "???" (près du nom de l'instruction) et sélectionnez un type de données dans le menu déroulant.

Tableau 7- 83 Types de données pour les paramètres

Paramètre	Type de données	Description
IN	Real, LReal	Entrée en virgule flottante
OUT	SInt, Int, DInt, USInt, UInt, UDIInt, Real, LReal	Sortie arrondie ou tronquée



7.7 Convertir

Tableau 7- 84 Etat de ENO

ENO	Description	Résultat OUT
1	Pas d'erreur	Résultat valide
0	IN est +/- INF ou +/- NaN.	+/- INF ou +/- NaN

7.7.4 Instructions Arrondir à l'entier supérieur et Arrondir à l'entier inférieur

Tableau 7- 85 Instructions CEIL et FLOOR

CONT/LOG	SCL	Description
	<code>out := CEIL(in);</code>	Convertit un nombre réel (Real ou LReal) au nombre entier le plus proche supérieur ou égal au nombre réel sélectionné ("arrondir à +infini" IEEE).
	<code>out := FLOOR(in);</code>	Convertit un nombre réel (Real ou LReal) au nombre entier le plus proche inférieur ou égal au nombre réel sélectionné ("arrondir à -infini" IEEE).

<sup>1</sup> Pour CONT et LOG : Cliquez sur "???" (près du nom de l'instruction) et sélectionnez un type de données dans le menu déroulant.

Tableau 7- 86 Types de données pour les paramètres

Paramètre	Type de données	Description
IN	Real, LReal	Entrée en virgule flottante
OUT	SInt, Int, DInt, USInt, UInt, UDIInt, Real, LReal	Sortie convertie


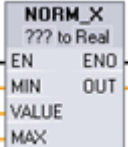
Tableau 7- 87 Etat de ENO

ENO	Description	Résultat OUT
1	Pas d'erreur	Résultat valide
0	IN est +/- INF ou +/- NaN.	+/- INF ou +/- NaN



## 7.7.5 Instructions Mise à l'échelle et Normaliser

Tableau 7- 88 Instructions SCALE\_X et NORM\_X

CONT/LOG	SCL	Description
	<pre>out :=SCALE_X(min:=_in_,                value:=_in_,                max:=_in_);</pre>	<p>Met à l'échelle le paramètre réel normalisé VALUE, avec <math>0.0 \leq \text{VALUE} \leq 1.0</math>, dans le type de données et la plage de valeurs indiqués par les paramètres MIN et MAX :</p> $\text{OUT} = \text{VALUE} (\text{MAX} - \text{MIN}) + \text{MIN}$
	<pre>out :=NORM_X(min:=_in_,               value:=_in_,               max:=_in_);</pre>	<p>Normalise le paramètre VALUE à l'intérieur de la plage de valeurs indiquée par les paramètres MIN et MAX :</p> $\text{OUT} = (\text{VALUE} - \text{MIN}) / (\text{MAX} - \text{MIN}),$ <p>avec <math>0.0 \leq \text{OUT} \leq 1.0</math></p>

<sup>1</sup> Pour CONT et LOG : Cliquez sur "???" et sélectionnez un type de données dans le menu déroulant.

Tableau 7- 89 Types de données pour les paramètres

Paramètre	Type de données <sup>1</sup>	Description
MIN	SInt, Int, DInt, USInt, UInt, UDInt, Real, LReal	Valeur d'entrée pour le minimum de la plage
VALUE	SCALE_X: Real, LReal NORM_X: SInt, Int, DInt, USInt, UInt, UDInt, Real, LReal	Valeur d'entrée à mettre à l'échelle ou à normaliser
MAX	SInt, Int, DInt, USInt, UInt, UDInt, Real, LReal	Valeur d'entrée pour le maximum de la plage
OUT	SCALE_X: SInt, Int, DInt, USInt, UInt, UDInt, Real, LReal NORM_X: Real, LReal	Valeur de sortie mise à l'échelle ou normalisée

<sup>1</sup> Pour SCALE\_X: Les paramètres MIN, MAX et OUT doivent avoir le même type de données.  
Pour NORM\_X: Les paramètres MIN, VALUE et MAX doivent avoir le même type de données.

**Remarque**

**Le paramètre VALUE de SCALE\_X doit être limité à  $0.0 \leq VALUE \leq 1.0$**

Si le paramètre VALUE est inférieur à 0.0 ou supérieur à 1.0 :

- La mise à l'échelle linéaire peut produire des valeurs OUT qui sont inférieures à la valeur du paramètre MIN ou supérieures à la valeur du paramètre MAX pour des valeurs OUT qui se situent dans la plage de valeurs du type de données OUT. Dans ces cas, l'exécution de SCALE\_X met ENO à VRAI.
- Il est possible de générer des nombres mis à l'échelle qui ne sont pas dans la plage du type de données OUT. Dans ces cas, la valeur du paramètre OUT est définie à une valeur intermédiaire égale à la partie la moins significative du nombre réel mis à l'échelle avant la conversion finale au type de données OUT. Dans ce cas, l'exécution de SCALE\_X met ENO à FAUX.

**Le paramètre VALUE de NORM\_X doit être limité à  $MIN \leq VALUE \leq MAX$**

Si le paramètre VALUE est inférieur à MIN ou supérieur à MAX, la mise à l'échelle linéaire peut produire des valeurs OUT normalisées qui sont inférieures à 0.0 ou supérieures à 1.0. Dans ce cas, l'exécution de NORM\_X met ENO à VRAI.

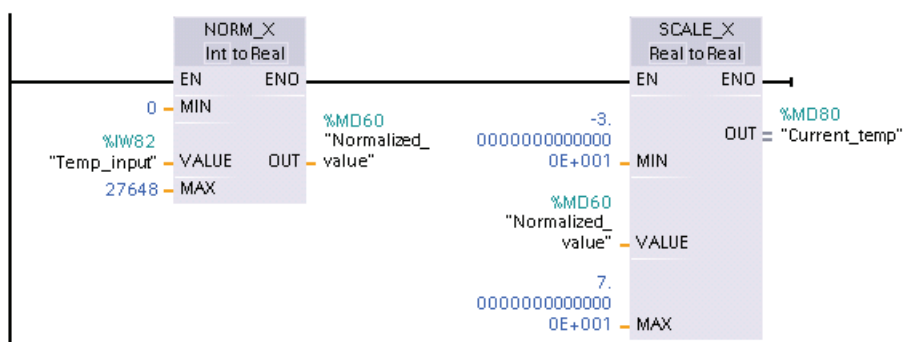
Tableau 7- 90 Etat de ENO

ENO	Situation d'erreur	Résultat OUT
1	Pas d'erreur	Résultat valide
0	Le résultat dépasse la plage valide pour le type de données OUT.	Résultat intermédiaire : partie la moins significative d'un nombre réel avant la conversion finale au type de données OUT.
0	Paramètre $MAX \leq MIN$	SCALE_X : La partie la moins significative du nombre réel VALUE complète la taille de OUT. NORM_X : VALUE dans le type de données VALUE étendu pour remplir une taille de double mot.
0	Paramètre VALUE = +/- INF ou +/- NaN	VALUE est écrit dans OUT

### Exemple (CONT) : Normalisation et mise à l'échelle d'une valeur d'entrée analogique

Une entrée analogique d'un module d'entrées-sorties ou Signal Board analogique avec entrée en courant se situe dans la plage 0 à 27648 des valeurs valides. Supposons qu'une entrée analogique représente une température, avec la valeur 0 de l'entrée analogique correspondant à -30.0 °C et la valeur 27648 correspondant à 70.0 °C.

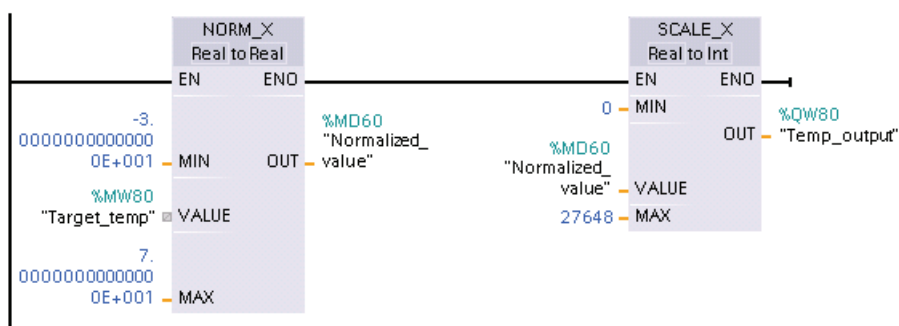
Pour transformer la valeur analogique en unités physiques correspondantes, normalisez l'entrée à une valeur comprise entre 0.0 et 1.0, puis mettez-la à l'échelle entre -30.0 et 70.0. La valeur résultante est la température représentée par l'entrée analogique en degrés Celsius :



Notez que si l'entrée analogique était celle d'un module d'entrées-sorties ou Signal Board analogique en tension, la valeur MIN pour l'instruction NORM\_X serait -27648 et non 0.

### Exemple (CONT) : Normalisation et mise à l'échelle d'une valeur de sortie analogique

Une sortie analogique à définir dans un module d'entrées-sorties ou Signal Board analogique avec sortie en courant doit se situer dans la plage 0 à 27648 des valeurs valides. Supposons qu'une sortie analogique représente un réglage de température, avec la valeur 0 de la sortie analogique correspondant à -30.0 °C et la valeur 27648 correspondant à 70.0 °C. Pour convertir une valeur de température comprise entre -30.0 et 70.0 en mémoire à une valeur pour la sortie analogique qui se situe dans la plage 0 à 27648, vous devez normaliser la valeur en unités physiques à une valeur comprise entre 0.0 et 1.0, puis la mettre à l'échelle à la plage de la sortie analogique, c'est-à-dire 0 à 27648 :



Notez que si la sortie analogique était destinée à un module d'entrées-sorties ou Signal Board analogique en tension, la valeur MIN pour l'instruction SCALE\_X serait -27648 et non 0.

Vous trouverez des informations supplémentaires sur les représentations des entrées analogiques (Page 822) et des sorties analogiques (Page 823) en tension et en courant dans les Caractéristiques techniques.

## 7.8 Gestion du programme

### 7.8.1 Présentation des instructions de contrôle SCL

Le langage SCL (Structured Control Language) fournit trois types d'instructions de contrôle pour structurer votre programme utilisateur :

- Instructions de sélection : Une instruction de sélection vous permet d'aiguiller le déroulement du programme vers différentes séquences d'instructions.
- Boucles : Vous pouvez programmer l'exécution de boucles à l'aide d'instructions d'itération. Une instruction d'itération précise quelles parties d'un programme doivent être répétées en fonction de certaines conditions.
- Sauts dans le programme : Un saut dans le programme provoque le saut immédiat à un repère de saut indiqué et donc à une autre instruction dans le même bloc.

Ces instructions de contrôle du programme utilisent la syntaxe du langage de programmation PASCAL.

Tableau 7- 91 Types d'instructions de contrôle SCL

Instruction de contrôle		Description
Instruction de sélection	Instruction IF-THEN (Page 237)	Permet d'aiguiller le déroulement du programme vers une branche d'une alternative selon qu'une condition est vraie ou fausse.
	Instruction CASE (Page 238)	Permet l'exécution sélective d'une variante parmi $n$ variantes en fonction de la valeur d'une variable.
Boucle	Instruction FOR (Page 239)	Répète une séquence d'instructions tant que la variable de contrôle reste dans la plage de valeurs indiquée.
	Instruction WHILE-DO (Page 240)	Répète une séquence d'instructions tant qu'une condition d'exécution est satisfaite.
	Instruction REPEAT-UNTIL (Page 241)	Répète une séquence d'instructions jusqu'à ce qu'une condition d'achèvement apparaisse.
Saut dans le programme	Instruction CONTINUE (Page 242)	Sert à abandonner l'exécution de l'itération en cours de la boucle.
	Instruction EXIT (Page 243)	Sert à quitter une boucle en n'importe quel point, que la condition d'abandon soit remplie ou non.
	Instruction GOTO (Page 243)	Provoque le saut immédiat du programme à un repère indiqué.
	Instruction IF-THEN (Page 237)	Permet de quitter le bloc en cours d'exécution et de retourner au bloc appelant.

**Voir aussi**

Instruction RETURN (Page 244)

## 7.8.2 Instruction IF-THEN

Une instruction IF-THEN est une instruction conditionnelle qui contrôle le déroulement du programme en exécutant un groupe d'instructions en fonction de l'évaluation de la valeur Bool d'une expression logique. Vous pouvez utiliser des crochets pour imbriquer ou structurer l'exécution de plusieurs instructions IF-THEN.

Tableau 7- 92 Eléments de l'instruction IF-THEN

SCL	Description
<pre>IF "condition" THEN   instruction_A;   instruction_B;   instruction_C; ;</pre>	<p>Si "condition" est VRAI ou 1, le programme exécute les instructions suivantes jusqu'à ce qu'il rencontre l'instruction END_IF.</p> <p>Si "condition" est FAUX ou 0, le programme va à l'instruction END_IF (à moins qu'il ne contienne des instructions ELSIF ou ELSE optionnelles).</p>
<pre>[ELSIF "condition-n" THEN   instruction_N; ;]</pre>	<p>L'instruction ELSIF<sup>1</sup> optionnelle fournit des conditions additionnelles à évaluer. Par exemple, si "condition" dans l'instruction IF-THEN est FAUX, le programme évalue "condition-n". Si "condition-n" est VRAI, le programme exécute "instruction_N".</p>
<pre>[ELSE   instruction_X; ;]</pre>	<p>L'instruction ELSE optionnelle fournit des instructions à exécuter lorsque "condition" dans l'instruction IF-THEN est FAUX.</p>
<pre>END_IF;</pre>	<p>L'instruction END_IF met fin à l'instruction IF-THEN.</p>

<sup>1</sup> Vous pouvez inclure plusieurs instructions ELSIF à l'intérieur d'une instruction IF-THEN.

Tableau 7- 93 Variables pour l'instruction IF-THEN

Variables	Description
"condition"	Obligatoire. L'expression logique est soit vraie (TRUE ou 1), soit fausse (FALSE ou 0).
"instruction_A"	Facultatif. Une ou plusieurs instructions à exécuter lorsque "condition" est VRAI.
"condition-n"	Facultatif. Expression logique que l'instruction optionnelle ELSIF doit évaluer.
"instruction_N"	Facultatif. Une ou plusieurs instructions à exécuter lorsque "condition-n" de l'instruction ELSIF est VRAI.
"instruction_X"	Facultatif. Une ou plusieurs instructions à exécuter lorsque "condition" de l'instruction IF-THEN est FAUX.

L'exécution d'une instruction IF est régie par les règles suivantes :

- La première séquence d'instructions dont l'expression logique est vraie (TRUE) est exécutée. Les séquences d'instructions restantes ne sont pas exécutées.
- Si aucune expression booléenne n'est vraie, c'est la séquence d'instructions introduite par ELSE qui est exécutée (ou aucune séquence d'instructions s'il n'y a pas de branche ELSE).
- Le nombre d'instructions ELSIF possibles est illimité.

**Remarque**

L'avantage d'utiliser une ou plusieurs branches ELSIF plutôt qu'une suite d'instructions IF réside dans le fait que les expressions logiques qui suivent une expression évaluée comme vraie ne sont plus évaluées. Cela permet de réduire la durée d'exécution du programme.

### 7.8.3 Instruction CASE

Tableau 7- 94 Eléments de l'instruction CASE

SCL	Description
<pre> CASE "Valeur_test" OF   "ListeValeurs": Instruction[; Instruction, ...]   "ListeValeurs": Instruction[; Instruction, ...] [ELSE Instruction-Else[; Instruction-Else, ...] END CASE;</pre>	<p>L'instruction CASE exécute un groupe d'instructions parmi plusieurs groupes en fonction de la valeur d'une expression.</p>

Tableau 7- 95 Paramètres

Paramètre	Description
"Valeur_test"	Obligatoire. Toute expression numérique de type de données Int
"ListeValeurs"	Obligatoire. Valeur unique ou liste de valeurs ou de plages de valeurs séparées par des virgules (utilisez deux points pour définir une plage de valeurs : 2..8). L'exemple suivant illustre les différents types de listes de valeurs : 1: Instruction_A; 2, 4: Instruction_B; 3, 5..7,9: Instruction_C;
Instruction	Obligatoire. Une ou plusieurs instructions qui sont exécutées lorsque "Valeur_test" correspond à une valeur quelconque dans la liste de valeurs.
Instruction-Else	Facultatif. Une ou plusieurs instructions qui sont exécutées s'il n'existe aucune occurrence de "Valeur_test" dans la liste "ListeValeurs".

L'exécution d'une instruction CASE est régie par les règles suivantes :

- L'expression de sélection doit renvoyer une valeur de type Int.
- Lors de l'exécution d'une instruction CASE, le programme vérifie si la valeur de l'expression de sélection figure dans la liste de valeurs spécifiée. Si tel est le cas, la section d'instructions affectée à la liste est exécutée.
- Si la valeur de l'expression de sélection est introuvable, c'est la section d'instructions suivant ELSE qui est exécutée ou aucune instruction s'il n'y a pas de branche ELSE.

Il est possible d'imbriquer des instructions CASE. Chaque instruction CASE imbriquée doit être associée à une instruction END\_CASE.

```

CASE var1 OF
    1 : var2 := "A";
    2 : var2 := "B";
ELSE
    CASE var3 OF
        65..90: var2 := "UpperCase";
        97..122: var2 := "LowerCase";
    ELSE
        var2:= "SpecialCharacter";
    END_CASE;
END_CASE;

```

## 7.8.4 Instruction FOR

Tableau 7- 96 Eléments de l'instruction FOR

SCL	Description
<pre> FOR "variable_contrôle" := "début" TO "fin" [BY "incrément"] DO     instruction; ; END_FOR; </pre>	<p>Une instruction FOR sert à répéter une séquence d'instructions aussi longtemps qu'une variable de contrôle se trouve dans la plage de valeurs spécifiée. La définition d'une boucle avec FOR comporte aussi l'indication d'une valeur initiale et d'une valeur finale. Ces deux valeurs doivent avoir le même type que la variable de contrôle.</p>

Tableau 7- 97 Paramètres

Paramètre	Description
"variable_contrôle"	Obligatoire. Entier (Int ou DInt) servant de compteur de boucle
"début"	Obligatoire. Expression simple qui indique la valeur initiale de la variable de contrôle.

Paramètre	Description
"fin"	Obligatoire. Expression simple qui indique la valeur finale de la variable de contrôle.
"incrément"	Facultatif. Quantité de laquelle la variable de contrôle est modifiée après chaque boucle. "incrément" doit avoir le même type de données que la variable de contrôle. Si aucun incrément n'est indiqué, la valeur des variables d'exécution est augmentée de 1 après chaque boucle. Vous ne pouvez pas modifier la valeur de "incrément" pendant l'exécution de l'instruction FOR.

L'instruction FOR s'exécute comme suit :

- Au début de la boucle, la variable de contrôle prend la valeur initiale (affectation initiale) puis, à chaque itération de la boucle, elle est incrémentée (en cas d'incrément positif) ou décrétementée (en cas d'incrément négatif) de l'incrément indiqué, et ce jusqu'à ce que la valeur finale soit atteinte.
- Après chaque exécution de la boucle, la condition (valeur finale atteinte) est vérifiée afin de constater si elle est ou non remplie. Si la condition est remplie, la séquence d'instructions est exécutée. Sinon, la boucle et sa séquence d'instructions sont sautées.

Règles de formulation des instructions FOR :

- La variable de contrôle doit être exclusivement du type de données Int ou DInt.
- Vous pouvez omettre l'instruction BY [incrément]. Si aucun incrément est indiqué, la valeur +1 est prise par défaut.

Utilisez l'Instruction EXIT (Page 243) pour quitter la boucle quel que soit l'état de l'expression "condition". L'instruction EXIT exécute l'instruction se trouvant immédiatement après l'instruction END\_FOR.

Utilisez l'Instruction CONTINUE (Page 242) pour sauter les instructions suivantes dans la boucle FOR et aller immédiatement vérifier que la condition d'abandon de la boucle est remplie ou non.

### 7.8.5 Instruction WHILE-DO

Tableau 7- 98 Instruction WHILE

SCL	Description
<pre>WHILE "condition" DO     Instruction;     Instruction;     ...; END WHILE;</pre>	<p>L'instruction WHILE exécute une série d'instructions tant qu'une condition donnée est vraie (TRUE).</p> <p>Vous pouvez imbriquer des boucles WHILE. L'instruction END_WHILE se réfère à la dernière instruction WHILE exécutée.</p>

Tableau 7- 99 Paramètres

Paramètre	Description
"condition"	Obligatoire. Expression logique qui peut prendre la valeur VRAI ou FAUX (une condition "nulle" est interprétée comme fausse).
Instruction	Facultatif. Une ou plusieurs instructions qui sont exécutées tant que la condition est vraie.



### Remarque

L'instruction WHILE évalue l'état de "condition" avant d'exécuter les instructions. Pour exécuter les instructions au moins une fois quel que soit l'état de "condition", utilisez l'instruction REPEAT.

L'exécution d'une instruction WHILE est régie par les règles suivantes :

- La condition d'exécution est évaluée avant chaque itération du corps de boucle.
- Le corps de boucle suivant DO est répété tant que la condition d'exécution a la valeur VRAI.
- Dès que la condition d'exécution prend la valeur FAUX, la boucle est sautée et l'instruction suivant la boucle est exécutée.

Utilisez l'Instruction EXIT (Page 243) pour quitter la boucle quel que soit l'état de l'expression "condition". L'instruction EXIT exécute l'instruction se trouvant immédiatement après l'instruction END\_WHILE.

Utilisez l'instruction CONTINUE pour sauter les instructions suivantes dans la boucle WHILE et aller immédiatement vérifier que la condition d'abandon de la boucle est remplie ou non.

## 7.8.6 Instruction REPEAT-UNTIL

Tableau 7- 100 Instruction REPEAT

SCL	Description
<pre> REPEAT   Instruction; ; UNTIL "condition" END_REPEAT;</pre>	<p>L'instruction REPEAT exécute un groupe d'instructions jusqu'à ce qu'une condition donnée soit vraie (TRUE).</p> <p>Vous pouvez imbriquer des boucles REPEAT. L'instruction END_REPEAT se réfère toujours à la dernière instruction REPEAT exécutée.</p>

Tableau 7- 101 Paramètres

Paramètre	Description
Instruction	Facultatif. Une ou plusieurs instructions qui sont exécutées jusqu'à ce que la condition soit vraie.
"condition"	Obligatoire. Une ou plusieurs expressions des deux types suivants : Expression numérique ou chaîne de caractères qui peut prendre la valeur VRAI ou FAUX. Une condition "nulle" est interprétée comme fausse.

**Remarque**

Avant d'évaluer l'état de "condition", l'instruction REPEAT exécute une fois les instructions dans la boucle (même si "condition" est FAUX). Pour que l'état de "condition" soit évalué avant l'exécution des instructions, utilisez l'instruction WHILE.

Utilisez l'Instruction EXIT (Page 243) pour quitter la boucle quel que soit l'état de l'expression "condition". L'instruction EXIT exécute l'instruction se trouvant immédiatement après l'instruction END\_REPEAT.

Utilisez l'Instruction CONTINUE (Page 242) pour sauter les instructions suivantes dans la boucle REPEAT et aller immédiatement vérifier que la condition d'abandon de la boucle est remplie ou non.

### 7.8.7 Instruction CONTINUE

Tableau 7- 102 Instruction CONTINUE

SCL	Description
<b>CONTINUE</b> Instruction; ;	L'instruction CONTINUE saute les instructions suivantes d'un programme de boucle (FOR, WHILE, REPEAT) et va immédiatement vérifier que la condition d'abandon de la boucle est remplie ou non. Si ce n'est pas le cas, la boucle continue.

L'exécution d'une instruction CONTINUE est régie par les règles suivantes :

- Cette instruction met immédiatement fin à l'exécution du corps de la boucle.
- Selon que la condition de répétition de la boucle est remplie ou non, le corps de la boucle est réexécuté ou bien l'instruction d'itération est abandonnée et l'instruction venant immédiatement après est exécutée.
- Dans une instruction FOR, la variable de contrôle est incrémentée de l'incrément indiqué aussitôt après une instruction CONTINUE.

Utilisez l'instruction CONTINUE exclusivement à l'intérieur d'une boucle. Dans les boucles imbriquées, CONTINUE fait toujours référence à la boucle dans laquelle il est directement inclus. CONTINUE est typiquement utilisé en combinaison avec une instruction IF.

S'il s'agit d'abandonner la boucle quel que soit le résultat du test d'achèvement, utilisez l'instruction EXIT.

L'exemple suivant montre l'utilisation de l'instruction CONTINUE pour éviter une division par 0 lors du calcul du pourcentage d'une valeur :

```

FOR x = 0 TO 10 DO
IF valeur[i] = 0 THEN CONTINUE; END_IF;
    p := partie / valeur[i] * 100;
    s := INT_TO_STRING(p);
    pourcentage=CONCAT(IN1:=s, IN2:="%");
END_FOR;
    
```

## 7.8.8 Instruction EXIT

Tableau 7- 103 Instruction EXIT

SCL	Description
<b>EXIT;</b>	L'instruction EXIT permet de quitter une boucle (FOR, WHILE ou REPEAT) en n'importe quel point, que la condition d'achèvement soit remplie ou non.

L'exécution d'une instruction EXIT est régie par les règles suivantes :

- Cette instruction provoque l'abandon instantané de l'instruction d'itération qui entoure immédiatement l'instruction EXIT.
- L'exécution du programme se poursuit après la fin de la boucle (par exemple, après END\_FOR).

Utilisez l'instruction EXIT à l'intérieur d'une boucle. Dans les boucles imbriquées, l'instruction EXIT rend la main à la boucle de niveau immédiatement supérieur.

```
FOR i = 0 TO 10 DO
CASE valeur[i, 0] OF
  1..10: valeur [i, 1] := "A";
  11..40: valeur [i, 1] := "B";
  41..100: valeur [i, 1] := "C";
ELSE
EXIT;
END_CASE;
END_FOR;
```

## 7.8.9 Instruction GOTO

Tableau 7- 104 Instruction GOTO

SCL	Description
<b>GOTO JumpLabel;</b> <b>instruction;</b> <b>... ;</b> <b>JumpLabel: instruction;</b>	L'instruction GOTO évite des instructions en sautant directement à un repère dans le même bloc.  Le repère de saut ("JumpLabel") et l'instruction GOTO doivent être dans le même bloc. Le nom d'un repère de saut ne peut être affecté qu'une fois à l'intérieur d'un bloc. Chaque repère de saut peut être la cible de plusieurs instructions GOTO.

Il n'est pas possible d'entrer dans une boucle par un saut (FOR, WHILE ou REPEAT). Il est possible de quitter une boucle par un saut.

Dans l'exemple suivant, selon la valeur de l'opérande "valeur\_variable", l'exécution du programme reprend au point défini par le repère de saut correspondant. Si "valeur\_variable" est égal à 2, l'exécution du programme se poursuit au repère de saut "MyLabel2" et évite "MyLabel1".

```
CASE "valeur_variable" OF
  1 : GOTO MyLabel1;
  2 : GOTO MyLabel2;
ELSE GOTO MyLabel3;
END_CASE;
MyLabel1: "variable_1" := 1;
```

```
MyLabel12: "variable_2" := 1;
MyLabel13: "variable_4" := 1;
```

### 7.8.10 Instruction RETURN

Tableau 7- 105 Instruction RETURN

SCL	Description
RETURN ;	L'instruction RETURN quitte sans condition le bloc de code en cours d'exécution. L'exécution du programme revient au bloc appelant ou au système d'exploitation (lors de la sortie d'un OB).

Exemple d'instruction RETURN :

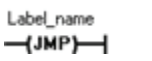

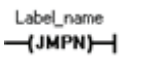



```
IF "Erreur" <> 0 THEN
RETURN ;
END_IF ;
```

#### Remarque

Après exécution de la dernière instruction, le bloc de code revient automatiquement au bloc appelant. N'insérez pas d'instruction RETURN à la fin du bloc de code.

### 7.8.11 Instructions Saut et Repère de saut

Tableau 7- 106 Instructions JMP, JMPN et LABEL

CONT	LOG	SCL	Description
		Voir l'instruction GOTO (Page 243).	S'il y a flux de courant vers la bobine JMP (CONT) ou si l'entrée de la boîte JMP est vraie (LOG), l'exécution du programme continue avec la première instruction suivant le repère de saut indiqué.
			S'il n'y a pas de flux de courant vers la bobine JMPN (CONT) ou si l'entrée de la boîte JMPN est fausse (LOG), l'exécution du programme continue avec la première instruction suivant le repère de saut indiqué.
			Destination pour une instruction de saut JMP ou JMPN.

<sup>1</sup> Vous créez vos noms de repère en tapant directement dans l'instruction LABEL. Utilisez l'icône d'aide des paramètres pour sélectionner les noms de repère disponibles pour le champ de nom de repère de JMP et JMPN. Vous pouvez également saisir un nom de repère directement dans l'instruction JMP ou JMPN.

Tableau 7- 107 Types de données pour les paramètres

Paramètre	Type de données	Description
Label_name	Identificateur de repère	Identificateur pour les instructions de saut et repère de destination de saut correspondant

- Chaque repère doit être univoque à l'intérieur d'un bloc de code.
- Vous pouvez sauter à l'intérieur d'un bloc de code, mais vous ne pouvez pas sauter d'un bloc de code à un autre bloc de code.
- Vous pouvez sauter en avant ou en arrière.
- Vous pouvez sauter au même repère à partir de plus d'un point dans le même bloc de code.

## 7.8.12 Instruction JMP\_LIST

Tableau 7- 108 Instruction JMP\_LIST


CONT/LOG	SCL	Description
	<pre> CASE k OF   0: GOTO dest0;   1: GOTO dest1;   2: GOTO dest2;   [n: GOTO destn;] END_CASE; </pre>	L'instruction JMP_LIST agit comme un répartiteur de sauts pour piloter l'exécution de sections de programme. Un saut est exécuté au repère de programme correspondant à la valeur de l'entrée K. L'exécution du programme se poursuit avec les instructions qui suivent le repère de saut de destination. Si la valeur de l'entrée K dépasse le nombre de repères - 1, aucun saut n'est effectué et le traitement se poursuit avec le réseau suivant du programme.

Tableau 7- 109 Types de données pour les paramètres

Paramètre	Type de données	Description
K	UInt	Valeur de commande du répartiteur de sauts
DEST0, DEST1, ..., DESTn.	Repères de programme	Repères de destination de saut correspondant à des valeurs spécifiques du paramètre K.  Si la valeur de K est égale à 0, un saut est effectué au repère de programme affecté à la sortie DEST0. Si la valeur de K est égale à 1, un saut est effectué au repère de programme affecté à la sortie DEST1 et ainsi de suite. Si la valeur de l'entrée K dépasse le nombre de repères - 1, aucun saut n'est effectué et le traitement se poursuit avec le réseau suivant du programme.

Pour CONT et LOG : Lorsque vous insérez initialement la boîte JMP\_LIST dans votre programme, elle ne comporte que deux sorties de repère de saut. Vous pouvez ajouter ou supprimer des destinations de saut.



Cliquez sur l'icône de création à l'intérieur de la boîte (à gauche du dernier paramètre DEST) pour ajouter de nouvelles sorties de repères de saut.



- Cliquez avec le bouton droit de la souris sur une ligne de sortie et sélectionnez la commande "Insérer sortie".
- Cliquez avec le bouton droit de la souris sur une ligne de sortie et sélectionnez la commande "Supprimer".

### 7.8.13 Instruction SWITCH

Tableau 7- 110 Instruction SWITCH

CONT/LOG	SCL	Description
	Non disponible	L'instruction SWITCH agit comme un répartiteur de sauts pour piloter l'exécution de sections de programme. La valeur de l'entrée K est comparée aux valeurs affectées aux entrées de comparaison indiquées et un saut est effectué au repère de programme correspondant au premier test de comparaison qui est vrai. Si aucune des comparaisons n'est vraie, il y a saut au repère affecté à ELSE. L'exécution du programme se poursuit avec les instructions qui suivent le repère de saut de destination.

- 1 Pour CONT et LOG : Cliquez sous le nom de la boîte et sélectionnez un type de données dans le menu déroulant.
- 2 Pour SCL : Utilisez un jeu de comparaisons IF-THEN.

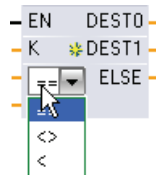
Tableau 7- 111 Types de données pour les paramètres

Paramètre	Type de données <sup>1</sup>	Description
K	SInt, Int, DInt, USInt, UInt, UDIInt, Real, LReal, Byte, Word, DWord, Time, TOD, Date	Entrée de valeur de comparaison commune
==, <>, <, <=, >, >=	SInt, Int, DInt, USInt, UInt, UDIInt, Real, LReal, Byte, Word, DWord, Time, TOD, Date	Entrées de valeurs de comparaison distinctes pour des types de comparaison spécifiques
DEST0, DEST1, ..., DESTn, ELSE	Repères de programme	Repères de destination de saut correspondant à des comparaisons spécifiques :  L'entrée de comparaison la plus proche en dessous de l'entrée K est traitée en premier et provoque un saut au repère affecté à DEST0 si la comparaison entre la valeur K et cette entrée est vraie. Le test de comparaison suivant utilise l'entrée du dessous suivante et provoque un saut au repère affecté à DEST1 si la comparaison est vraie, et ainsi de suite avec les comparaisons restantes. Si aucune des comparaisons n'est vraie, il y a saut au repère affecté à la sortie ELSE.

<sup>1</sup> L'entrée K et les entrées de comparaison (==, <>, <, <=, >, >=) doivent avoir le même type de données.

### Ajout et suppression d'entrées et indication des types de comparaison

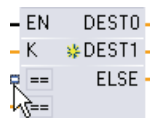
Lorsque la boîte SWITCH CONT ou LOG est initialement insérée dans votre programme, elle comporte deux entrées de comparaison. Vous pouvez affecter des types de comparaison et ajouter des entrées/destinations de saut comme illustré ci-dessous.



Cliquez sur un opérateur de comparaison dans la boîte et sélectionnez un nouvel opérateur dans la liste déroulante.



Cliquez sur l'icône de création à l'intérieur de la boîte (à gauche du dernier paramètre DEST ) pour ajouter de nouveaux paramètres de destination pour la comparaison.



- Cliquez avec le bouton droit de la souris sur une ligne d'entrée et sélectionnez la commande "Insérer entrée".
- Cliquez avec le bouton droit de la souris sur une ligne d'entrée et sélectionnez la commande "Supprimer".

Tableau 7- 112 Sélection de type de données et opérations de comparaison autorisées pour la boîte SWITCH

Type de données	Comparaison	Syntaxe de l'opérateur
Byte, Word, DWord	Egal	==
	Différent	<>
SInt, Int, DInt, USInt, UInt, UDInt, Real, LReal, Time, TOD, Date	Egal	==
	Différent	<>
	Supérieur ou égal	>=
	Inférieur ou égal	<=
	Supérieur à	>
	Inférieur à	<

### Règles de placement de la boîte SWITCH

- Aucune connexion d'instruction CONT/LOG devant l'entrée de comparaison n'est autorisée.
- Il n'existe pas d'entrée ENO. On ne peut donc avoir qu'une instruction SWITCH dans un réseau et l'instruction SWITCH doit être la dernière opération du réseau.

### 7.8.14 Instruction de gestion d'exécution RET

L'instruction RET optionnelle sert à mettre fin à l'exécution du bloc en cours. Si et seulement si il y a flux de courant vers la bobine RET (CONT) ou si l'entrée de la boîte RET est vraie (CONT), l'exécution du programme du bloc en cours est interrompue en ce point et les instructions au-delà de l'instruction RET ne sont pas exécutées. Si le bloc en cours est un OB, le paramètre "Return\_Value" n'est pas pris en compte. Si le bloc en cours est une FC ou un FB, la valeur du paramètre "Return\_Value" est transmise au programme appelant en tant que valeur ENO de la boîte appelée.

Vous n'avez pas besoin d'utiliser une instruction RET comme dernière instruction dans un bloc ; cela est fait automatiquement pour vous. Vous pouvez avoir plusieurs instructions RET à l'intérieur d'un seul bloc.

Pour SCL, reportez-vous à l'instruction RETURN (Page 244).

Tableau 7- 113 Instruction de gestion d'exécution RET (Renvoyer la valeur)



CONT	LOG	SCL	Description
		RETURN ;	Met fin à l'exécution du bloc en cours.



Tableau 7- 114 Types de données pour les paramètres

Paramètre	Type de données	Description
Return_Value	Bool	Le paramètre "Return_value" de l'instruction RET est affecté à la sortie ENO de la boîte d'appel de bloc dans le bloc appelant.

Modèle d'étapes pour l'utilisation de l'instruction RET dans une FC :


1. Créez un nouveau projet et insérez-y une FC.
2. Editez la FC :
  - Ajoutez des instructions depuis l'arborescence d'instructions.
  - Ajoutez une instruction RET en indiquant l'une des valeurs suivantes pour le paramètre "Return\_Value" :  
VRAI, FAUX ou une adresse de mémoire contenant la valeur de retour requise.
  - Ajoutez d'autres instructions.
3. Appelez la FC depuis le programme principal [OB1].

L'entrée EN de la boîte FC dans l'OB 1 doit être vraie pour que l'exécution de la FC commence.

La valeur indiquée par l'instruction RET dans la FC sera présente au niveau de la sortie ENO de la boîte FC dans le bloc OB 1 après l'exécution de la FC pour laquelle le flux de courant vers l'instruction RET est vrai.

### 7.8.15 Instruction Redéclencher le temps de surveillance du cycle

Tableau 7- 115 RE\_TRIGR, instruction

CONT/LOG	SCL	Description
	<code>RE_TRIGR ( ) ;</code>	RE_TRIGR (Redémarrer surveillance du temps de cycle) sert à allonger le temps maximum autorisé avant que la temporisation de surveillance du cycle ne génère une erreur.

Utilisez l'instruction RE\_TRIGR pour redémarrer la temporisation de surveillance du cycle pendant un cycle unique. Cela a pour effet d'allonger le temps de cycle maximum autorisé d'une période de temps de cycle maximum, à partir de la dernière exécution de la fonction RE\_TRIGR.

**Remarque**

Avant la version de firmware 2.2 des CPU S7-1200, RE\_TRIGR ne pouvait être exécuté qu'à partir d'un OB de cycle de programme et pouvait être utilisé pour allonger le temps de cycle API indéfiniment. ENO = FALSE et la temporisation de surveillance du cycle n'est pas réinitialisée lorsque RE\_TRIGR était exécuté à partir d'un OB de démarrage, d'un OB d'alarme ou d'un OB d'erreur.

A partir de la version de firmware 2.2, RE-TRIGR peut être exécuté dans n'importe quel OB (OB de démarrage, d'alarme et d'erreur compris). Toutefois, le cycle API ne peut être allongé que de 10 fois le temps de cycle maximum configuré au plus.

**Définition du temps de cycle maximum de l'automate**

Configurez la valeur du temps de cycle maximum sous "Temps de cycle" dans la configuration des appareils.

Tableau 7- 116 Valeurs du temps de cycle

Surveillance du temps de cycle	Valeur minimale	Valeur maximale	Valeur par défaut
Temps de cycle maximum	1 ms	6000 ms	150 ms

**Expiration de la temporisation de surveillance du cycle**

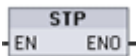
Une erreur est générée si la temporisation de cycle maximum expire avant que le cycle ne soit achevé. Si un bloc de code de gestion d'erreur OB 80 est inclus dans le programme utilisateur, la CPU exécute l'OB 80 dans lequel vous pouvez insérer de la logique pour créer une réaction spécifique. En l'absence d'OB 80, la première expiration n'est pas prise en compte et la CPU passe à l'état ARRET.

Si une deuxième expiration du temps de cycle maximum se produit pendant le même cycle du programme (2 fois la valeur du temps de cycle maximum), une erreur provoquant le passage de la CPU à l'état ARRET est générée.

A l'état ARRET, l'exécution de votre programme s'arrête alors que la communication système de la CPU et le diagnostic système se poursuivent.

## 7.8.16 Instruction Commuter en STOP

Tableau 7- 117 Instruction STP

CONT/LOG	SCL	Description
	<code>STP ( ) ;</code>	STP (Arrêter) fait passer la CPU à l'état ARRET. Lorsque la CPU est à l'état ARRET, l'exécution de votre programme et les actualisations physiques depuis la mémoire image s'arrêtent.

Pour plus d'informations, reportez-vous à Configuration des sorties lors d'un passage de MARCHE à ARRET (Page 93).

Si EN = VRAI, la CPU passe à l'état ARRET, l'exécution du programme s'arrête et l'état de ENO est sans objet. Sinon, EN = ENO = 0.

## 7.8.17 Instructions de traitement d'erreur

Les instructions de traitement d'erreur fournissent des informations sur les erreurs d'exécution des blocs de programme. Si vous ajoutez une instruction GetError ou GetErrorID à votre bloc de code, vous pouvez gérer les erreurs de programme à l'intérieur de votre bloc de programme.

### GetError

Tableau 7- 118 Instruction GetError

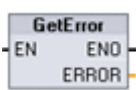
CONT/LOG	SCL	Description
	<code>GET_ERROR ( _o ut_ ) ;</code>	Signale qu'une erreur d'exécution du bloc de programme s'est produite et remplit une structure de données d'erreur prédéfinie avec des informations d'erreur détaillées.

Tableau 7- 119 Types de données pour les paramètres

Paramètre	Type de données	Description
ERROR	ErrorStruct	Structure de données d'erreur : Vous pouvez renommer la structure mais pas les éléments à l'intérieur de la structure.

Tableau 7- 120 Eléments de la structure de données ErrorStruct

Eléments de la structure		Type de données	Description					
ERROR_ID		Word	Code d'erreur					
FLAGS		Byte	Indique si une erreur s'est produite pendant un appel de bloc. <ul style="list-style-type: none"> <li>• 16#01: Erreur pendant un appel de bloc</li> <li>• 16#00: Aucune erreur pendant un appel de bloc</li> </ul>					
REACTION		Byte	Réaction par défaut : <ul style="list-style-type: none"> <li>• 0: Ne pas tenir compte de l'erreur (erreur d'écriture)</li> <li>• 1: Poursuivre avec la valeur de remplacement "0" (erreur de lecture)</li> <li>• 2: Sauter l'instruction (erreur système)</li> </ul>					
CODE_ADDRESS		CREF	Information sur l'adresse et le type de bloc					
	BLOCK_TYPE	Byte	Type de bloc où l'erreur s'est produite : <ul style="list-style-type: none"> <li>• 1: OB</li> <li>• 2: FC</li> <li>• 3: FB</li> </ul>					
	CB_NUMBER	UInt	Numéro du bloc de code					
	OFFSET	UDInt	Référence à la mémoire interne					
MODE		Byte	Mode d'accès : Les informations suivantes peuvent être transmises selon le type d'accès :					
			Mode	(A)	(B)	(C)	(D)	(E)
			0					
			1					Décalage
			2			Zone		
			3	Adresse	Portée		Numéro	
			4			Zone		Décalage
			5			Zone	N° DB	Décalage
			6	N° Ptr /Acc		Zone	N° DB	Décalage
7	N° Ptr /Acc	N° empl./Portée	Zone	N° DB	Décalage			
OPERAND_NUMBER		UInt	Numéro d'opérande de la commande machine					
POINTER_NUMBER_LOCATION		UInt	(A) Pointeur interne					
SLOT_NUMBER_SCOPE		UInt	(B) Zone de stockage en mémoire interne					
DATA_ADDRESS		NREF	Information sur l'adresse d'un opérande					

Éléments de la structure		Type de données	Description
	AREA	Byte	(C) Zone de mémoire <ul style="list-style-type: none"> <li>• L : 16#40 – 4E, 86, 87, 8E, 8F, C0 – CE</li> <li>• I : 16#81</li> <li>• Q : 16#82</li> <li>• M : 16#83</li> <li>• DB : 16#84, 85, 8A, 8B</li> </ul>
	DB_NUMBER	UInt	(D) Numéro du bloc de données
	OFFSET	UDInt	(E) Adresse relative de l'opérande

## GetErrorID

Tableau 7- 121 Instruction GetErrorID


CONT/LOG	SCL	Description
	<pre>GET_ERR_ID () ;</pre>	Signale qu'une erreur d'exécution du bloc de programme s'est produite et transmet l'ID (code identificateur) de l'erreur.

Tableau 7- 122 Types de données pour les paramètres

Paramètre	Type de données	Description
ID	Word	Valeurs de l'identificateur d'erreur pour l'élément ERROR_ID de ErrorStruct

Tableau 7- 123 Valeurs de Error\_ID

ERROR_ID hexadécimal	ERROR_ID décimal	Erreur d'exécution du bloc de programme
0	0	Pas d'erreur
2503	9475	Pointeur non initialisé
2522	9506	Erreur de lecture, opérande hors plage
2523	9507	Erreur d'écriture, opérande hors plage
2524	9508	Erreur de lecture, zone invalide
2525	9509	Erreur d'écriture, zone invalide
2528	9512	Erreur de lecture, alignement de données (alignement de bits incorrect)
2529	9513	Erreur d'écriture, alignement de données (alignement de bits incorrect)
2530	9520	DB protégé en écriture
253A	9530	DB global inexistant
253C	9532	Mauvaise version ou FC inexistante
253D	9533	Instruction inexistante

ERROR_ID hexadécimal	ERROR_ID décimal	Erreur d'exécution du bloc de programme
253E	9534	Mauvaise version ou FB inexistant
253F	9535	Instruction inexistante
2575	9589	Erreur de profondeur d'imbrication de programme
2576	9590	Erreur d'allocation de données locales
2942	10562	Entrée physique inexistante
2943	10563	Sortie physique inexistante

## Fonctionnement

Par défaut, la CPU réagit à une erreur d'exécution de bloc en consignait une erreur dans la mémoire tampon de diagnostic. Toutefois, si vous placez une ou plusieurs instructions `GetError` ou `GetErrorID` dans un bloc de code, ce bloc pourra gérer les erreurs se produisant à l'intérieur du bloc. Dans ce cas, la CPU ne consigne pas d'erreur dans la mémoire tampon de diagnostic. Au lieu de cela, l'information d'erreur est transmise à la sortie de l'instruction `GetError` ou `GetErrorID`. Vous pouvez lire les informations d'erreur détaillées avec l'instruction `GetError` ou uniquement l'identificateur de l'erreur avec l'instruction `GetErrorID`. Normalement, la première erreur est la plus importante, les erreurs suivantes n'étant que des conséquences de la première erreur.

La première exécution d'une instruction `GetError` ou `GetErrorID` à l'intérieur d'un bloc renvoie la première erreur détectée pendant l'exécution du bloc. Cette erreur peut s'être produite à n'importe quel endroit entre le début du bloc et l'exécution de `GetError` ou `GetErrorID`. Les exécutions suivantes de `GetError` ou `GetErrorID` renvoient la première erreur depuis l'exécution précédente de `GetError` ou `GetErrorID`. L'historique des erreurs n'est pas sauvegardé et l'exécution de l'une ou l'autre instruction réarme le système d'automatisation pour la capture de l'erreur suivante.

Vous pouvez ajouter le type de données `ErrorStruct` utilisé par l'instruction `GetError` dans l'éditeur de bloc de données et les éditeurs d'interface de bloc afin que la logique de votre programme puisse accéder à ces valeurs. Sélectionnez `ErrorStruct` dans la liste déroulante des types de données pour ajouter cette structure. Vous pouvez créer plusieurs structures `ErrorStruct` en leur donnant des noms univoques. Il n'est pas possible de renommer les éléments d'une structure `ErrorStruct`.

## Situation d'erreur signalée par ENO

Si `EN = TRUE` et `GetError` ou `GetErrorID` s'exécute, alors :

- `ENO = VRAI` signale une erreur d'exécution de bloc de code et la présence d'une erreur de données ;
- `ENO = FAUX` signale qu'aucune erreur d'exécution de bloc de code ne s'est produite.

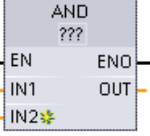
Vous pouvez connecter de la logique de programme de réaction aux erreurs à `ENO` qui s'active si une erreur s'est produite. En présence d'une erreur, le paramètre de sortie mémorise les données d'erreur là où votre programme peut y accéder.

Vous pouvez utiliser `GetError` et `GetErrorID` pour envoyer des informations d'erreur du bloc en cours d'exécution (bloc appelé) à un bloc appelant. Placez l'instruction dans le dernier réseau du programme du bloc appelé afin de renvoyer l'état d'exécution final au bloc appelé.

## 7.9 Opérations logiques sur mots

### 7.9.1 Instructions AND, OR et XOR

Tableau 7- 124 Instructions AND, OR et XOR

CONT/LOG	SCL	Description
	<code>out := in1 AND in2;</code>	AND : ET logique
	<code>out := in1 OR in2;</code>	OR : OU logique
	<code>out := in1 XOR in2;</code>	XOR : OU exclusif logique

<sup>1</sup> Pour CONT et LOG : Cliquez sur "???" et sélectionnez un type de données dans le menu déroulant.



Pour ajouter une entrée , cliquez sur l'icône de création ou cliquez avec le bouton droit de la souris sur la ligne d'entrée de l'un des paramètres IN existants et sélectionnez la commande "Insérer entrée".

Pour supprimer une entrée, cliquez avec le bouton droit de la souris sur la ligne d'entrée de l'un des paramètres IN existants (lorsqu'il y a plus d'entrées que les deux entrées d'origine) et sélectionnez la commande "Supprimer".

Tableau 7- 125 Types de données pour les paramètres


Paramètre	Type de données	Description
IN1, IN2	Byte, Word, DWord	Entrées logiques
OUT	Byte, Word, DWord	Sortie logique

<sup>1</sup> La sélection du type de données donne le même type de données aux paramètres IN1, IN2 et OUT.

Les valeurs binaires correspondantes de IN1 et IN2 sont combinées pour produire un résultat logique binaire qui est transmis au paramètre OUT. ENO est toujours VRAI après l'exécution de ces instructions.

### 7.9.2 Instruction Inverser

Tableau 7- 126 Instruction INV

CONT/LOG	SCL	Description
	Non disponible	Calcule le complément à un binaire du paramètre IN. Le complément à un est formé en inversant chaque valeur de bit dans le paramètre IN (changement de chaque 0 en 1 et de chaque 1 en 0). ENO est toujours VRAI après l'exécution de cette instruction.

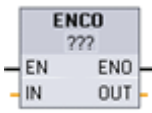
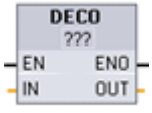
<sup>1</sup> Pour CONT et LOG : Cliquez sur "???" et sélectionnez un type de données dans le menu déroulant.

Tableau 7- 127 Types de données pour les paramètres

Paramètre	Type de données	Description
IN	SInt, Int, DInt, USInt, UInt, UDInt, Byte, Word, DWord	Élément de données à inverser
OUT	SInt, Int, DInt, USInt, UInt, UDInt, Byte, Word, DWord	Sortie inversée

### 7.9.3 Instructions Encoder et Décoder

Tableau 7- 128 Instructions ENCO et DECO

CONT/LOG	SCL	Description
	<code>out := ENCO(_in_);</code>	Code un motif binaire en un nombre binaire. L'instruction ENCO convertit le paramètre IN en un nombre binaire correspondant à la position du bit mis à 1 le moins significatif du paramètre IN et renvoie le résultat dans le paramètre OUT. Si le paramètre IN est soit 0000 0001 soit 0000 0000, la valeur 0 est renvoyée dans le paramètre OUT. Si la valeur du paramètre IN est 0000 0000, ENO est mis à FAUX.
	<code>out := DECO(_in_);</code>	Décode un nombre binaire en un motif binaire. L'instruction DECO décode un nombre binaire dans le paramètre IN en mettant à 1 la position de bit correspondante dans le paramètre OUT (tous les autres bits sont mis à 0). ENO est toujours VRAI après l'exécution de l'instruction DECO.  Remarque : Le type de données par défaut pour l'instruction DECO est DWORD. Dans SCL, modifiez le nom de l'instruction en DECO_BYTE ou DECO_WORD pour décoder une valeur d'octet ou de mot, respectivement, et indiquez une variable ou une adresse octet ou mot.

<sup>1</sup> Pour CONT et LOG : Cliquez sur "???" et sélectionnez un type de données dans le menu déroulant.



Tableau 7- 129 Types de données pour les paramètres

Paramètre	Type de données	Description
IN	ENCO : Byte, Word, DWord DECO : UInt	ENCO : Motif binaire à encoder DECO : Valeur à décoder
OUT	ENCO : Int DECO : Byte, Word, DWord	ENCO : Valeur encodée DECO : Motif binaire décodé

Tableau 7- 130 Paramètre OUT pour ENCO

ENO	Situation d'erreur	Résultat (OUT)
1	Pas d'erreur	Numéro de bit valide
0	IN est zéro.	OUT est mis à zéro.

La sélection du type de données Byte, Word ou DWord pour le paramètre OUT de DECO restreint la plage utile du paramètre IN. Si la valeur du paramètre IN dépasse la plage utile, une opération modulo est exécutée pour extraire les bits les moins significatifs comme illustré ci-dessous.

Plage du paramètre IN de DECO :

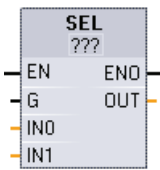
- 3 bits (valeurs 0-7) IN sont utilisés pour mettre à 1 une position de bit dans un octet OUT (Byte OUT).
- 4 bits (valeurs 0-15) IN sont utilisés pour mettre à 1 une position de bit dans un mot OUT (Word OUT).
- 5 bits (valeurs 0-31) IN sont utilisés pour mettre à 1 une position de bit dans un double mot OUT (DWord OUT).

Tableau 7- 131 Exemples

Valeur IN de DECO			Valeur OUT de DECO (Décoder une position de bit unique)
Byte OUT 8 bits	Min. IN	0	00000001
	Max. IN	7	10000000
Word OUT 16 bits	Min. IN	0	0000000000000001
	Max. IN	15	1000000000000000
DWord OUT 32 bits	Min. IN	0	00000000000000000000000000000001
	Max. IN	31	10000000000000000000000000000000

### 7.9.4 Instructions Sélectionner, Multiplexeur et Démultiplexeur

Tableau 7- 132 Instruction SEL (Sélectionner)

CONT/LOG	SCL	Description
	<pre>out := SEL(     g:=_bool_in,     in0:=-variant_in,     in1:=-variant_in);</pre>	SEL affecte une valeur parmi deux valeurs d'entrée au paramètre OUT en fonction de la valeur du paramètre G.

<sup>1</sup> Pour CONT et LOG : Cliquez sur "???" et sélectionnez un type de données dans le menu déroulant.

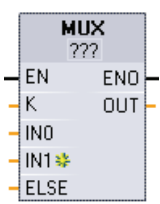
Tableau 7- 133 Types de données pour l'instruction SEL

Paramètre	Type de données <sup>1</sup>	Description
G	Bool	<ul style="list-style-type: none"> <li>• 0 sélectionne IN0</li> <li>• 1 sélectionne IN1</li> </ul>
IN0, IN1	SInt, Int, DInt, USInt, UInt, UDIInt, Real, LReal, Byte, Word, DWord, Time, Char	Entrées
OUT	SInt, Int, DInt, USInt, UInt, UDIInt, Real, LReal, Byte, Word, DWord, Time, Char	Sortie

<sup>1</sup> Les variables d'entrée et la variable de sortie doivent avoir le même type de données.

**Codes d'erreur :** ENO est toujours vrai après l'exécution de l'instruction SEL.

Tableau 7- 134 Instruction MUX (Multiplexeur)

CONT/LOG	SCL	Description
	<pre>out := MUX(     k:=_unit_in,     in1:=variant_in,     in2:=variant_in,     [...in32:=variant_in,]     inelse:=variant_in);</pre>	MUX copie une valeur parmi plusieurs valeurs d'entrée dans le paramètre OUT en fonction de la valeur du paramètre K. Si la valeur du paramètre K dépasse (IN <sub>n</sub> - 1), la valeur du paramètre ELSE est copiée dans le paramètre OUT.

<sup>1</sup> Pour CONT et LOG : Cliquez sur "???" et sélectionnez un type de données dans le menu déroulant.



Pour ajouter une entrée , cliquez sur l'icône de création ou cliquez avec le bouton droit de la souris sur la ligne d'entrée de l'un des paramètres IN existants et sélectionnez la commande "Insérer entrée".

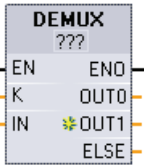
Pour supprimer une entrée, cliquez avec le bouton droit de la souris sur la ligne d'entrée de l'un des paramètres IN existants (lorsqu'il y a plus d'entrées que les deux entrées d'origine) et sélectionnez la commande "Supprimer".

Tableau 7- 135 Types de données pour l'instruction MUX

Paramètre	Type de données	Description
K	UInt	<ul style="list-style-type: none"> <li>• 0 sélectionne IN1</li> <li>• 1 sélectionne IN2</li> <li>• <math>n</math> sélectionne IN<math>n</math></li> </ul>
IN0, IN1, .. INn	SInt, Int, DInt, USInt, UInt, UDIInt, Real, LReal, Byte, Word, DWord, Time, Char	Entrées
ELSE	SInt, Int, DInt, USInt, UInt, UDIInt, Real, LReal, Byte, Word, DWord, Time, Char	Valeur de substitution d'entrée (optionnelle)
OUT	SInt, Int, DInt, USInt, UInt, UDIInt, Real, LReal, Byte, Word, DWord, Time, Char	Sortie

<sup>1</sup> Les variables d'entrée et la variable de sortie doivent avoir le même type de données.

Tableau 7- 136 Instruction DEMUX (Démultiplexeur)

CONT/LOG	SCL	Description
	<pre> DEMUX (     k:=_unit_in,     in:=variant_in,     out1:=variant_in,     out2:=variant_in,      [...out32:=variant_in,]      outelse:=variant_in); </pre>	<p>DEMUX copie la valeur de l'adresse affectée au paramètre IN dans une sortie parmi plusieurs sorties. La valeur du paramètre K indique la sortie sélectionnée comme destination de la valeur IN. Si la valeur de K est supérieure au nombre (OUT<math>n</math> - 1), la valeur dans IN est copiée dans le paramètre ELSE.</p>

<sup>1</sup> Pour CONT et LOG : Cliquez sur "???" et sélectionnez un type de données dans le menu déroulant.

Pour ajouter une sortie, cliquez sur l'icône de création ou cliquez avec le bouton droit de la souris sur la ligne de sortie de l'un des paramètres OUT existants et sélectionnez la commande "Insérer sortie". Pour supprimer une sortie, cliquez avec le bouton droit de la souris sur la ligne de sortie de l'un des paramètres OUT existants (lorsqu'il y a plus de sorties que les deux sorties d'origine) et sélectionnez la commande "Supprimer".



Pour ajouter une sortie, cliquez sur l'icône de création ou cliquez avec le bouton droit de la souris sur la ligne de sortie de l'un des paramètres OUT existants et sélectionnez la commande "Insérer sortie".

Pour supprimer une sortie, cliquez avec le bouton droit de la souris sur la ligne de sortie de l'un des paramètres OUT existants (lorsqu'il y a plus de sorties que les deux sorties d'origine) et sélectionnez la commande "Supprimer".

Tableau 7- 137 Types de données pour l'instruction DEMUX

Paramètre	Type de données <sup>1</sup>	Description
K	UInt	Valeur de sélecteur : <ul style="list-style-type: none"> <li>• 0 sélectionne OUT1</li> <li>• 1 sélectionne OUT2</li> <li>• n sélectionne OUTn</li> </ul>
IN	SInt, Int, DInt, USInt, UInt, UDIInt, Real, LReal, Byte, Word, DWord, Time, Char	Entrée
OUT0, OUT1, .. OUTn	SInt, Int, DInt, USInt, UInt, UDIInt, Real, LReal, Byte, Word, DWord, Time, Char	Sorties
ELSE	SInt, Int, DInt, USInt, UInt, UDIInt, Real, LReal, Byte, Word, DWord, Time, Char	Sortie de substitution lorsque K est supérieur à (OUTn - 1)

<sup>1</sup> La variable d'entrée et les variables de sortie doivent avoir le même type de données.

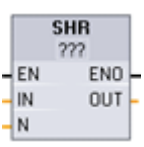
Tableau 7- 138 Etat de ENO pour les instructions MUX et DEMUX

ENO	Situation d'erreur	Résultat OUT
1	Pas d'erreur	MUX : La valeur de IN sélectionnée est copiée dans OUT. DEMUX : La valeur de IN est copiée dans la sortie OUT sélectionnée.
0	MUX : K est supérieur au nombre d'entrées -1	<ul style="list-style-type: none"> <li>• Pas de valeur ELSE fournie : OUT reste inchangé</li> <li>• Valeur ELSE fournie : la valeur de ELSE est affectée à OUT</li> </ul>
	DEMUX : K est supérieur au nombre de sorties -1	<ul style="list-style-type: none"> <li>• Pas de valeur ELSE fournie : les sorties restent inchangées</li> <li>• Valeur ELSE fournie : la valeur de IN est copiée dans ELSE</li> </ul>

## 7.10 Décalage et rotation

### 7.10.1 Instructions de décalage

Tableau 7- 139 Instructions SHR et SHL

CONT/LOG	SCL	Description
	<pre> out := SHR(   in:=_variant_in_,   n:=_uint_in_); out := SHL(   in:=_variant_in_,   n:=_uint_in_); </pre>	<p>Utilisez les instructions de décalage (SHL et SHR) pour décaler le motif binaire du paramètre IN. Le résultat est affecté au paramètre OUT. Le paramètre N indique le nombre de positions de bit à décaler.</p> <ul style="list-style-type: none"> <li>• SHR : Décaler le motif binaire vers la droite</li> <li>• SHL : Décaler le motif binaire vers la gauche</li> </ul>

<sup>1</sup> Pour CONT et LOG : Cliquez sur "???" et sélectionnez les types de données dans le menu déroulant.

Tableau 7- 140 Types de données pour les paramètres

Paramètre	Type de données	Description
IN	Byte, Word, DWord	Motif binaire à décaler
N	UInt	Nombre de positions de bit à décaler
OUT	Byte, Word, DWord	Motif binaire après l'opération de décalage

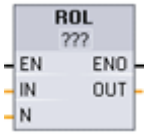
- Pour N=0, aucun décalage n'est effectué. La valeur IN est affectée à OUT.
- Des zéros sont insérés dans les positions de bit vidées par l'opération de décalage.
- Si le nombre de positions à décaler (N) dépasse le nombre de bits dans la valeur cible (8 pour Byte, 16 pour Word, 32 pour DWord), toutes les valeurs de bit d'origine sont décalées et remplacées par des zéros (zéro est affecté à OUT).
- ENO est toujours VRAI pour les instructions de décalage.

Tableau 7- 141 Exemple SHL pour des données de type Word

Décaler les bits d'un mot vers la gauche en insérant des zéros par la droite (N = 1)			
IN	1110 0010 1010 1101	Valeur OUT avant le premier décalage :	1110 0010 1010 1101
		Après le premier décalage vers la gauche :	1100 0101 0101 1010
		Après le deuxième décalage vers la gauche :	1000 1010 1011 0100
		Après le troisième décalage vers la gauche :	0001 0101 0110 1000

### 7.10.2 Instructions de rotation

Tableau 7- 142 Instructions ROR et ROL

CONT/LOG	SCL	Description
	<pre> out := ROL(     in:=_variant_in_,     n:=_uint_in); out := ROR(     in:=_variant_in_,     n:=_uint_in);                     </pre>	Utilisez les instructions de rotation (ROR et ROL) pour opérer une rotation sur le motif binaire du paramètre IN. Le résultat est affecté au paramètre OUT. Le paramètre N définit le nombre de positions de bit à faire tourner. <ul style="list-style-type: none"> <li>• ROR : Rotation du motif binaire vers la droite</li> <li>• ROL : Rotation du motif binaire vers la gauche</li> </ul>

<sup>1</sup> Pour CONT et LOG : Cliquez sur "???" et sélectionnez les types de données dans le menu déroulant.

Tableau 7- 143 Types de données pour les paramètres

Paramètre	Type de données	Description
IN	Byte, Word, DWord	Motif binaire devant faire l'objet de la rotation
N	UInt	Nombre de positions de bit à faire tourner
OUT	Byte, Word, DWord	Motif binaire après l'opération de rotation

- Pour N=0, aucune rotation n'est effectuée. La valeur IN est affectée à OUT.
- Les données binaires objet de la rotation qui sortent d'un côté de la valeur cible sont insérées de l'autre côté de la valeur cible de sorte qu'aucune valeur binaire d'origine n'est perdue.
- Si le nombre de positions de bit à faire tourner (N) dépasse le nombre de bits dans la valeur cible (8 pour Byte, 16 pour Word, 32 pour DWord), la rotation est quand même exécutée.
- ENO est toujours VRAI après l'exécution des instructions de rotation.

Tableau 7- 144 Exemple ROR pour des données de type Word

Rotation de bits sortant par la droite et entrant par la gauche (N = 1)			
IN	0100 0000 0000 0001	Valeur OUT avant la première rotation :	0100 0000 0000 0001
		Après la première rotation vers la droite :	1010 0000 0000 0000
		Après la deuxième rotation vers la droite :	0101 0000 0000 0000

## Instructions avancées

### 8.1 Date et heure

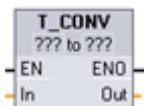
#### 8.1.1 Instructions sur la date et l'heure

Utilisez les instructions sur la date et l'heure pour programmer des calculs sur le calendrier et l'heure.

- T\_CONV convertit le type de données d'une valeur de temps : (Time en DInt) ou (DInt en Time)
- T\_ADD additionne des valeurs Time et DTL : (Time + Time = Time) ou (DTL + Time = DTL)
- T\_SUB soustrait des valeurs Time et DTL : (Time - Time = Time) ou (DTL - Time = DTL)
- T\_DIFF fournit la différence entre deux valeurs DTL en tant que valeur Time : DTL - DTL = Time
- T\_COMBINE combine une valeur Date et une valeur Time\_and\_Date pour créer une valeur DTL.

Pour plus d'informations sur la structure des données DTL et Time, reportez-vous au paragraphe sur les types de données "date et heure" (Page 103).

Tableau 8- 1 Instruction T\_CONV (Convertir temps)

CONT/LOG	SCL	Description
	<pre>out := T_CONV( in:=_variant_in);</pre>	T_CONV convertit un type de données Time en type de données DInt ou, à l'inverse, un type de données DInt en type de données Time.


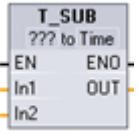
<sup>1</sup> Pour CONT et LOG : Cliquez sur "???" et sélectionnez les types de données dans le menu déroulant.

Tableau 8- 2 Types de données pour les paramètres de T\_CONV

Paramètre et type	Type de données	Description
IN	DInt, Time	Valeur d'entrée Time ou DInt
OUT	DInt, Time	Valeur convertie DInt ou Time

8.1 Date et heure

Tableau 8- 3 Instructions T\_ADD (Additionner temps) et T\_SUB (Soustraire temps)

CONT/LOG	SCL	Description
	<pre>out := T_ADD(   in1:=_variant_in,   in2:=_time_in);</pre>	<p>T_ADD additionne la valeur d'entrée IN1 (de type de données DTL ou Time) à la valeur d'entrée IN2 de type Time. Le paramètre OUT fournit le résultat de type DTL ou Time. Deux opérations sont possibles selon les types de données :</p> <ul style="list-style-type: none"> <li>• Time + Time = Time</li> <li>• DTL + Time = DTL</li> </ul>
	<pre>out := T_SUB(   in1:=_variant_in,   in2:=_time_in);</pre>	<p>T_SUB soustrait la valeur IN2 de type Time de la valeur IN1 (de type DTL ou Time). Le paramètre OUT fournit la valeur de différence en tant que type de données DTL ou Time. Deux opérations sont possibles selon les types de données :</p> <ul style="list-style-type: none"> <li>• Time - Time = Time</li> <li>• DTL - Time = DTL</li> </ul>

1 Pour CONT et LOG : Cliquez sur "???" et sélectionnez les types de données dans le menu déroulant.

Tableau 8- 4 Types de données pour les paramètres de T\_ADD et T\_SUB

Paramètre et type	Type de données	Description
IN1 <sup>1</sup>	IN	DTL, Time
IN2	IN	Time
OUT	OUT	DTL, Time

1 Sélectionnez le type de données de IN1 dans la liste déroulante disponible sous le nom de l'instruction. La sélection du type de données de IN1 définit également le type de données du paramètre OUT.

Tableau 8- 5 Instruction T\_DIFF (Différence de temps)


CONT/LOG	SCL	Description
	<pre>out := T_DIFF(   in1:=_DTL_in,   in2:=_DTL_in);</pre>	<p>T_DIFF soustrait la valeur DTL (IN2) de la valeur DTL (IN1). Le paramètre OUT fournit la valeur de différence en tant que type de données Time.</p> <ul style="list-style-type: none"> <li>• DTL - DTL = Time</li> </ul>

Tableau 8- 6 Types de données pour les paramètres de T\_DIFF

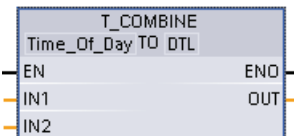
Paramètre et type	Type de données	Description
IN1	IN	DTL
IN2	IN	DTL
OUT	OUT	Time



**Codes d'erreur :** ENO = 1 signifie qu'aucune erreur ne s'est produite. Erreurs si ENO = 0 et le paramètre OUT = 0 :

- Valeur DTL invalide
- Valeur Time invalide

Tableau 8- 7 Instruction T\_COMBINE (Combiner valeurs de temps)

CONT/LOG	SCL	Description
	<pre>out := CONCAT_DATE_TOD (   In1 := _date_in,   In2 := _tod_in);</pre>	T_COMBINE combine une valeur Date et une valeur Time_of_Day pour créer une valeur DTL.

<sup>1</sup> Notez que l'instruction avancée T\_COMBINE correspond à la fonction CONCAT\_DATE\_TOD dans SCL.


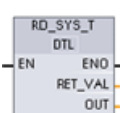
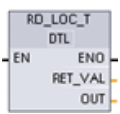
Tableau 8- 8 Types de données pour les paramètres de T\_COMBINE

Paramètre et type		Type de données	Description
IN1	IN	Date	La valeur Date à combiner doit être comprise entre DATE#1990-01-01 et DATE#2089-12-31.
IN2	IN	Time_of_Day	Valeur Time_of_Day à combiner
OUT	OUT	DTL	Valeur DTL

### 8.1.2 Régler et lire l'horloge système

Utilisez les instructions d'horloge pour régler et lire l'horloge système de la CPU. Le type de données DTL (Page 103) sert à fournir des valeurs de date et d'heure.

Tableau 8- 9 Instructions pour l'heure système

CONT/LOG	SCL	Description
	<pre>ret_val := WR_SYS_T(   in:=_DTL_in);</pre>	WR_SYS_T (Ecrire l'heure système) règle l'horloge temps réel de la CPU à l'aide de la valeur DTL dans le paramètre IN. Cette valeur de temps n'inclut pas les décalages pour le fuseau horaire local et l'heure d'été.
	<pre>ret_val := RD_SYS_T(   out=&gt;_DTL_out);</pre>	RD_SYS_T (Lire l'heure système) lit l'heure système en cours dans la CPU. Cette valeur de temps n'inclut pas les décalages pour le fuseau horaire local et l'heure d'été.
	<pre>ret_val := RD_LOC_T(   out=&gt;_DTL_out);</pre>	RD_LOC_T (Lire l'heure locale) fournit l'heure locale en cours de la CPU sous forme de type de données DTL. Cette valeur correspond au fuseau horaire local avec les ajustements appropriés pour l'heure d'été (si configurée).

8.1 Date et heure

Tableau 8- 10 Types de données pour les paramètres

Paramètre et type		Type de données	Description
IN	IN	DTL	Heure à régler dans l'horloge système de la CPU
RET_VAL	OUT	Int	Code d'erreur d'exécution
OUT	OUT	DTL	RD_SYS_T : Heure système en cours de la CPU RD_LOC_T : Heure locale en cours incluant tout ajustement pour l'heure d'été (si configurée)

- L'heure locale se calcule à l'aide des décalages de fuseau horaire et d'heure d'été que vous avez paramétrés dans l'onglet général "Heure" de la configuration d'appareil.
- La configuration du fuseau horaire constitue un décalage par rapport à l'heure UTC ou GMT.
- La configuration de l'heure d'été précise le mois, la semaine, le jour et l'heure auxquels l'heure d'été entre en vigueur.
- La configuration de l'heure standard précise également le mois, la semaine, le jour et l'heure auxquels l'heure standard entre en vigueur.
- Le décalage de fuseau horaire est toujours appliqué à la valeur de l'heure système. Le décalage d'heure d'été n'est appliqué que lorsque l'heure d'été est en vigueur.

**Remarque**

**Configuration du passage à l'heure d'été et à l'heure d'hiver**

La propriété "Heure" pour "Début de l'heure d'été" de la configuration d'appareil de la CPU doit être votre heure locale.

**Codes d'erreur :** ENO = 1 signifie qu'aucune erreur ne s'est produite. ENO = 0 signifie qu'une erreur d'exécution s'est produite et un code d'erreur est fourni par la valeur RET\_VAL.

Tableau 8- 11 Codes d'erreur

RET_VAL (W#16#....)	Description
0000	L'heure locale en cours est l'heure d'hiver.
0001	L'heure d'été a été configurée et l'heure locale en cours est l'heure d'été.
8080	Heure locale non disponible
8081	Valeur de l'année incorrecte
8082	Valeur du mois incorrecte
8083	Valeur du jour incorrecte
8084	Valeur de l'heure incorrecte
8085	Valeur des minutes incorrecte
8086	Valeur des secondes incorrecte
8087	Valeur des nanosecondes incorrecte
80B0	Défaillance de l'horloge temps réel

### 8.1.3 Instruction Compteur d'heures de fonctionnement

Tableau 8- 12 Instruction RTM

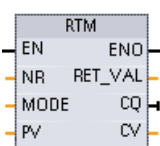
CONT/LOG	SCL	Description
	<pre>RTM(NR:=_uint_in_,     MODE:=_byte_in_,     PV:=_dint_in_,     CQ=&gt;_bool_out_,     CV=&gt;_dint_out_);</pre>	L'instruction RTM (Compteur d'heures de fonctionnement) permet de régler, démarrer, arrêter et lire les compteurs d'heures de fonctionnement dans la CPU.

Tableau 8- 13 Types de données pour les paramètres

Paramètre et type	Type de données	Description
NR	IN	UInt
MODE	IN	Byte
PV	IN	DInt
RET_VAL	OUT	Int
CQ	OUT	Bool
CV	OUT	DInt

La CPU exploite jusqu'à dix compteurs d'heures de fonctionnement pour suivre les heures de fonctionnement des sous-systèmes de commande critiques. Vous devez démarrer les compteurs d'heures individuels avec une instruction RTM pour chaque compteur. Tous les compteurs d'heures de fonctionnement s'arrêtent lorsque la CPU passe de l'état MARCHE à l'état ARRÊT. Vous pouvez également arrêter des compteurs d'heures individuellement à l'aide du mode d'exécution 2 de RTM.

Lorsqu'une CPU passe de l'état ARRÊT à l'état MARCHE, vous devez redémarrer les compteurs d'heures avec une instruction RTM pour chaque compteur démarré. Lorsque la valeur d'un compteur d'heures de fonctionnement est supérieure à 2147483647 heures, le comptage s'arrête et l'erreur "Débordement" est émise. Vous devez exécuter l'instruction RTM une fois pour chaque compteur d'heures que vous voulez réinitialiser ou modifier.

8.1 Date et heure

Une coupure de courant ou une mise hors tension de la CPU déclenche une procédure d'arrêt qui enregistre les valeurs en cours des compteurs d'heures de fonctionnement en mémoire rémanente. A la remise sous tension de la CPU, les valeurs des compteurs d'heures de fonctionnement sauvegardées sont rechargées dans les compteurs d'heures sans perte des totaux d'heures de fonctionnement précédents. Les compteurs d'heures de fonctionnement doivent être redémarrés pour comptabiliser de nouvelles heures de fonctionnement.

Vous pouvez également utiliser le mode d'exécution 7 de RTM dans votre programme pour sauvegarder les valeurs des compteurs d'heures de fonctionnement en carte mémoire. L'état de tous les compteurs d'heures à l'instant où le mode 7 de RTM est exécuté est alors stocké dans la carte mémoire. Ces valeurs enregistrées peuvent devenir incorrectes avec le temps du fait des démarrages et arrêts éventuels des compteurs d'heures pendant une session d'exécution du programme. Vous devez périodiquement actualiser les valeurs en carte mémoire pour capturer les événements de fonctionnement importants. L'avantage de stocker les valeurs RTM en carte mémoire est que vous pouvez insérer la carte mémoire dans une CPU de remplacement où votre programme est les valeurs RTM sauvegardées seront alors disponibles. Si vous n'aviez pas sauvegardé les valeurs RTM dans la carte mémoire, elles seraient perdues (dans une CPU de remplacement).

**Remarque**

**Évitez les appels excessifs d'opérations d'écriture en carte mémoire dans votre programme**

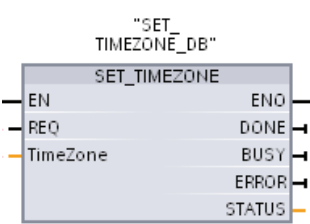
Réduisez les opérations d'écriture en carte mémoire flash au minimum pour allonger la durée de vie de la carte mémoire.

Tableau 8- 14 Codes d'erreur

RET_VAL (W#16#....)	Description
0	Pas d'erreur
8080	Numéro de compteur d'heures de fonctionnement incorrect
8081	Une valeur négative a été transmise au paramètre PV.
8082	Débordement du compteur d'heures de fonctionnement
8091	Le paramètre d'entrée MODE contient une valeur incorrecte.
80B1	Impossible de sauvegarder les valeurs en carte mémoire (MODE = 7)

## 8.1.4 Instruction SET\_TIMEZONE

Tableau 8- 15 Instruction SET\_TIMEZONE

CONT/LOG	SCL	Description
 <p>The diagram shows a block labeled "SET_TIMEZONE" with the following connections:      - EN (input) connected to the top left.      - REQ (input) connected to the top left.      - Timezone (input) connected to the top left.      - END (output) connected to the top right.      - DONE (output) connected to the top right.      - BUSY (output) connected to the top right.      - ERROR (output) connected to the top right.      - STATUS (output) connected to the bottom right.</p>	<pre>"SET_TIMEZONE_DB" (     REQ:=_bool_in,     Timezone:=_struct_in,     DONE=&gt;_bool_out_,     BUSY=&gt;_bool_out_,     ERROR=&gt;_bool_out_,     STATUS=&gt;_word_out_);</pre>	Définit les paramètres de fuseau horaire local et d'heure d'été utilisés pour convertir l'heure système CPU en heure locale.

<sup>1</sup> Dans l'exemple SCL, "SET\_TIMEZONE\_DB" est le nom du DB d'instance.

Tableau 8- 16 Types de données pour les paramètres

Paramètre et type		Type de données	Description
REQ	IN	Bool	REQ=1 : la fonction est exécutée
Timezone	IN	TimeTransformationRule	Règles pour la conversion de l'heure système en heure locale
DONE	OUT	Bool	Fonction achevée
BUSY	OUT	Bool	Fonction occupée
ERROR	OUT	Bool	Erreur détectée
STATUS	OUT	Word	Résultat de la fonction / message d'erreur

Pour configurer manuellement les paramètres de fuseau horaire pour la CPU, utilisez les propriétés "Heure" de l'onglet "Général" de la configuration d'appareil.

Utilisez l'instruction SET\_TIMEZONE pour configurer l'heure locale par programme. Les paramètres de la structure "TimeTransformationRule" indiquent le fuseau horaire local et les informations de temps pour la commutation automatique entre heure d'hiver et heure d'été.

Tableau 8- 17 "Structure "TimeTransformationRule

Paramètre	Type de données	Description
Bias	Int	Différence de temps entre l'heure UTC et l'heure locale [min]
DaylightBias	Int	Différence de temps entre l'heure d'hiver et l'heure d'été [min]
DaylightStartMonth	USInt	Mois de passage à l'heure d'été
DaylightStartWeek	USInt	Semaine de passage à l'heure d'été : <ul style="list-style-type: none"> <li>• 1 = Première occurrence du jour de la semaine dans le mois</li> <li>• ...</li> <li>• 5 = Dernière occurrence du jour de la semaine dans le mois</li> </ul>

8.2 Chaînes de caractères et caractères

Paramètre	Type de données	Description
DaylightStartWeekday	USInt	Jour de passage à l'heure d'été : <ul style="list-style-type: none"> <li>• 1 = dimanche</li> <li>• ...</li> <li>• 7 = samedi</li> </ul>
DaylightStartHour	USInt	Heure de passage à l'heure d'été
StandardStartMonth	USInt	Mois de passage à l'heure d'hiver
StandardStartWeek	USInt	Semaine de passage à l'heure d'hiver : <ul style="list-style-type: none"> <li>• 1 = Première occurrence du jour de la semaine dans le mois</li> <li>• ...</li> <li>• 5 = Dernière occurrence du jour de la semaine dans le mois</li> </ul>
StandardStartWeekday	USInt	Jour de passage à l'heure d'hiver : <ul style="list-style-type: none"> <li>• 1 = dimanche</li> <li>• ...</li> <li>• 7 = samedi</li> </ul>
StandardStartHour	USInt	Heure de passage à l'heure d'hiver
Time Zone Name	STRING [80]	Nom du fuseau : (GMT +01:00) Amsterdam, Berlin, Berne, Rome, Stockholm, Vienne

## 8.2 Chaînes de caractères et caractères

### 8.2.1 Présentation des données chaînes de caractères (type String)

#### Type de données STRING

Les données chaînes de caractères (de type String) sont stockées sous forme d'un en-tête de deux octets suivi de 254 octets au maximum de codes de caractères ASCII. Un en-tête de chaîne de caractères contient deux longueurs. Le premier octet correspond à la longueur maximale qui est indiquée entre crochets lorsque vous initialisez une chaîne (254 par défaut). Le deuxième octet de l'en-tête est la longueur en cours qui est le nombre de caractères valides dans la chaîne. La longueur en cours doit être inférieure ou égale à la longueur maximale. Le nombre d'octets occupés par le format String est supérieur de 2 octets à la longueur maximale.

#### Initialisation des données String

Les données d'entrée et de sortie String doivent être initialisées en tant que chaînes valides en mémoire avant exécution de toute instruction sur chaîne.

## Données String valides

Une chaîne valide a une longueur maximale qui doit être supérieure à zéro mais inférieure à 255. La longueur en cours doit être inférieure ou égale à la longueur maximale.

Il n'est pas possible d'affecter des chaînes aux zones de mémoire I ou Q.

Pour plus d'informations, reportez-vous à Format du type de données String (Page 105).

## 8.2.2 Instruction S\_MOVE

Tableau 8- 18 Instruction Copier chaîne

CONT/LOG	SCL	Description
	<code>out := in;</code>	Copier la chaîne de caractères source IN dans la destination OUT. L'exécution de S_MOVE n'affecte pas le contenu de la chaîne source.

Tableau 8- 19 Types de données pour les paramètres

Paramètre	Type de données	Description
IN	String	Chaîne de caractères source
OUT	String	Adresse cible

Si la longueur effective de la chaîne dans l'entrée IN dépasse la longueur maximale d'une chaîne stockée dans la sortie OUT, seule la partie de la chaîne IN qui tient dans la chaîne OUT est copiée.

## 8.2.3 Instructions de conversion de chaîne

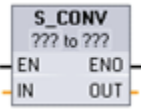
### 8.2.3.1 Conversions de chaîne en valeur et de valeur en chaîne

Vous pouvez convertir des chaînes de caractères numériques en valeurs numériques ou des valeurs numériques en chaînes de caractères numériques à l'aide des instructions suivantes :

- S\_CONV convertit une chaîne numérique en valeur numérique ou bien une valeur numérique en chaîne numérique.
- STRG\_VAL convertit une chaîne numérique en valeur numérique avec des options de format.
- VAL\_STRG convertit une valeur numérique en chaîne numérique avec des options de format.

### S\_CONV (conversions chaîne en valeur)

Tableau 8- 20 Instruction de conversion de chaîne

CONT/LOG	SCL	Description
	<pre>out := &lt;Type&gt;_TO_&lt;Type&gt;(in) ;</pre>	Convertit une chaîne de caractères en sa valeur correspondante ou une valeur en sa chaîne de caractères correspondante. L'instruction S_CONV ne comporte pas d'options de formatage de la sortie. Cela rend S_CONV plus simple à utiliser mais moins souple que les instructions STRG_VAL et VAL_STRG.

- 1 Pour CONT / LOG : Cliquez sur "???" et sélectionnez le type de données dans la liste déroulante.
- 2 Pour SCL : Sélectionnez S\_CONV dans les instructions avancées et répondez aux messages concernant les types de données pour la conversion. STEP 7 propose alors l'instruction de conversion appropriée.

Tableau 8- 21 Types de données (chaîne en valeur)

Paramètre et type		Type de données	Description
IN	IN	String	Chaîne de caractères d'entrée
OUT	OUT	String, Char, SInt, Int, DInt, USInt, UInt, UDIInt, Real, LReal	Valeur numérique de sortie

La conversion du paramètre chaîne IN commence au premier caractère et continue jusqu'à la fin de la chaîne ou jusqu'au premier caractère rencontré qui n'est pas "0" à "9", "+", "-" ou ".". La valeur résultante est fournie à l'adresse indiquée dans le paramètre OUT. Si la valeur numérique en sortie ne tient pas dans la plage du type de données de OUT, le paramètre OUT est mis à 0 et ENO est défini à FAUX. Sinon, le paramètre OUT contient un résultat valide et ENO est défini à VRAI.

Règles de format concernant la chaîne d'entrée :

- Si un séparateur décimal est utilisé dans la chaîne IN, vous devez utiliser le caractère ".".
- Les virgules "," utilisées comme séparateurs des milliers à gauche du séparateur décimal sont autorisées mais ne sont pas prises en compte.
- Les espaces en tête ne sont pas pris en compte.

### S\_CONV (conversions valeur en chaîne)

Tableau 8- 22 Types de données (valeur en chaîne)

Paramètre et type		Type de données	Description
IN	IN	String, Char, SInt, Int, DInt, USInt, UInt, UDIInt, Real, LReal	Valeur numérique d'entrée
OUT	OUT	String	Chaîne de caractères de sortie



Un paramètre IN entier, entier non signé ou nombre à virgule flottante est converti en sa chaîne de caractères correspondante dans OUT. Le paramètre OUT doit référencer une chaîne valide pour que la conversion soit exécutée. Une chaîne valide comprend la longueur de chaîne maximale dans le premier octet, la longueur de chaîne en cours dans le deuxième octet et les caractères de la chaîne en cours dans les octets suivants. La chaîne convertie remplace les caractères dans la chaîne OUT en commençant au premier caractère et l'octet de longueur en cours de la chaîne OUT est ajusté. L'octet de longueur maximale de la chaîne OUT n'est pas modifié.

Le nombre de caractères remplacés dépend du type de données et de la valeur numérique du paramètre IN. Le nombre de caractères remplacés doit tenir dans la longueur de la chaîne du paramètre OUT. La longueur de chaîne maximale (premier octet) de la chaîne OUT doit être supérieure ou égale au nombre maximum attendu de caractères convertis. Le tableau suivant montre les longueurs de chaîne maximales requises pour chaque type de données pris en charge.

Tableau 8- 23 Longueurs maximales des chaînes de caractères pour chaque type de données


Type de données de IN	Nombre maximum de caractères convertis dans la chaîne OUT	Exemple	Longueur de chaîne totale incluant les octets de longueur maximale et en cours
USInt	3	255	5
SInt	4	-128	6
UInt	5	65535	7
Int	6	-32768	8
UDInt	10	4294967295	12
DInt	11	-2147483648	13

Règles de format concernant la chaîne de sortie :

- Les valeurs écrites dans le paramètre OUT n'utilisent pas de signe "+" en tête.
- La représentation en virgule fixe est utilisée (et non la notation exponentielle).
- Le point "." est utilisé pour représenter le séparateur décimal lorsque le paramètre IN est de type de données Real.

### Instruction STRG\_VAL

Tableau 8- 24 Instruction Chaîne en valeur

CONT/LOG	SCL	Description
	<pre>"STRG_VAL" (   in:=_string_in,   format:=_word_in,   p:=uint_in,   out=&gt;_variant_out);</pre>	Convertit une chaîne de caractères numérique en sa représentation entière ou virgule flottante correspondante.

1 Pour CONT / LOG : Cliquez sur "???" et sélectionnez le type de données dans la liste déroulante.

8.2 Chaînes de caractères et caractères

Tableau 8- 25 Types de données pour l'instruction STRG\_VAL

Paramètre et type		Type de données	Description
IN	IN	String	Chaîne de caractères ASCII à convertir
FORMAT	IN	Word	Options de format de sortie
P	IN	UInt, Byte, USInt	IN : Indice pointant sur le premier caractère à convertir (premier caractère = 1)
OUT	OUT	SInt, Int, DInt, USInt, UInt, UDIInt, Real, LReal	Valeur numérique convertie

La conversion commence dans la chaîne IN au décalage de caractère P et continue jusqu'à la fin de la chaîne ou jusqu'au premier caractère rencontré qui n'est pas "+", "-", ".", ",", "e", "E" ou "0" à "9". Le résultat est placé à l'adresse indiquée dans le paramètre OUT.

Les données String doivent être initialisées avant l'exécution en tant que chaîne valide en mémoire.

Le paramètre FORMAT pour l'instruction STRG\_VAL est défini ci-dessous. Les positions de bit inutilisées doivent être mises à zéro.

Tableau 8- 26 Format de l'instruction STRG\_VAL

<b>Bit 16</b>							<b>Bit 8</b>	<b>Bit 7</b>								<b>Bit 0</b>
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	f	r

- f = format de notation                      1= notation exponentielle
- 0 = notation à virgule fixe
- r = format du séparateur                 1 = "," (virgule)
- décimal                                        0 = "." (point)

Tableau 8- 27 Valeurs du paramètre FORMAT


FORMAT (W#16#)	Format de notation	Représentation du séparateur décimal
0000 (par défaut)	Virgule fixe	" . "
0001		" , "
0002	Exponentielle	" . "
0003		" , "
0004 à FFFF	Valeurs illicites	

Règles pour la conversion STRG\_VAL :

- Si le point "." est utilisé comme séparateur décimal, les virgules "," à gauche du séparateur décimal sont considérées comme caractères de séparation des milliers. Les virgules sont autorisées mais ne sont pas prises en compte.
- Si la virgule "," est utilisée comme séparateur décimal, les points "." à gauche du séparateur décimal sont considérés comme caractères de séparation des milliers. Ces points sont autorisés mais ne sont pas pris en compte.
- Les espaces en tête ne sont pas pris en compte.

### Instruction VAL\_STRG

Tableau 8- 28 Instruction Valeur en chaîne

CONT/LOG	SCL	Description
	<pre>"VAL_STRG" (   in:=_variant_in,   size:=_usint_in,   prec:=_usint_in,   format:=_word_in,   p:=uint_in,   out=&gt;_string_out);</pre>	Convertit une valeur entière, entière non signée ou virgule flottante en sa représentation chaîne de caractères correspondante.

1 Pour CONT / LOG : Cliquez sur "???" et sélectionnez le type de données dans la liste déroulante.

Tableau 8- 29 Types de données pour l'instruction VAL\_STRG

Paramètre et type		Type de données	Description
IN	IN	SInt, Int, DInt, USInt, UInt, UDInt, Real, LReal	Valeur à convertir
SIZE	IN	USInt	Nombre de caractères à écrire dans la chaîne OUT
PREC	IN	USInt	Précision ou taille de la partie fractionnaire. Cela n'inclut pas le séparateur décimal.
FORMAT	IN	Word	Options de format de sortie
P	IN	UInt, Byte, USInt	IN : Indice pointant sur le premier caractère de la chaîne OUT à remplacer (premier caractère = 1)
OUT	OUT	String	Chaîne convertie

La valeur représentée par le paramètre IN est convertie en une chaîne référencée par le paramètre OUT. Le paramètre OUT doit être une chaîne valide pour que la conversion soit exécutée.

La chaîne convertie remplacera les caractères dans la chaîne OUT en commençant au décalage de caractères P et en allant jusqu'au nombre de caractères indiqué par le paramètre SIZE. Le nombre de caractères dans SIZE doit tenir dans la longueur de la chaîne OUT, le comptage commençant à la position de caractère P. Cette instruction est utile pour insérer des nombres dans une chaîne de texte. Vous pouvez, par exemple, insérer le nombre "120" dans la chaîne "Pression pompe = 120 psi".

8.2 Chaînes de caractères et caractères

Le paramètre PREC indique la précision ou le nombre de chiffres pour la partie fractionnaire de la chaîne. Si la valeur du paramètre IN est un entier, PREC indique la position du séparateur décimal. Par exemple, si la valeur de données est 123 et que PREC est égal à 1, le résultat sera "12.3". La précision maximale acceptée pour le type de données Real est de 7 chiffres.

Si le paramètre P est supérieur à la taille en cours de la chaîne OUT, des espaces sont ajoutés jusqu'à la position P et le résultat est ajouté à la fin de la chaîne. La conversion s'arrête si la longueur maximale de la chaîne OUT est atteinte.

Le paramètre FORMAT pour l'instruction VAL\_STRG est défini ci-dessous. Les positions de bit inutilisées doivent être mises à zéro.

Tableau 8- 30 Format de l'instruction VAL\_STRG

<b>Bit 16</b>							<b>Bit 8</b>	<b>Bit 7</b>							<b>Bit 0</b>
0	0	0	0	0	0	0	0	0	0	0	0	0	s	f	r

- |                                  |                                                                                                 |
|----------------------------------|-------------------------------------------------------------------------------------------------|
| s = signe du nombre              | 1= utiliser les caractères "+" et "-" pour le signe<br>0 = utiliser le caractère "-" uniquement |
| f = format de notation           | 1= notation exponentielle<br>0 = notation à virgule fixe                                        |
| r = format du séparateur décimal | 1 = "," (virgule)<br>0 = "." (point)                                                            |

Tableau 8- 31 Valeurs du paramètre FORMAT

FORMAT (WORD)	Signe du nombre	Format de notation	Représentation du séparateur décimal
W#16#0000	"-" uniquement	Virgule fixe	"."
W#16#0001			" "
W#16#0002		Exponentielle	"."
W#16#0003	" "		
W#16#0004	"+" et "-"	Virgule fixe	"."
W#16#0005			" "
W#16#0006		Exponentielle	"."
W#16#0007			" "
W#16#0008 à W#16#FFFF	Valeurs illicites		

Règles de format concernant la chaîne OUT :

- Des espaces sont ajoutés à l'extrême gauche de la chaîne lorsque la chaîne convertie est inférieure à la taille précisée.
- Lorsque le bit de signe du paramètre FORMAT est FAUX, les valeurs de type de données entier non signé et entier signé sont écrites dans la mémoire tampon de sortie sans être précédées du signe "+". Le signe "-" est utilisé si nécessaire.  
<espaces en tête><chiffres sans zéros en tête>'.'- Lorsque le bit de signe est VRAI, les valeurs de type de données entier non signé et entier signé sont toujours écrites dans la mémoire tampon de sortie avec un caractère de signe en tête.  
<espaces en tête><signe><chiffres sans zéros en tête>'.'- Lorsque la notation exponentielle est paramétrée dans FORMAT, les valeurs de type de données Real sont écrites comme suit dans la mémoire tampon de sortie :  
<espaces en tête><signe><chiffre> '.' <chiffres PREC>'E' <signe><chiffres sans zéro en tête>
- Lorsque la notation en virgule fixe est paramétrée dans FORMAT, les valeurs de type de données entier, entier non signé et réel sont écrites comme suit dans la mémoire tampon de sortie :  
<espaces en tête><signe><chiffres sans zéros en tête>'.'- Les zéros en tête à gauche du séparateur décimal sont supprimés (excepté le chiffre à côté du séparateur décimal).
- Les valeurs à droite du séparateur décimal sont arrondies pour correspondre au nombre de chiffres à droite du séparateur décimal indiqué par le paramètre PREC.
- La taille de la chaîne de sortie doit être au minimum de trois octets supérieure au nombre de chiffres à droite du séparateur décimal.
- Les valeurs sont alignées à droite dans la chaîne de sortie.

### Situations d'erreur signalées par ENO

Lorsqu'une erreur se produit pendant l'opération de conversion, les résultats suivants sont renvoyés :

- ENO est mis à 0.
- OUT est mis à 0 ou défini comme montré dans les exemples de conversion de chaîne en valeur.
- OUT reste inchangé ou est défini comme montré dans les exemples où OUT est une chaîne.

Tableau 8- 32 Etat de ENO

ENO	Description
1	Pas d'erreur
0	Paramètre illicite ou invalide ; par exemple, accès à un DB qui n'existe pas.
0	Chaîne illicite : la longueur maximale de la chaîne est 0 ou 255.

8.2 Chaînes de caractères et caractères

ENO	Description
0	Chaîne illicite : la longueur en cours est supérieure à la longueur maximale.
0	La valeur numérique convertie est trop grande pour le type de données spécifié pour OUT.
0	La taille de chaîne maximale du paramètre OUT doit être suffisamment grande pour recevoir le nombre de caractères indiqué par le paramètre SIZE, en commençant à la position de caractère indiquée par le paramètre P.
0	Valeur P illicite : P = 0 ou P est supérieur à la longueur de chaîne en cours.
0	Le paramètre SIZE doit être supérieur au paramètre PREC.

Tableau 8- 33 Exemples de conversion de chaîne en valeur S\_CONV

Chaîne IN	Type de données de OUT	Valeur OUT	ENO
"123"	Int ou DInt	123	VRAI
"-00456"	Int ou DInt	-456	VRAI
"123.45"	Int ou DInt	123	VRAI
"+2345"	Int ou DInt	2345	VRAI
"00123AB"	Int ou DInt	123	VRAI
"123"	Real	123.0	VRAI
"123.45"	Real	123.45	VRAI
"1.23e-4"	Real	1.23	VRAI
"1.23E-4"	Real	1.23	VRAI
"12,345.67"	Real	12345.67	VRAI
"3.4e39"	Real	3.4	VRAI
"-3.4e39"	Real	-3.4	VRAI
"1.17549e-38"	Real	1.17549	VRAI
"12345"	SInt	0	FAUX
"A123"	Sans objet	0	FAUX
""	Sans objet	0	FAUX
"++123"	Sans objet	0	FAUX
"+-123"	Sans objet	0	FAUX

Tableau 8- 34 Exemples de conversion de valeur en chaîne S\_CONV

Type de données	Valeur IN	Chaîne OUT	ENO
UInt	123	"123"	VRAI
UInt	0	"0"	VRAI
UDInt	12345678	"12345678"	VRAI
Real	-INF	"INF"	FAUX
Real	+INF	"INF"	FAUX
Real	NaN	"NaN"	FAUX

Tableau 8- 35 Exemples de conversion STRG\_VAL

Chaîne IN	FORMAT (W#16#....)	Type de données de OUT	Valeur OUT	ENO
"123"	0000	Int ou DInt	123	VRAI
"-00456"	0000	Int ou DInt	-456	VRAI
"123.45"	0000	Int ou DInt	123	VRAI
"+2345"	0000	Int ou DInt	2345	VRAI
"00123AB"	0000	Int ou DInt	123	VRAI
"123"	0000	Real	123.0	VRAI
"-00456"	0001	Real	-456.0	VRAI
"+00456"	0001	Real	456.0	VRAI
"123.45"	0000	Real	123.45	VRAI
"123.45"	0001	Real	12345.0	VRAI
"123,45"	0000	Real	12345.0	VRAI
"123,45"	0001	Real	123.45	VRAI
".00123AB"	0001	Real	123.0	VRAI
"1.23e-4"	0000	Real	1.23	VRAI
"1.23E-4"	0000	Real	1.23	VRAI
"1.23E-4"	0002	Real	1.23E-4	VRAI
"12,345.67"	0000	Real	12345.67	VRAI
"12,345.67"	0001	Real	12.345	VRAI
"3.4e39"	0002	Real	+INF	VRAI
"-3.4e39"	0002	Real	-INF	VRAI
"1.1754943e-38" (et inférieur)	0002	Real	0.0	VRAI
"12345"	Sans objet	SInt	0	FAUX
"A123"	Sans objet	Sans objet	0	FAUX
""	Sans objet	Sans objet	0	FAUX
"++123"	Sans objet	Sans objet	0	FAUX
"+-123"	Sans objet	Sans objet	0	FAUX

Les exemples suivants de conversions VAL\_STRG sont basés sur une chaîne OUT initialisée comme suit :

"Current Temp = xxxxxxxxxxx C"

Le caractère "x" représente des espaces alloués pour la valeur convertie.

Tableau 8- 36 Exemples de conversion VAL\_STRG

Type de données	Valeur IN	P	SIZE	FORMAT (W#16#....)	PREC	Chaîne OUT	ENO
UInt	123	16	10	0000	0	Current Temp = xxxxxxxx123 C	VRAI
UInt	0	16	10	0000	2	Current Temp = xxxxxx0.00 C	VRAI
UDInt	12345678	16	10	0000	3	Current Temp = x12345.678 C	VRAI
UDInt	12345678	16	10	0001	3	Current Temp = x12345,678 C	VRAI

Type de données	Valeur IN	P	SIZE	FORMAT (W#16#...)	PREC	Chaîne OUT	ENO
Int	123	16	10	0004	0	Current Temp = xxxxxx+123 C	VRAI
Int	-123	16	10	0004	0	Current Temp = xxxxxx-123 C	VRAI
Real	-0.00123	16	10	0004	4	Current Temp = xxx-0.0012 C	VRAI
Real	-0.00123	16	10	0006	4	Current Temp = -1.2300E-3 C	VRAI
Real	-INF	16	10	Sans objet	4	Current Temp = xxxxxx-INF C	FAUX
Real	+INF	16	10	Sans objet	4	Current Temp = xxxxxx+INF C	FAUX
Real	NaN	16	10	Sans objet	4	Current Temp = xxxxxxNaN C	FAUX
UDInt	12345678	16	6	Sans objet	3	Current Temp = xxxxxxxxxxxx C	FAUX

### 8.2.3.2 Conversions de chaîne en caractères et de caractères en chaîne

Chars\_TO\_Strg copie un tableau d'octets de caractères ASCII dans une chaîne de caractères.

Strg\_TO\_Chars copie une chaîne de caractères ASCII dans un tableau d'octets de caractères.

---

#### Remarque

Seuls les types de tableaux de base zéro (Array [0..n] of Char ou Array [0..n] of Byte) sont autorisés comme paramètre d'entrée Chars pour l'instruction Chars\_TO\_Strg ou comme paramètre d'entrée/sortie Chars pour l'instruction Strg\_TO\_Chars .

---

Tableau 8- 37 Instruction Chars\_TO\_Strg

CONT/LOG	SCL	Description
	<pre>Chars_TO_Strg(   Chars:=_variant_in_,   pChars:=_dint_in_,   Cnt:=_uint_in_,   Strg=&gt;_string_out_);</pre>	<p>L'ensemble ou une partie d'un tableau de caractères est copié dans une chaîne.</p> <p>La chaîne de sortie doit être déclarée avant l'exécution de Chars_TO_Strg. La chaîne est alors écrasée par l'exécution de l'instruction Chars_TO_Strg.</p> <p>Il est possible d'utiliser des chaînes de toutes les longueurs maximales acceptées (1..254).</p> <p>La longueur de chaîne maximale n'est pas modifiée par l'instruction Chars_TO_Strg . La copie du tableau dans la chaîne s'arrête lorsque la longueur de chaîne maximale est atteinte.</p> <p>Une valeur de caractère nul (\$00) ou 16#00 dans le tableau de caractères constitue un délimiteur qui met fin à la copie de caractères dans la chaîne.</p>



Tableau 8- 38 Types de données pour les paramètres (Chars\_TO\_Strg)

Paramètre et type		Type de données	Description
Chars	IN	Variant	Le paramètre Chars est un pointeur désignant un tableau de caractères de base zéro [0..n] à convertir en chaîne. Le tableau peut être déclaré dans un DB ou sous forme de variables locales dans l'interface de bloc. Exemple : "DB1".MyArray désigne les éléments de MyArray [0..10] of Char dans le DB1.
pChars	IN	Dint	Numéro d'élément du premier caractère du tableau à copier. L'élément [0] du tableau est pris par défaut.
Cnt	IN	UInt	Nombre de caractères à copier ; 0 signifie tous les caractères.
Strg	OUT	String	Chaîne cible

Tableau 8- 39 Instruction Strg\_TO\_Chars

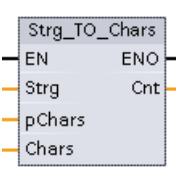
CONT/LOG	SCL	Description
	<pre>Strg_TO_Chars (   Strg:=_string_in_,   pChars:=_dint_in_,   Cnt=&gt;_uint_out_,   Chars:=_variant_inout_);</pre>	<p>La chaîne d'entrée complète Strg est copiée dans un tableau de caractères indiqué par le paramètre IN_OUT Chars.</p> <p>L'opération écrase les octets en commençant à l'élément du tableau indiqué par le paramètre pChars.</p> <p>Il est possible d'utiliser des chaînes de toutes les longueurs maximales acceptées (1..254).</p> <p>Aucun délimiteur de fin n'est écrit ; c'est à vous de vous en charger. Pour placer un délimiteur de fin juste après le dernière caractère écrit dans le tableau, utilisez le numéro d'élément de tableau suivant [pChars+Cnt].</p>

Tableau 8- 40 Types de données pour les paramètres (Strg\_TO\_Chars)

Paramètre et type		Type de données	Description
Strg	IN	String	Chaîne de caractères source
pChars	IN	Dint	Numéro d'élément de tableau pour le premier caractère de la chaîne écrit dans le tableau cible
Chars	IN_OUT	Variant	Le paramètre Chars est un pointeur désignant un tableau de caractères de base zéro [0..n] à copier depuis la chaîne d'entrée. Le tableau peut être déclaré dans un DB ou sous forme de variables locales dans l'interface de bloc. Exemple : "DB1".MyArray désigne les éléments de MyArray [0..10] of Char dans le DB1.
Cnt	OUT	UInt	Nombre de caractères copiés

Tableau 8- 41 Etat de ENO

ENO	Description
1	Pas d'erreur
0	Chars_TO_Strg : Tentative de copier dans la chaîne de sortie plus d'octets de caractères que ne l'autorise l'octet de longueur maximale dans la déclaration de la chaîne
0	Chars_TO_Strg : La valeur de caractère nul (16#00) a été trouvée dans le tableau de caractères d'entrée.
0	Strg_TO_Chars : Tentative de copier dans le tableau de sortie plus d'octets de caractères que ne l'autorise la limite de nombre d'éléments

### 8.2.3.3 Conversions de ASCII en hexadécimal et de hexadécimal en ASCII

Utilisez les instructions ATH (ASCII en hexadécimal) et HTA (Hexadécimal en ASCII) pour effectuer des conversions entre octets de caractères ASCII (caractères 0 à 9 et majuscules A à F uniquement) et les quartets hexadécimaux correspondants.

Tableau 8- 42 Instruction ATH

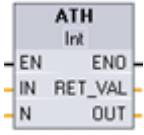
CONT/LOG	SCL	Description
	<pre>ret_val := ATH(     in:=_variant_in_,     n:=_int_in_,     out=&gt;_variant_out_);</pre>	Convertit des caractères ASCII en chiffres hexadécimaux condensés.

Tableau 8- 43 Types de données pour l'instruction ATH

Type de paramètre	Type de données	Description
IN	IN	Variant
N	IN	UInt
RET_VAL	OUT	Word
OUT	OUT	Variant

La conversion commence à l'emplacement indiqué par le paramètre IN et se poursuit sur N octets. Le résultat est placé à l'emplacement indiqué par OUT. Seuls les caractères ASCII valides 0 à 9 et majuscules A à F peuvent être convertis. Tout autre caractère est converti en un zéro.

Les caractères codés ASCII de 8 bits sont convertis en quartets hexadécimaux. Deux caractères ASCII peuvent être stockés dans un octet unique.

Les paramètres IN et OUT correspondent à des tableaux d'octets et non à des données de type String hexadécimales. Les caractères ASCII sont convertis et placés dans la sortie hexadécimale dans l'ordre où ils sont lus. S'il y a un nombre impair de caractères ASCII, le quartet de droite du dernier chiffre hexadécimal converti est complété avec des zéros.

Tableau 8- 44 Exemples de conversion d'ASCII en hexadécimal (ATH)

Octets caractères dans IN	N	Valeur OUT	ENO
'0123'	4	W#16#0123	VRAI
'123AFx1a23'	10	16#123AF01023	FAUX
'a23'	3	W#16#A230	VRAI

Tableau 8- 45 Instruction HTA

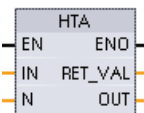
CONT/LOG	SCL	Description
	<pre>ret_val := HTA(     in:=_variant_in_,     n:=_uint_in_,     out=&gt;_variant_out_);</pre>	Convertit des chiffres hexadécimaux condensés en octets de caractères ASCII correspondants.

Tableau 8- 46 Types de données pour l'instruction HTA

Paramètre et type	Type de données	Description	
IN	IN	Variant	Pointeur désignant le tableau d'octets d'entrée
N	IN	UInt	Nombre d'octets à convertir (chaque octet d'entrée comporte deux quartets ce qui produit 2N caractères ASCII)
RET_VAL	OUT	Word	Code d'erreur d'exécution
OUT	OUT	Variant	Pointeur désignant le tableau d'octets de caractères ASCII

La conversion commence à l'emplacement indiqué par le paramètre IN et se poursuit sur N octets. Chaque quartet est converti en un unique caractère ASCII de 8 bits, ce qui produit 2N octets de caractères ASCII de sortie. Les 2N octets de la sortie sont écrits sous forme de caractères ASCII de 0 à 9 et A à F majuscules. Le paramètre OUT correspond à un tableau d'octets et non à une chaîne de caractères.

Les quartets des octets hexadécimaux sont convertis en caractères dans l'ordre où ils sont lus (le quartet de gauche d'un octet hexadécimal est converti le premier, suivi par le quartet de droite de ce même octet).

Tableau 8- 47 Exemples de conversion de hexadécimal en ASCII (HTA)

Valeur IN	N	Octets caractères dans OUT	ENO (ENO est toujours VRAI après l'exécution de HTA)
W#16#0123	2	'0123'	VRAI
DW#16#123AF012	4	'123AF012'	VRAI

8.2 Chaînes de caractères et caractères

Tableau 8- 48 Codes d'erreur de ATH and HTA

RET_VAL (W#16#....)	Description	ENO
0000	Pas d'erreur	VRAI
0007	Caractère d'entrée ATH invalide : Un caractère qui n'était pas un caractère ASCII 0 à 9, a à f minuscule ou A à F majuscule a été trouvé.	FAUX
8101	Pointeur d'entrée illicite ou invalide ; par exemple, accès à un DB qui n'existe pas.	FAUX
8120	La chaîne d'entrée a un format invalide, c'est-à-dire max=0, max=255, valeur en cours>max ou longueur allouée dans le pointeur < max.	FAUX
8182	Mémoire tampon d'entrée trop petite pour N	FAUX
8151	Type de données interdit pour la mémoire tampon d'entrée	FAUX
8301	Pointeur de sortie illicite ou invalide ; par exemple, accès à un DB qui n'existe pas.	FAUX
8320	La chaîne de sortie a un format invalide, c'est-à-dire max=0, max=255, valeur en cours>max ou longueur allouée dans le pointeur < max.	FAUX
8382	Mémoire tampon de sortie trop petite pour N	FAUX
8351	Type de données interdit pour la mémoire tampon de sortie	FAUX

8.2.4 Instructions sur chaîne

Votre programme de commande peut utiliser les instructions sur chaîne et sur caractère suivantes pour créer des messages destinés aux afficheurs et aux journaux du processus.

8.2.4.1 LEN

Tableau 8- 49 Instruction Longueur


CONT/LOG	SCL	Description
	<pre>out := LEN(in);</pre>	LEN (Longueur) transmet la longueur en cours de la chaîne IN à la sortie OUT. Une chaîne vide a une longueur de zéro.

Tableau 8- 50 Types de données pour les paramètres

Paramètre et type	Type de données	Description
IN	IN	String
OUT	OUT	Int, DInt, Real, LReal

Tableau 8- 51 Etat de ENO

ENO	Situation d'erreur	OUT
1	Pas de chaîne invalide	Longueur de chaîne valide
0	La longueur en cours de IN dépasse la longueur maximale de IN.	La longueur en cours est mise à 0.
	La longueur maximale de IN n'entre pas dans la zone de mémoire allouée.	
	La longueur maximale de IN est 255 (longueur interdite).	

### 8.2.4.2 CONCAT

Tableau 8- 52 Instruction Concaténer des chaînes

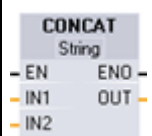
CONT/LOG	SCL	Description
	<pre>out := CONCAT(in1, in2);</pre>	CONCAT (Concaténer des chaînes) combine les chaînes IN1 et IN2 pour former une nouvelle chaîne dans OUT. Après la concaténation, la chaîne IN1 est la partie gauche et la chaîne IN2 la partie droite de la chaîne combinée.

Tableau 8- 53 Types de données pour les paramètres

Paramètre et type	Type de données	Description
IN1	IN	String
IN2	IN	String
OUT	OUT	String

Tableau 8- 54 Etat de ENO

ENO	Situation d'erreur	OUT
1	Pas d'erreur détectée	Caractères valides
0	La chaîne résultante après concaténation est plus grande que la longueur maximale de la chaîne OUT.	Les caractères de la chaîne résultante sont copiés jusqu'à ce que la longueur maximale de OUT soit atteinte.
	La longueur en cours de IN1 dépasse la longueur maximale de IN1, la longueur en cours de IN2 dépasse la longueur maximale de IN2 ou la longueur en cours de OUT dépasse la longueur maximale de OUT (chaîne invalide).	La longueur en cours est mise à 0.
	La longueur maximale de IN1, IN2 ou OUT n'entre pas dans la zone de mémoire allouée.	
	La longueur maximale de IN1 ou IN2 est 255 ou la longueur maximale de OUT est 0 ou 255.	

8.2.4.3 LEFT, RIGHT et MID

Tableau 8- 55 Instructions Lire les caractères de gauche d'une chaîne, Lire les caractères de droite d'une chaîne et Lire les caractères du milieu d'une chaîne

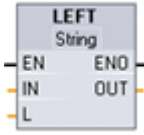
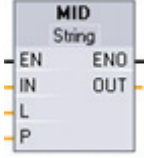
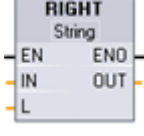
CONT/LOG	SCL	Description
	<pre>out := LEFT(in, L);</pre>	<p>LEFT (Lire les caractères de gauche d'une chaîne) renvoie une sous-chaîne constituée des L premiers caractères du paramètre chaîne IN.</p> <ul style="list-style-type: none"> <li>• Si L est supérieur à la longueur en cours de la chaîne IN, la chaîne IN entière est renvoyée dans OUT.</li> <li>• Si l'entrée est une chaîne vide, une chaîne vide est renvoyée dans OUT.</li> </ul>
	<pre>out := MID(in, L, p);</pre>	<p>MID (Lire les caractères du milieu d'une chaîne) renvoie la partie médiane d'une chaîne. La sous-chaîne médiane fait L caractères de long et commence à la position de caractère P (inclusive).</p> <p>Si la somme de L et de P dépasse la longueur en cours de la chaîne IN, une sous-chaîne commençant à la position de caractère P et allant jusqu'à la fin de la chaîne IN est renvoyée.</p>
	<pre>out := RIGHT(in, L);</pre>	<p>RIGHT (Lire les caractères de droite d'une chaîne) renvoie les L derniers caractères d'une chaîne.</p> <ul style="list-style-type: none"> <li>• Si L est supérieur à la longueur en cours de la chaîne IN, la chaîne IN entière est renvoyée dans le paramètre OUT.</li> <li>• Si l'entrée est une chaîne vide, une chaîne vide est renvoyée dans OUT.</li> </ul>

Tableau 8- 56 Types de données pour les paramètres

Paramètre et type	Type de données	Description
IN	IN	String Chaîne d'entrée
L	IN	Int Longueur de la sous-chaîne à créer : <ul style="list-style-type: none"> <li>• LEFT utilise le nombre de caractères les plus à gauche dans la chaîne.</li> <li>• RIGHT utilise le nombre de caractères les plus à droite dans la chaîne.</li> <li>• MID utilise le nombre de caractères à partir de la position P dans la chaîne.</li> </ul>
P	IN	Int MID uniquement : Position du premier caractère de la sous-chaîne à copier P= 1 pour la position de caractère de début de la chaîne IN
OUT	OUT	String Chaîne de sortie

Tableau 8- 57 Etat de ENO

ENO	Situation d'erreur	OUT
1	Pas d'erreur détectée	Caractères valides
0	<ul style="list-style-type: none"> <li>L ou P est inférieur ou égal à 0.</li> <li>P est supérieur à la longueur maximale de IN.</li> <li>La longueur en cours de IN dépasse la longueur maximale de IN ou la longueur en cours de OUT dépasse la longueur maximale de OUT.</li> <li>La longueur maximale de IN ou OUT n'entre pas dans la zone de mémoire allouée.</li> <li>La longueur maximale de IN ou OUT est 0 ou 255.</li> </ul>	La longueur en cours est mise à 0.
	La longueur (L) de la sous-chaîne à copier est supérieure à la longueur maximale de la chaîne OUT.	Les caractères sont copiés jusqu'à ce que la longueur maximale de OUT soit atteinte.
	MID uniquement : L ou P est inférieur ou égal à 0.	La longueur en cours est mise à 0.
	MID uniquement : P est supérieur à la longueur maximale de IN	
	La longueur en cours de IN1 dépasse la longueur maximale de IN1 ou la longueur en cours de IN2 dépasse la longueur maximale de IN2 (chaîne invalide).	La longueur en cours est mise à 0.
	La longueur maximale de IN1, IN2 ou OUT n'entre pas dans la zone de mémoire allouée.	
La longueur maximale de IN1, IN2 ou OUT est 0 ou 255 (longueur interdite).		

### 8.2.4.4 DELETE

Tableau 8- 58 Instruction Supprimer des caractères dans une chaîne

CONT/LOG	SCL	Description
	<pre>out := DELETE(in, L, p);</pre>	<p>Supprime L caractères dans la chaîne IN. La suppression des caractères commence à la position P (incluse) et la sous-chaîne restante est renvoyée dans le paramètre OUT.</p> <ul style="list-style-type: none"> <li>Si L est égal à zéro, la chaîne d'entrée est renvoyée dans OUT.</li> <li>Si la somme de L et de P est supérieure à la longueur de la chaîne d'entrée, la chaîne est effacée jusqu'à la fin.</li> </ul>

Tableau 8- 59 Types de données pour les paramètres

Paramètre et type	Type de données	Description
IN	IN	String
L	IN	Int
P	IN	Int
OUT	OUT	String

Tableau 8- 60 Etat de ENO

ENO	Situation d'erreur	OUT
1	Pas d'erreur détectée	Caractères valides
0	P est supérieur à la longueur en cours de IN	IN est copié dans OUT sans qu'aucun caractère ne soit supprimé.
	La chaîne résultante après suppression des caractères est plus grande que la longueur maximale de la chaîne OUT.	Les caractères de la chaîne résultante sont copiés jusqu'à ce que la longueur maximale de OUT soit atteinte.
	L est inférieur à 0 ou P est inférieur ou égal à 0.	La longueur en cours est mise à 0.
	La longueur en cours de IN dépasse la longueur maximale de IN ou la longueur en cours de OUT dépasse la longueur maximale de OUT.	
	La longueur maximale de IN ou OUT n'entre pas dans la zone de mémoire allouée.	
La longueur maximale de IN ou OUT est 0 ou 255.		

### 8.2.4.5 INSERT

Tableau 8- 61 Instruction Insérer des caractères dans une chaîne

CONT/LOG	SCL	Description
	<pre>out := INSERT(in1, in2, p);</pre>	Insère la chaîne IN2 dans la chaîne IN1. L'insertion commence après le caractère à la position P.

Tableau 8- 62 Types de données pour les paramètres

Paramètre et type	Type de données	Description	
IN1	IN	String	Chaîne d'entrée 1
IN2	IN	String	Chaîne d'entrée 2
P	IN	Int	Dernière position de caractère dans la chaîne IN1 avant le point d'insertion de la chaîne IN2. Le premier caractère de la chaîne IN1 correspond au numéro de position 1.
OUT	OUT	String	Chaîne résultante



Tableau 8- 63 Etat de ENO

ENO	Situation d'erreur	OUT
1	Pas d'erreur détectée	Caractères valides
0	P est supérieur à la longueur de IN1.	IN2 est concaténé à IN1 immédiatement après le dernier caractère de IN1.
	P est inférieur à 0.	La longueur en cours est mise à 0.
	La chaîne résultante après insertion est plus grande que la longueur maximale de la chaîne OUT.	Les caractères de la chaîne résultante sont copiés jusqu'à ce que la longueur maximale de OUT soit atteinte.
	La longueur en cours de IN1 dépasse la longueur maximale de IN1, la longueur en cours de IN2 dépasse la longueur maximale de IN2 ou la longueur en cours de OUT dépasse la longueur maximale de OUT (chaîne invalide).	La longueur en cours est mise à 0.
	La longueur maximale de IN1, IN2 ou OUT n'entre pas dans la zone de mémoire allouée.	
	La longueur maximale de IN1 ou IN2 est 255 ou la longueur maximale de OUT est 0 ou 255.	

### 8.2.4.6 REPLACE

Tableau 8- 64 Instruction Remplacer des caractères dans une chaîne

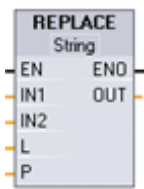
CONT/LOG	SCL	Description
	<pre> out := REPLACE(     in1:=_string_in_,     in2:=_string_in_,     L:=_int_in_,     p:=_int_in_);         </pre>	Remplace L caractères dans la chaîne IN1. La substitution commence à la position de caractère P (incluse) de la chaîne IN1, les caractères de substitution provenant de la chaîne IN2.

Tableau 8- 65 Types de données pour les paramètres

Paramètre et type	Type de données	Description	
IN1	IN	String	Chaîne d'entrée
IN2	IN	String	Chaîne de caractères de substitution
L	IN	Int	Nombre de caractères à remplacer
P	IN	Int	Position du premier caractère à remplacer
OUT	OUT	String	Chaîne résultante

Si le paramètre L est égal à zéro, la chaîne IN2 est insérée à la position P de la chaîne IN1 sans qu'aucun caractère de la chaîne IN1 ne soit effacé.

Si P est égal à 1, les L premiers caractères de la chaîne IN1 sont remplacés par les caractères de la chaîne IN2.

Tableau 8- 66 Etat de ENO

ENO	Situation d'erreur	OUT
1	Pas d'erreur détectée	Caractères valides
0	P est supérieur à la longueur de IN1.	IN2 est concaténé à IN1 immédiatement après le dernier caractère de IN1.
	P pointe à l'intérieur de IN1 mais il reste moins de L caractères dans IN1.	IN2 remplace les derniers caractères de IN1 en commençant à la position P.
	La chaîne résultante après substitution est plus grande que la longueur maximale de la chaîne OUT.	Les caractères de la chaîne résultante sont copiés jusqu'à ce que la longueur maximale de OUT soit atteinte.
	La longueur maximale de IN1 est 0.	IN2 Les caractères de IN2 sont copiés dans OUT
	L est inférieur à 0 ou P est inférieur ou égal à 0.	La longueur en cours est mise à 0.
	La longueur en cours de IN1 dépasse la longueur maximale de IN1, la longueur en cours de IN2 dépasse la longueur maximale de IN2 ou la longueur en cours de OUT dépasse la longueur maximale de OUT.	
	La longueur maximale de IN1, IN2 ou OUT n'entre pas dans la zone de mémoire allouée.	
	La longueur maximale de IN1 ou IN2 est 255 ou la longueur maximale de OUT est 0 ou 255.	

8.2.4.7 FIND

Tableau 8- 67 Instruction Trouver des caractères dans une chaîne

CONT/LOG	SCL	Description
	<pre>out := FIND(     in1:=_string_in_,     in2:=_string_in_);</pre>	Renvoie la position de la sous-chaîne indiquée par IN2 à l'intérieur de la chaîne IN1. La recherche commence par la gauche. La position de caractère de la première occurrence de la chaîne IN2 est renvoyée dans OUT. Si la chaîne IN2 est introuvable dans la chaîne IN1, zéro est renvoyé.

Tableau 8- 68 Types de données pour les paramètres

Paramètre et type	Type de données	Description	
IN1	IN	String	Recherche à l'intérieur de cette chaîne
IN2	IN	String	Chaîne recherchée
OUT	OUT	Int	Position de caractère dans la chaîne IN1 de la première occurrence de la chaîne recherchée

Tableau 8- 69 Etat de ENO

ENO	Situation d'erreur	OUT
1	Pas d'erreur détectée	Position de caractère valide
0	IN2 est plus grand que IN1	La position de caractère est mise à 0.
	La longueur en cours de IN1 dépasse la longueur maximale de IN1 ou la longueur en cours de IN2 dépasse la longueur maximale de IN2 (chaîne invalide).	
	La longueur maximale de IN1 ou IN2 n'entre pas dans la zone de mémoire allouée.	
	La longueur maximale de IN1 ou IN2 est 255.	

## 8.3 Périphérie décentralisée (PROFINET, PROFIBUS ou AS-i)

### 8.3.1 Instructions de périphérie décentralisée

Vous pouvez utiliser les instructions de périphérie décentralisée suivantes avec PROFINET, PROFIBUS ou AS-i :

- Instruction RDREC (Page 292): Vous pouvez lire un enregistrement de numéro INDEX dans un module ou un périphérique.
- Instruction WRREC (Page 292): Vous pouvez transférer un enregistrement de numéro INDEX dans un module ou un périphérique défini par ID.
- Instruction RALRM (Page 295): Vous pouvez recevoir d'un module ou d'un périphérique une alarme avec toutes les informations correspondantes et fournir ces informations au niveau des paramètres de sortie.
- Instruction DPRD\_DAT (Page 302): Vous devez lire les zones de données cohérentes de plus de 64 octets dans un module ou un périphérique à l'aide de l'instruction DPRD\_DAT.
- Instruction DPWR\_DAT (Page 302): Vous devez écrire les zones de données cohérentes de plus de 64 octets dans un module ou un périphérique à l'aide de l'instruction DPWR\_DAT.

L'instruction DPNRM\_DG (Page 304) ne peut être utilisée qu'avec PROFIBUS. Vous pouvez lire les données de diagnostic en cours d'un esclave DP dans le format spécifié par EN 50 170 Volume 2, PROFIBUS.

### 8.3.2 RDREC et WRREC

Vous pouvez utiliser les instructions RDREC (Lecture enregistrement) et WRREC (Ecriture enregistrement) avec PROFINET, PROFIBUS et AS-i.

Tableau 8- 70 Instructions RDREC et WRREC

CONT/LOG	SCL	Description
<p>"RDREC_DB"</p>	<pre>"RDREC_DB" (   req:=_bool_in_,   ID:=_word_in_,   index:=_dint_in_,   mlen:=_uint_in_,   valid=&gt;_bool_out_,   busy=&gt;_bool_out_,   error=&gt;_bool_out_,   status=&gt;_dword_out_,   len=&gt;_uint_out_,   record:=_variant_inout_);</pre>	<p>Utilisez l'instruction RDREC pour lire l'enregistrement de numéro INDEX dans le composant désigné par le paramètre ID, par exemple un châssis central ou un composant décentralisé (PROFIBUS DP ou PROFINET IO). Indiquez dans MLEN le nombre maximum d'octets à lire. La longueur de la zone cible RECORD doit donc être d'au moins MLEN octets.</p>
<p>"WRREC_DB"</p>	<pre>"WRREC_DB" (   req:=_bool_in_,   ID:=_word_in_,   index:=_dint_in_,   len:=_uint_in_,   done=&gt;_bool_out_,   busy=&gt;_bool_out_,   error=&gt;_bool_out_,   status=&gt;_dword_out_,   record:=_variant_inout_);</pre>	<p>Utilisez l'instruction WRREC pour transférer l'enregistrement RECORD de numéro INDEX dans un esclave DP/périphérique PROFINET IO désigné par ID, par exemple un module dans le châssis central ou un composant décentralisé (PROFIBUS DP ou PROFINET IO).</p> <p>Indiquez dans LEN la longueur en octets de l'enregistrement à transmettre. La longueur de la zone source RECORD doit donc être d'au moins LEN octets.</p>

- 1 STEP 7 crée automatiquement le DB lorsque vous insérez l'instruction.
- 2 Dans l'exemple SCL, "RDREC\_DB" et "WRREC\_DB" sont les noms des DB d'instance.

Tableau 8- 71 Types de données pour les paramètres de RDREC et WRREC

Paramètre et type		Type de données	Description
REQ	IN	Bool	REQ = 1 : Transférer un enregistrement
ID	IN	HW_IO (Word)	<p>Adresse logique de l'esclave DP/composant PROFINET IO (module ou sous-module) :</p> <ul style="list-style-type: none"> <li>• Pour un module de sorties, le bit 15 doit être à 1 (par exemple, pour l'adresse 5 : ID:= DW#16#8005).</li> <li>• Pour un module d'entrées-sorties, la plus petite des deux adresses doit être indiquée.</li> </ul> <p><b>Remarque</b> : L'ID d'appareil peut être déterminée de l'une des deux façons suivantes :</p> <ul style="list-style-type: none"> <li>• Via les sélections suivantes dans la vue du réseau : <ul style="list-style-type: none"> <li>– Appareil (boîte grise)</li> <li>– "Propriétés" de l'appareil</li> <li>– "Identificateur matériel"</li> </ul> <p><b>Remarque</b> : Tous les appareils n'affichent toutefois pas leur identificateur matériel.</p> </li> <li>• Via les sélections suivantes dans la vue du projet : <ul style="list-style-type: none"> <li>– Variables API</li> <li>– Table de variables par défaut</li> <li>– Onglet Constantes système</li> </ul> </li> </ul> <p>Tous les identificateurs matériels d'appareils configurés sont affichés.</p>
INDEX	IN	Byte, Word, USInt, UInt, SInt, Int, DInt	Numéro de l'enregistrement
MLEN	IN	Byte, USInt, UInt	Longueur maximale en octets des informations de l'enregistrement à lire (RDREC)
VALID	OUT	Bool	Un nouvel enregistrement valide a été reçu (RDREC). Le bit VALID est VRAI pour un cycle lorsque la dernière demande s'est achevée sans erreur.
DONE	OUT	Bool	L'enregistrement a été transféré (WRREC). Le bit DONE est VRAI pour un cycle lorsque la dernière demande s'est achevée sans erreur.
BUSY	OUT	Bool	<ul style="list-style-type: none"> <li>• BUSY = 1 : L'opération de lecture (RDREC) ou d'écriture (WRREC) n'est pas encore terminée.</li> <li>• BUSY = 0 : La transmission de l'enregistrement est achevée.</li> </ul>
ERROR	OUT	Bool	ERROR = 1 : Une erreur de lecture (RDREC) ou d'écriture (WRREC) s'est produite. Le bit ERROR est VRAI pour un cycle lorsque la dernière demande s'est achevée avec une erreur. La valeur de code d'erreur dans le paramètre STATUS ne vaut que pendant le cycle où ERROR est à VRAI.
STATUS	OUT	DWord	Etat du bloc ou information d'erreur

Paramètre et type		Type de données	Description
LEN	OUT (RDREC) IN (WRREC)	UInt	<ul style="list-style-type: none"> <li>Longueur des informations lues de l'enregistrement (RDREC)</li> <li>Longueur maximale en octets de l'enregistrement à écrire (WRREC)</li> </ul>
RECORD	IN_OUT	Variant	<ul style="list-style-type: none"> <li>Zone cible pour l'enregistrement lu (RDREC)</li> <li>Enregistrement (WRREC)</li> </ul>

Les instructions RDREC et WRREC ont un fonctionnement asynchrone, c'est-à-dire que le traitement s'étend sur plusieurs appels d'instruction. Lancez la tâche en appelant RDREC ou WRREC avec REQ égal à 1.

L'état de la tâche est indiqué dans le paramètre de sortie BUSY et dans les deux octets centraux du paramètre de sortie STATUS. Le transfert de l'enregistrement est achevé lorsque le paramètre de sortie BUSY a la valeur FALSE.

La valeur TRUE (pendant un cycle seulement) du paramètre de sortie VALID (RDREC) ou DONE (WRREC) signifie que l'enregistrement a été transféré avec succès dans la zone cible RECORD (RDREC) ou dans l'appareil cible (WRREC). Pour RDREC, le paramètre de sortie LEN contient la longueur des données lues en octets.

Le paramètre de sortie ERROR indique (pendant un cycle seulement lorsque ERROR = TRUE) qu'une erreur de transmission d'enregistrement s'est produite. Si c'est le cas, le paramètre de sortie STATUS contient les informations d'erreur correspondantes (pendant un cycle seulement lorsque ERROR = TRUE).

Les enregistrements sont définis par le fabricant de l'appareil. Reportez-vous à la documentation du fabricant concernant l'appareil pour obtenir plus de détails sur un enregistrement.

---

#### Remarque

Si un esclave DPV1 est configuré au moyen d'un fichier GSD (GSD version 3 ou plus) et que l'interface DP du maître DP est paramétrée à "S7 compatible", vous ne pourrez pas lire d'enregistrements des modules d'entrées-sorties dans le programme utilisateur avec "RDREC" ou écrire dans les modules d'entrées-sorties avec "WRREC". Dans ce cas, le maître DP accède au mauvais emplacement (emplacement configuré + 3).

Solution : Définissez l'interface du maître DP à "DPV1".

---

#### Remarque

Les interfaces des instructions "RDREC" et "WRREC" sont identiques aux FB "RDREC" et "WRREC" définis dans la publication "PROFIBUS Guideline PROFIBUS Communication and Proxy Function Blocks according to IEC 61131-3".

---

#### Remarque

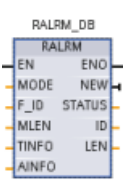
Si vous utilisez "RDREC" ou "WRREC" pour lire ou écrire un enregistrement pour PROFINET IO, des valeurs négatives dans les paramètres INDEX, MLEN et LEN sont interprétées en tant qu'entiers de 16 bits non signés.

---

### 8.3.3 RALRM

Vous pouvez utiliser l'instruction RALRM (Lecture d'alarme) avec PROFINET et PROFIBUS.

Tableau 8- 72 Instruction RALRM

CONT/LOG	SCL	Description
	<pre>"RALRM_DB" (   mode:=_int_in_,   f_ID:=_word_in_,   mlen:=_uint_in_,   new=&gt;_bool_out_,   status=&gt;_dword_out_,   ID=&gt;_word_out_,   len=&gt;_uint_out_,   tinfo:=_variant_inout_,   ainfo:=_variant_inout_);</pre>	<p>Utilisez l'instruction RALRM (Lecture d'alarme) pour lire des informations d'alarme de diagnostic en provenance de modules ou de périphérique PROFIBUS ou PROFINET IO.</p> <p>Les informations dans les paramètres de sortie contiennent les informations de déclenchement de l'OB appelé, ainsi que des informations de la source de l'alarme.</p> <p>Appelez RALRM dans un OB d'alarme pour renvoyer des informations concernant le ou les événements ayant provoqué l'alarme. Seules les alarmes de diagnostic (OB82) sont prises en charge dans le S7-1200.</p>

- STEP 7 crée automatiquement le DB lorsque vous insérez l'instruction.
- Dans l'exemple SCL, "RALRM\_DB" est le nom du DB d'instance.

Tableau 8- 73 Types de données pour les paramètres

Paramètre et type	Type de données	Description
MODE	IN	Byte, USInt, SInt, Int
MODE	IN	Mode de fonctionnement
F_ID	IN	HW_IO (Word)
F_ID	IN	Adresse de début logique du composant (module) dont il faut recevoir les alarmes <b>Remarque :</b> L'ID d'appareil peut être déterminée de l'une des deux façons suivantes : <ul style="list-style-type: none"> <li>Via les sélections suivantes dans la vue du réseau : <ul style="list-style-type: none"> <li>Appareil (boîte grise)</li> <li>"Propriétés" de l'appareil</li> <li>"Identificateur matériel"</li> </ul> <b>Remarque :</b> Tous les appareils n'affichent pas leur identificateur matériel. </li> <li>Via les sélections suivantes dans la vue du projet : <ul style="list-style-type: none"> <li>Variables API</li> <li>Table de variables par défaut</li> <li>Onglet Constantes système</li> <li>Tous les identificateurs matériels d'appareils configurés sont affichés.</li> </ul> </li> </ul>
MLEN	IN	Byte, USInt, UInt
MLEN	IN	Longueur maximale en octets des informations d'alarme à recevoir. MLEN égal à 0 permet de recevoir autant d'informations d'alarme qu'il y en a dans la zone cible AINFO.
NEW	OUT	Bool
NEW	OUT	Une nouvelle alarme a été reçue.
STATUS	OUT	DWord
STATUS	OUT	Etat de l'instruction RALRM. Reportez-vous à "Paramètre STATUS pour RDREC, WRREC et RALRM" (Page 298) pour plus d'informations.

Paramètre et type		Type de données	Description
ID	OUT	HW_IO (Word)	Identificateur matériel du module d'E/S ayant provoqué l'alarme de diagnostic <b>Remarque</b> : La description du paramètre F_ID explique comment déterminer l'ID d'appareil.
LEN	OUT	DWord, UInt, UInt, DInt, Real, LReal	Longueur des informations d'alarme AINFO reçues
TINFO	IN_OUT	Variant	Informations sur la tâche : Plage cible pour les informations de déclenchement et de gestion d'OB. La longueur TINFO est toujours de 32 octets.
AINFO	IN_OUT	Variant	Informations sur l'alarme : Zone cible pour les informations d'en-tête et les informations d'alarme supplémentaires. Indiquez une longueur d'au moins MLEN octets pour AINFO si MLEN est supérieur à 0. La longueur AINFO est variable.

---

**Remarque**

Si vous appelez "RALRM" dans un OB dont l'événement déclencheur n'est pas une alarme d'E/S, l'instruction fournira des informations réduites dans ses sorties.

Veillez à utiliser des DB d'instance différents lorsque vous appelez "RALRM" dans différents OB. Si vous évaluez des données résultant d'un appel "RALRM" en dehors de l'OB d'alarme associé, utilisez un DB d'instance distinct pour chaque événement déclencheur d'OB.

---

**Remarque**

L'interface de l'instruction "RALRM" est identique au FB "RALRM" défini dans la publication "PROFIBUS Guideline PROFIBUS Communication and Proxy Function Blocks according to IEC 61131-3".

---



## Appel de RALRM

Vous pouvez appeler l'instruction RALRM dans trois modes différents (MODE).

Tableau 8- 74 Modes de fonctionnement de l'instruction RALRM

MODE	Description
0	<ul style="list-style-type: none"> <li>ID contient l'identificateur matériel du module d'E/S ayant déclenché l'alarme.</li> <li>Le paramètre de sortie NEW est mis à TRUE.</li> <li>LEN produit une sortie de 0.</li> <li>AINFO et TINFO ne sont pas actualisés.</li> </ul>
1	<ul style="list-style-type: none"> <li>ID contient l'identificateur matériel du module d'E/S ayant déclenché l'alarme.</li> <li>Le paramètre de sortie NEW est mis à TRUE.</li> <li>LEN produit une sortie de la quantité de données AINFO renvoyées en octets.</li> <li>AINFO et TINFO sont actualisés par des informations relatives à l'alarme.</li> </ul>
2	<p>Si l'identificateur matériel affecté au paramètre d'entrée F_ID a déclenché l'alarme, alors :</p> <ul style="list-style-type: none"> <li>ID contient l'identificateur matériel du module d'E/S ayant déclenché l'alarme. Il devrait être identique à la valeur indiquée dans F_ID.</li> <li>Le paramètre de sortie NEW est mis à TRUE.</li> <li>LEN produit une sortie de la quantité de données AINFO renvoyées en octets.</li> <li>AINFO et TINFO sont actualisés par des informations relatives à l'alarme.</li> </ul>

### Remarque

Si vous définissez une zone de destination trop courte pour TINFO ou AINFO, RALRM ne pourra pas renvoyer les informations complètes.

MLEN peut limiter la quantité de données AINFO renvoyées.

Reportez-vous aux paramètres AINFO et aux paramètres TINFO dans le système d'information en ligne de STEP 7 pour obtenir des informations sur la manière d'interpréter les données TINFO et AINFO.

### 8.3.4 Paramètre STATUS pour RDREC, WRREC et RALRM

Le paramètre de sortie STATUS contient des informations d'erreur sous forme d'un tableau ARRAY[1...4] OF BYTE ayant la structure suivante :

Tableau 8- 75 Tableau de sortie STATUS

Elément de tableau	Nom	Description
STATUS[1]	Function_Num	<ul style="list-style-type: none"> <li>B#16#00 en l'absence d'erreur</li> <li>ID de fonction de PDU DPV1 : Si une erreur se produit, B#16#80 une opération logique OU est exécutée avec B#16#80 (pour la lecture d'enregistrement : B#16#DE ; pour l'écriture d'enregistrement : B#16#DF). Si aucun élément de protocole DPV1 n'est utilisé, B#16#C0 est transmis.</li> </ul>
STATUS[2]	Error Decode	Emplacement du code d'erreur
STATUS[3]	Error_Code_1	Code d'erreur
STATUS[4]	Error_Code_2	Extension de code d'erreur spécifique du fabricant

Tableau 8- 76 Valeurs de STATUS[2]

Error_decode (B#16#...)	Source	Description
00 à 7F	CPU	Pas d'erreur ni d'avertissement
80	DPV1	Erreur selon CEI 61158-6
81 à 8F	CPU	B#16#8x montre une erreur dans le x-ième paramètre d'appel de l'instruction.
FE, FF	Profil DP	Erreur spécifique du profil

Tableau 8- 77 Valeurs de STATUS[3]

Error_decode (B#16#...)	Error_code_1 (B#16#...)	Explication (DVP1)	Description
00	00		Pas d'erreur ni d'avertissement
70	00	Réservé, rejet	Appel initial ; pas de transfert d'enregistrement actif
	01	Réservé, rejet	Appel initial ; un transfert d'enregistrement a commencé
	02	Réservé, rejet	Appel intermédiaire ; un transfert d'enregistrement est encore actif
80	90	Réservé, acceptation	Adresse de début logique invalide
	92	Réservé, acceptation	Type incorrect de pointeur Variant
	93	Réservé, acceptation	Le composant DP adressé via ID ou F_ID n'est pas configuré.

## 8.3 Périphérie décentralisée (PROFINET, PROFIBUS ou AS-i)

Error_decode (B#16#....)	Error_code_1 (B#16#....)	Explication (DVP1)	Description
	96		"RALRM (Page 295)" ne peut pas fournir les informations de déclenchement d'OB, les informations de gestion, les informations d'en-tête ou les informations d'alarme additionnelles. Pour les OB 4x, 55, 56, 57, 82 et 83, vous pouvez utiliser l'instruction "DPNRM_DG (Page 304)" pour lire de manière asynchrone la trame de message de diagnostic en cours de l'esclave DP concerné (informations d'adresse dans les informations de déclenchement d'OB).
	A0	Erreur de lecture	Acquittement négatif pendant la lecture dans le module
	A1	Erreur d'écriture	Acquittement négatif pendant l'écriture dans le module
	A2	Défaillance du module	Erreur de protocole DP, couche 2 (par exemple, défaillance d'esclave ou problèmes de bus)
	A3	Réservé, acceptation	<ul style="list-style-type: none"> <li>PROFIBUS DP : Erreur de protocole DP avec DDLM (Direct Data Link Mapper) ou interface utilisateur/utilisateur</li> <li>PROFINET IO : Erreur CM générale</li> </ul>
	A4	Réservé, acceptation	Communication interrompue sur le bus de communication
	A5	Réservé, acceptation	-
	A7	Réservé, acceptation	Module ou esclave DP occupé (erreur temporaire)
	A8	Conflit de version	Le module ou l'esclave DP signale des versions non compatibles.
	A9	Fonction non prise en charge	Fonction non prise en charge par le module ou l'esclave DP
	AA à AF	Spécifique utilisateur	Le module ou l'esclave DP signale une erreur spécifique du fabricant dans son application. Consultez la documentation du fabricant du module ou de l'esclave DP.
	B0	Indice invalide	L'enregistrement est inconnu dans le module ; numéro d'enregistrement incorrect $\geq 256$
	B1	Erreur de longueur d'écriture	<p>L'information de longueur dans le paramètre RECORD est incorrecte.</p> <ul style="list-style-type: none"> <li>Pour "RALRM" : Erreur de longueur dans AINFO</li> </ul> <p><b>Remarque :</b> Reportez-vous au système d'information en ligne de STEP 7 pour obtenir des informations sur l'interprétation des mémoires tampons "AINFO" renvoyées.</p> <ul style="list-style-type: none"> <li>Pour "RDREC (Page 292)" et "WRREC (Page 292)" : Erreur de longueur dans MLEN</li> </ul>
	B2	Emplacement incorrect	L'emplacement configuré n'est pas occupé.
	B3	Conflit de type	Le type du module réel ne correspond pas au type de module spécifié.
	B4	Zone invalide	Le module ou l'esclave DP signale un accès à une zone invalide.

Error_decode (B#16#...)	Error_code_1 (B#16#...)	Explication (DVP1)	Description
	B5	Conflit d'état	Le module ou l'esclave DP n'est pas prêt.
	B6	Accès refusé	Le module ou l'esclave DP refuse l'accès.
	B7	Plage invalide	Le module ou l'esclave DP signale une plage invalide pour un paramètre ou une valeur.
	B8	Paramètre invalide	Le module ou l'esclave DP signale un paramètre invalide.
	B9	Type invalide	Le module ou l'esclave DP signale un tyoe invalide : <ul style="list-style-type: none"> <li>Pour "RDREC (Page 292)" : Mémoire tampon trop petite (impossible de lire les sous-ensembles)</li> <li>Pour "WRREC (Page 292)" : Mémoire tampon trop petite (impossible d'écrire les sous-ensembles)</li> </ul>
	BA à BF	Spécifique utilisateur	Le module ou l'esclave DP signale une erreur spécifique du fabricant lors de l'accès. Consultez la documentation du fabricant du module ou de l'esclave DP.
	C0	Conflit de limite de lecture	<ul style="list-style-type: none"> <li>Pour "WRREC (Page 292)" : L'écriture des données n'est possible que lorsque la CPU est à l'état ARRET. <b>Remarque</b> : Cela signifie que les données ne peuvent pas être écrites par le programme utilisateur. Vous pouvez uniquement écrire les données en ligne avec une PG/un PC.</li> <li>Pour "RDREC (Page 292)" : Le module transmet l'enregistrement mais soit il n'y a pas de données, soit les données peuvent uniquement être lues lorsque la CPU est à l'état ARRET. <b>Remarque</b> : Si les données peuvent uniquement être lues lorsque la CPU est à l'état ARRET, aucune évaluation par le programme utilisateur n'est possible. Dans ce cas, vous pouvez uniquement lire les données en ligne avec une PG/un PC.</li> </ul>
	C1	Conflit de limite d'écriture	Les données de la précédente demande d'écriture dans le module pour le même enregistrement n'ont pas encore été traitées par le module.
	C2	Ressource occupée	Le module traite actuellement le nombre maximum de tâches possibles pour une CPU.
	C3	Ressource non disponible	Les ressources d'exploitation requises sont actuellement occupées.
	C4		Erreur temporaire interne. La tâche n'a pas pu être exécutée. Relancez la tâche. Si cette erreur se produit souvent, vérifiez que votre installation ne présente pas de sources d'interférence électrique.
	C5		Le module ou l'esclave DP n'est pas disponible.
	C6		Le transfert d'enregistrement a été annulé du fait d'une annulation de classe de priorité.
	C7		Tâche interrompue en raison d'un démarrage à chaud ou à froid du maître DP.

## 8.3 Périphérie décentralisée (PROFINET, PROFIBUS ou AS-i)

Error_decode (B#16#....)	Error_code_1 (B#16#....)	Explication (DVP1)	Description
	C8 à CF		Le module ou l'esclave DP signale une erreur de ressource spécifique du fabricant. Consultez la documentation du fabricant du module ou de l'esclave DP.
	Dx	Spécifique utilisateur	Spécifique de l'esclave DP. Consultez la description de l'esclave DP.
81	00 à FF		Erreur dans le premier paramètre d'appel (pour "RALRM (Page 295)" : MODE)
	00		Mode de fonctionnement interdit
82	00 à FF		Erreur dans le deuxième paramètre d'appel
88	00 à FF		Erreur dans le huitième paramètre d'appel (pour "RALRM (Page 295)" : TINFO) <b>Remarque</b> : Reportez-vous au système d'information en ligne de STEP 7 pour obtenir des informations sur l'interprétation des mémoires tampons "TINFO" renvoyées.
	01		ID de syntaxe illicite
	23		Structure de quantité dépassée ou zone de destination trop petite
	24		ID de plage illicite
	32		Numéro de DB/DI en dehors de la plage utilisateur
	3A		Numéro de DB/DI nul pour l'ID de zone DB/DI ou DB/DI indiqué inexistant
89	00 à FF		Erreur dans le neuvième paramètre d'appel (pour "RALRM (Page 295)" : AINFO) <b>Remarque</b> : Reportez-vous au système d'information en ligne de STEP 7 pour obtenir des informations sur l'interprétation des mémoires tampons "AINFO" renvoyées.
	01		ID de syntaxe illicite
	23		Structure de quantité dépassée ou zone de destination trop petite
	24		ID de plage illicite
	32		Numéro de DB/DI en dehors de la plage utilisateur
	3A		Numéro de DB/DI nul pour l'ID de zone DB/DI ou DB/DI indiqué inexistant
8A	00 à FF		Erreur dans le dixième paramètre d'appel
8F	00 à FF		Erreur dans le quinzième paramètre d'appel
FE, FF	00 à FF		Erreur spécifique du profil

**Élément de tableau STATUS[4]**

En présence d'erreurs DPV1, le maître DP transmet STATUS[4] à la CPU et à l'instruction. En l'absence d'erreur DPV1, cette valeur est mise à 0, aux exceptions suivantes près pour RDREC :

- STATUS[4] contient la longueur de la zone cible de RECORD si MLEN > longueur de la zone cible de RECORD.
- STATUS[4] est égal à MLEN si longueur réelle de l'enregistrement < MLEN < longueur de la zone cible de RECORD.
- STATUS[4] est égal à 0 si STATUS[4] > 255 ; devrait être défini.

Dans PROFINET IO, STATUS[4] a la valeur 0.

**8.3.5 DPRD\_DAT et DPWR\_DAT**

Vous pouvez utiliser les instructions DPRD\_DAT (Lecture de données cohérentes) et DPWR\_DAT (Ecriture de données cohérentes) avec PROFINET et PROFIBUS.

Tableau 8- 78 Instructions DPRD\_DAT et DPWR\_DAT

CONT/LOG	SCL	Description
	<pre>ret_val := DPRD_DAT(     laddr:=_word_in_,     record=&gt;_variant_out_);</pre>	<p>Utilisez l'instruction DPRD_DAT pour lire les données cohérentes d'un esclave norme DP/périphérique PROFINET IO. Si aucune erreur ne survient pendant le transfert de données, les données lues sont entrées dans la zone cible définie par le paramètre RECORD. La zone cible doit avoir la même longueur que celle que vous avez configurée avec STEP 7 pour le module sélectionné. Lorsque vous appelez l'instruction DPRD_DAT, vous pouvez accéder uniquement aux données d'un module / identificateur DP à l'adresse de début configurée.</p>
	<pre>ret_val := DPWR_DAT(     laddr:=_word_in_,     record:=_variant_in_);</pre>	<p>Utilisez l'instruction DPWR_DAT pour écrire les données dans RECORD de manière cohérente dans l'esclave norme DP/périphérique PROFINET IO adressé. La zone source doit avoir la même longueur que celle que vous avez configurée avec STEP 7 pour le module sélectionné.</p>

La CPU prend en charge jusqu'à 64 octets de données cohérentes. Vous devez utiliser les instructions DPRD\_DAT et DPWR\_DAT pour des zones de données cohérentes supérieures à 64 octets. Si nécessaire, ces instructions peuvent être utilisées pour des zones de données à partir de 1 octet. Le code d'erreur W#16#8090 est renvoyé si l'accès est refusé.

**Remarque**

Si vous utilisez les instructions DPRD\_DAT et DPWR\_DAT avec des données cohérentes, vous devez supprimer ces dernières de l'actualisation automatique de la mémoire image. Reportez-vous à "Concepts API : Exécution du programme utilisateur" (Page 71) pour plus d'informations.

Tableau 8- 79 Types de données pour les paramètres

Paramètre et type		Type de données	Description
LADDR	IN	HW_IO (Word)	<ul style="list-style-type: none"> <li>Adresse de début configurée de la zone I du module dans lequel les données seront lues (DPRD_DAT)</li> <li>Adresse de début configurée de la mémoire image des sorties du module dans lequel les données seront écrites (DPWR_DAT)</li> </ul> Il faut entrer les adresses en format hexadécimal (ainsi, l'adresse d'entrée ou de sortie 100 correspond à LADDR:=W#16#64).
RECORD	OUT	Variant	Zone de destination pour les données utilisateur lues (DPRD_DAT) ou zone source pour les données utilisateur à écrire (DPWR_DAT). Cette zone doit être exactement de la taille que vous avez configurée pour le module sélectionné avec STEP 7. Seul le type de données Byte est autorisé.
RET_VAL	OUT	Int	Si une erreur se produit pendant que la fonction est active, la valeur en retour contient un code d'erreur.

### Fonctionnement de DPRD\_DAT

La zone de destination doit avoir la même longueur que celle configurée pour le module sélectionné avec STEP 7. Si aucune erreur ne se produit pendant le transfert de données, les données qui ont été lues sont entrées dans la zone de destination identifiée par RECORD.

Si vous effectuez la lecture dans un esclave norme DP de conception modulaire ou ayant plusieurs identificateurs DP, vous pouvez uniquement accéder aux données d'un module / identificateur DP à chaque appel de l'instruction DPRD\_DAT, en indiquant l'adresse de début configurée.

### Fonctionnement de DPWR\_DAT

Vous transférez les données se trouvant dans RECORD de manière cohérente dans l'esclave norme DP/périphérique PROFINET IO adressé. Les données sont transférées de manière synchrone, c'est-à-dire que l'opération d'écriture est achevée à l'achèvement de l'instruction.

La zone source doit avoir la même longueur que celle que vous avez configurée pour le module sélectionné avec STEP 7.

Si l'esclave norme DP est de conception modulaire, vous ne pouvez accéder qu'à un module de l'esclave DP.

Tableau 8- 80 Codes d'erreur de DPRD\_DAT et DPWR\_DAT

Code d'erreur	Description
0000	Pas d'erreur
808x	Erreur système avec le coupleur DP externe

8.3 Périphérie décentralisée (PROFINET, PROFIBUS ou AS-i)

Code d'erreur	Description
8090	L'un des cas suivants s'est produit : <ul style="list-style-type: none"> <li>• Vous n'avez pas configuré de module pour l'adresse de base logique indiquée.</li> <li>• Vous n'avez pas tenu compte de la restriction concernant la longueur des données cohérentes.</li> <li>• Vous n'avez pas entré l'adresse de début dans le paramètre LADDR en format hexadécimal.</li> </ul>
8092	Un type autre que Byte est indiqué dans la référence Any.
8093	Il n'existe pas de module DP/périphérique PROFINET IO dans lequel vous pouvez lire (DPRD_DAT) ou écrire (DPWR_DAT) des données cohérentes à l'adresse logique indiquée dans LADDR.
80A0	Erreur d'accès détectée pendant l'accès à des périphériques IO (DPRD_DAT).
80A1	Erreur d'accès détectée pendant l'accès à des périphériques IO (DPWR_DAT).
80B0	Défaillance d'esclave sur le coupleur DP externe
80B1	La longueur de la zone de destination (DPRD_DAT) ou source (DPWR_DAT) indiquée n'est pas identique à la longueur des données utilisateur configurée avec STEP 7 Basic.
80B2, 80B3, 80C2, 80Fx	Erreur système avec le coupleur DP externe (DPRD_DAT et DPWR_DAT)
87xy, 808x	Erreur système avec le coupleur DP externe (DPRD_DAT)
85xy	Erreur système avec le coupleur DP externe (DPWR_DAT)
80C0	Les données n'ont pas encore été lues par le module (DPRD_DAT).
80C1	Les données de la tâche d'écriture précédente sur le module n'ont pas encore été traitées par le module (DPWR_DAT).
8xyy <sup>1</sup>	Informations d'erreur générales

Reportez-vous à "Instructions avancées, périphérie décentralisée : Informations d'erreur pour RDREC, WRREC et RALRM" (Page 298) pour plus d'informations sur les codes d'erreur généraux.

**Remarque**

Si vous accédez à des esclaves DPV1, les informations d'erreur de ces esclaves peuvent être transmises du maître DP à l'instruction.

**8.3.6 DPNRM\_DG**

Vous pouvez utiliser l'instruction DPNRM\_DG (Lecture de données de diagnostic) avec PROFIBUS.

Tableau 8- 81 Instruction DPNRM\_DG

CONT/LOG	SCL	Description
	<pre>ret_val := DPNRM_DG(     req:= bool_in_,     laddr:= word_in_,     record=&gt;_variant_out_,     busy=&gt; bool_out );</pre>	<p>Utilisez l'instruction DPNRM_DG pour lire les données de diagnostic en cours d'un esclave DP dans le format spécifié par EN 50 170 Volume 2, PROFIBUS. Les données qui ont été lues sont entrées dans la zone de destination indiquée par RECORD lorsque le transfert de données s'est effectué sans erreur.</p>



Tableau 8- 82 Types de données pour les paramètres de l'instruction DPNRM\_DG

Paramètre et type		Type de données	Description
REQ	IN	Bool	REQ=1 : Demande de lecture
LADDR	IN	HW_DPSLAVE	Adresse de diagnostic configurée de l'esclave DP : doit être l'adresse de la station et non celle du périphérique I/O Sélectionnez la station (et non l'image du périphérique) dans la vue de réseau de la configuration de l'appareil afin de déterminer l'adresse de diagnostic.  Entrez les adresses au format hexadécimal. L'adresse de diagnostic 1022 signifie par exemple LADDR:=W#16#3FE.
RET_VAL	OUT	Int	Si une erreur se produit pendant que la fonction est active, la valeur en retour contient un code d'erreur. En l'absence d'erreur, la longueur des données effectivement transférées est fournie dans RET_VAL.
RECORD	OUT	Variante	Zone de destination pour les données de diagnostic qui ont été lues. Seul le type de données Byte est autorisé. La longueur minimum de l'enregistrement à lire ou de la zone de destination est 6. La longueur maximum de l'enregistrement à envoyer est 240.  Les esclaves norme peuvent fournir plus de 240 octets de données de diagnostic, avec un maximum de 244 octets. Dans ce cas, les 240 premiers octets sont transférés dans la zone de destination et le bit de débordement est mis à 1 dans les données.
BUSY	OUT	Bool	BUSY=1 : La tâche de lecture n'est pas encore achevée.

Vous démarrez la tâche de lecture en mettant le paramètre d'entrée REQ à 1 dans l'appel de l'instruction DPNRM\_DG. La tâche de lecture est exécutée de manière asynchrone ; en d'autres termes, elle nécessite plusieurs appels de l'instruction DPNRM\_DG. L'état de la tâche est indiqué par les paramètres de sortie RET\_VAL et BUSY.

Tableau 8- 83 Structure des données de diagnostic esclave

Octet	Description
0	Etat de station 1
1	Etat de station 2
2	Etat de station 3
3	Numéro de la station maître
4	ID de fournisseur (octet de poids fort)
5	ID de fournisseur (octet de poids faible)
6 ...	Informations de diagnostic supplémentaires spécifiques de l'esclave

Tableau 8- 84 Codes d'erreur de l'instruction DPNRM\_DG

Code d'erreur	Description	Restriction
0000	Pas d'erreur	-
7000	Premier appel avec REQ=0 : Pas de transfert de données actif ; BUSY a la valeur 0.	-
7001	Premier appel avec REQ=1 : Pas de transfert de données actif ; BUSY a la valeur 1.	E/S décentralisées

8.3 Périphérie décentralisée (PROFINET, PROFIBUS ou AS-i)

Code d'erreur	Description	Restriction
7002	Appel intermédiaire (REQ non significatif) : Un transfert de données est déjà actif ; BUSY a la valeur 1.	E/S décentralisées
8090	Adresse de base logique indiquée invalide : Il n'y a pas d'adresse de base.	-
8092	Le type indiqué dans la référence Any n'est pas Byte.	-
8093	<ul style="list-style-type: none"> <li>Cette instruction n'est pas autorisée pour le module indiqué par LADDR (les modules S7 DP pour le S7-1200 sont autorisés).</li> <li>LADDR spécifie le périphérique I/O au lieu de spécifier la station. Sélectionnez la station (et non l'image du périphérique) dans la vue de réseau de la configuration de l'appareil afin de déterminer l'adresse de diagnostic pour LADDR.</li> </ul>	-
80A2	<ul style="list-style-type: none"> <li>Erreur de protocole DP, couche 2 (par exemple, défaillance d'esclave ou problèmes de bus)</li> <li>Pour l'ET200S, l'enregistrement ne peut pas être lu en mode DPV0.</li> </ul>	E/S décentralisées
80A3	Erreur de protocole DP avec l'interface utilisateur/utilisateur	E/S décentralisées
80A4	Problème de communication sur le bus de communication	L'erreur se produit entre la CPU et le coupleur DP externe.
80B0	<ul style="list-style-type: none"> <li>L'instruction n'est pas possible pour le type de module.</li> <li>Le module ne reconnaît pas l'enregistrement.</li> <li>Le numéro d'enregistrement 241 est interdit.</li> </ul>	-
80B1	La longueur indiquée dans le paramètre RECORD est incorrecte.	longueur indiquée > longueur de l'enregistrement
80B2	L'emplacement configuré n'est pas occupé.	-
80B3	Le type du module réel ne correspond pas au type de module demandé.	-
80C0	Il n'y a pas d'informations de diagnostic.	-
80C1	Les données de la précédente tâche d'écriture dans le module pour le même enregistrement n'ont pas encore été traitées par le module.	-
80C2	Le module traite actuellement le nombre maximum de tâches possibles pour une CPU.	-
80C3	Les ressources requises (mémoire, etc.) sont actuellement occupées.	-
80C4	Erreur temporaire interne. La tâche n'a pas pu être traitée. Relancez la tâche. Si cette erreur se produit fréquemment, vérifiez que votre système ne présente pas de sources d'interférence électrique.	-
80C5	E/S décentralisées non disponibles	E/S décentralisées
80C6	Le transfert d'enregistrement a été arrêté en raison d'une annulation de classe de priorité (redémarrage ou arrière-plan).	E/S décentralisées
8xyy <sup>1</sup>	Codes d'erreur généraux	

Reportez-vous à "Instructions avancées, périphérie décentralisée : Informations d'erreur pour RDREC, WRREC et RALRM" (Page 298) pour plus d'informations sur les codes d'erreur généraux.

## 8.4 Alarmes

### 8.4.1 Instructions ATTACH et DETACH

Vous pouvez activer et désactiver des sous-programmes d'alarme déclenchés sur événement par le biais des instructions ATTACH et DETACH.

Tableau 8- 85 Instructions ATTACH et DETACH

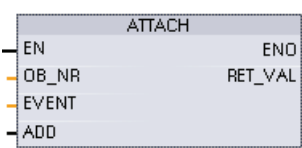
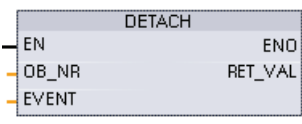
CONT/LOG	SCL	Description
	<pre>ret_val := ATTACH(   ob_nr:=_int_in_,   event:=_event_att_in_,   add:=_bool_in_);</pre>	ATTACH active l'exécution d'un sous-programme d'OB d'alarme pour un événement d'alarme de processus.
	<pre>ret_val := DETACH(   ob_nr:=_int_in_,   event:=_event_att_in_);</pre>	DETACH désactive l'exécution d'un sous-programme d'OB d'alarme pour un événement d'alarme de processus.

Tableau 8- 86 Types de données pour les paramètres

Paramètre et type		Type de données	Description
OB_NR	IN	OB_ATT	Identificateur de bloc d'organisation : Faites votre choix parmi les OB d'alarme de processus disponibles qui ont été créés via la fonction "Ajouter nouveau bloc". Double-cliquez sur le champ de paramètre, puis cliquez sur l'icône d'aide pour voir les OB disponibles.
EVENT	IN	EVENT_ATT	Identificateur d'événement : Faites votre choix parmi les événements d'alarme de processus disponibles qui ont été activés dans la configuration d'appareil de l'automate pour les entrées TOR ou les compteurs rapides. Double-cliquez sur le champ de paramètre, puis cliquez sur l'icône d'aide pour voir les événements disponibles.
ADD (ATTACH uniquement)	IN	Bool	<ul style="list-style-type: none"> <li>ADD = 0 (valeur par défaut) : Cet événement remplace toutes les associations d'événement précédentes pour cet OB.</li> <li>ADD = 1 : Cet événement est ajouté aux associations d'événement précédentes pour cet OB.</li> </ul>
RET_VAL	OUT	Int	Code d'erreur d'exécution

### Evénements alarme de processus

La CPU prend en charge les événements alarme de processus suivants :

- Evénements front montant (toutes les entrées TOR intégrées de la CPU et toutes les entrées TOR SB)
  - Un front montant se produit lorsque l'entrée TOR passe de l'état désactivé à l'état activé en réaction à un changement du signal provenant de l'appareil de terrain connecté à l'entrée.
- Evénements front descendant (toutes les entrées TOR intégrées de la CPU et toutes les entrées TOR SB)
  - Un front descendant se produit lorsque l'entrée TOR passe de l'état activé à l'état désactivé.
- Evénements valeur en cours = valeur de référence (CV = RV) de compteur rapide (compteurs HSC 1 à 6)
  - Une alarme CV = RV pour un compteur rapide est générée lorsque la valeur de comptage en cours passe d'une valeur adjacente à la valeur correspondant exactement à une valeur de référence définie précédemment.
- Evénements inversion du sens HSC (compteurs HSC 1 à 6)
  - Un événement inversion du sens se produit lorsque le système détecte que le HSC est passé de l'incréméntation à la décréméntation ou de la décréméntation à l'incréméntation.
- Evénements réinitialisation externe HSC (compteurs HSC 1 à 6)
  - Certains modes HSC autorisent la définition d'une entrée TOR en tant qu'entrée de réinitialisation externe qui est utilisée pour remettre à zéro la valeur de comptage du compteur rapide. Un événement réinitialisation externe se produit pour un tel compteur rapide lorsque cette entrée passe de l'état désactivé à l'état activé.

### Activation des événements alarme de processus dans la configuration des appareils

Les alarmes de processus doivent être activées pendant la configuration des appareils. Vous devez cocher la case d'activation de l'événement dans la configuration d'appareil pour une voie d'entrée TOR ou un compteur rapide si vous voulez associer cet événement pendant la configuration ou au moment de l'exécution.

Options à cocher dans la configuration d'appareil de l'automate :

- Entrée TOR
  - Activer la détection du front montant
  - Activer la détection du front descendant
- Compteur rapide (HSC)
  - Activer ce compteur rapide
  - Générer une alarme pour un événement de type valeur de comptage égale à valeur de référence
  - Générer une alarme pour un événement de réinitialisation externe
  - Générer une alarme pour un événement d'inversion de sens

## Ajout de nouveaux OB d'alarme de processus à votre programme

Par défaut, aucun OB n'est associé à un événement lorsque l'événement est activé pour la première fois. Cela est signalé par la mention "<non connecté>" dans la configuration des alarmes de processus. Seuls des OB d'alarme de processus peuvent être associés à un événement d'alarme de processus. Tous les OB d'alarme de processus existants apparaissent dans la liste déroulante "Alarme de processus :". Si aucun OB n'est listé, vous devez créer un OB de type "alarme de processus" de la manière suivante. Dans la branche "Blocs de programme" de l'arborescence de projet :

1. Double-cliquez sur "Ajouter nouveau bloc", sélectionnez "Bloc d'organisation (OB)", puis choisissez "Alarme de processus".
2. Vous pouvez optionnellement renommer l'OB, sélectionner le langage de programmation (CONT ou LOG) et sélectionner le numéro de bloc (commutez en manuel et choisissez un numéro de bloc différent de celui suggéré).
3. Editez l'OB pour y ajouter la réaction programmée que vous voulez exécuter lorsque l'événement se produit. Vous pouvez appeler des FC et des FB à partir de cet OB, jusqu'à une profondeur d'imbrication de quatre.

## Paramètre OB\_NR

Tous les OB d'alarme de processus existants apparaissent dans la liste déroulante "Alarme de processus :" de la configuration d'appareil et dans la liste déroulante du paramètre OB\_NR de ATTACH / DETACH.

## Paramètre EVENT

Lorsqu'un événement d'alarme de processus est activé, un nom d'événement par défaut unique est affecté à cet événement particulier. Vous pouvez modifier ce nom d'événement en éditant la boîte d'édition "Nom d'événement :", mais ce doit être un nom unique. Ces noms d'événement deviennent des noms de variable dans la table de variables "Constantes" et apparaissent dans la liste déroulante du paramètre EVENT des boîtes d'instruction ATTACH et DETACH. La valeur de la variable est un numéro interne utilisé pour identifier l'événement.

## Fonctionnement général

Chaque événement d'alarme de processus peut être associé à un OB d'alarme de processus qui sera mis en file d'attente pour exécution lorsque l'événement d'alarme de processus se produit. L'association OB-événement peut se faire au moment de la configuration ou à l'exécution.

Vous pouvez associer un OB à un événement activé ou l'en dissocier au moment de la configuration. Pour associer un OB à un événement lors de la configuration, vous devez utiliser la liste déroulante "Alarme de processus :" (cliquez sur la flèche descendante à droite) et sélectionner un OB dans la liste des OB d'alarme de processus disponibles. Sélectionnez le nom de l'OB approprié dans la liste ou sélectionnez "<non connecté>" pour annuler l'association.

8.4 Alarmes

Vous pouvez également associer ou dissocier un événement d'alarme de processus activé pendant l'exécution. Utilisez les instructions de programme ATTACH ou DETACH pendant l'exécution - à plusieurs reprises si vous le désirez - pour associer un événement d'alarme activé à l'OB approprié ou pour l'en dissocier. En l'absence d'association à un OB (par sélection de "<non connecté>" dans la configuration des appareils ou par exécution d'une instruction DETACH), l'événement d'alarme de processus activé n'est pas pris en compte.

Fonctionnement de DETACH

Utilisez l'instruction DETACH pour dissocier un événement particulier ou tous les événements d'un OB particulier. Si un événement EVENT est indiqué, seul cet événement est dissocié de l'OB de numéro OB\_NR indiqué ; tout autre événement actuellement associé à ce numéro d'OB OB\_NR lui reste associé. Si aucun événement EVENT n'est indiqué, tous les événements actuellement associés au numéro d'OB OB\_NR en sont dissociés.

Codes d'erreur

Tableau 8- 87 Codes d'erreur

RET_VAL (W#16#....)	ENO	Description
0000	1	Pas d'erreur
0001	1	Rien à dissocier (DETACH uniquement)
8090	0	OB inexistant
8091	0	Type d'OB erroné
8093	0	Événement inexistant

8.4.2 Alarmes cycliques

8.4.2.1 SET\_CINT (Régler alarme cyclique)

Tableau 8- 88 SET\_CINT (Régler alarme cyclique)

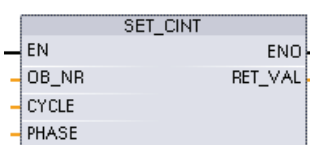
CONT/LOG	SCL	Description
	<pre>ret_val := SET_CINT(   ob_nr:=_int_in_,   cycle:=_udint_in_,   phase:=_udint_in_);</pre>	<p>Permet de définir l'exécution cyclique interrompant le cycle du programme pour l'OB d'alarme indiqué.</p>

Tableau 8- 89 Types de données pour les paramètres

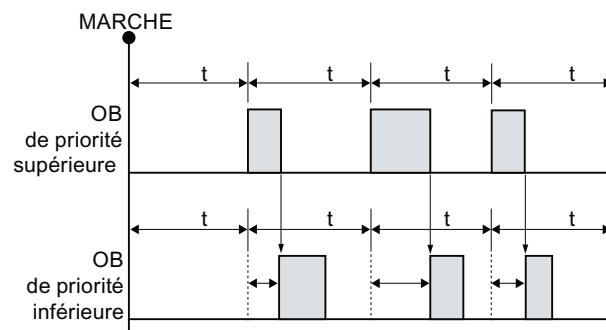
Paramètre et type		Type de données	Description
OB_NR	IN	OB_CYCLIC	Numéro d'OB (noms symboliques acceptés)
CYCLE	IN	UDInt	Intervalle de temps, en microsecondes
PHASE	IN	UDInt	Déphasage, en microsecondes
RET_VAL	OUT	Int	Code d'erreur d'exécution

Exemples de paramètres de temps :

- Si le temps CYCLE est égal à 100  $\mu$ s, l'OB d'alarme référencé par OB\_NR interrompt le cycle du programme toutes les 100  $\mu$ s. L'OB d'alarme s'exécute puis rend le contrôle au cycle du programme au niveau du point d'interruption.
- Si le temps CYCLE est égal à 0, l'événement d'alarme est désactivé et l'OB d'alarme n'est pas exécuté.
- Le déphasage PHASE est un temps de retard défini qui se produit avant que l'intervalle de temps CYCLE ne commence. Le déphasage peut vous servir à gérer l'instant d'exécution des OB de priorité inférieure.

Si des OB de priorité différente sont appelés dans le même intervalle de temps, l'OB de priorité inférieure n'est appelé qu'une fois le traitement de l'OB de priorité supérieure achevé. Le début de l'exécution de l'OB de priorité inférieure peut donc être décalé selon la durée de traitement des OB de priorité supérieure.

Appel d'OB sans déphasage



8.4 Alarmes

Si vous voulez démarrer l'exécution d'un OB de faible priorité selon un rythme fixe, définissez un temps de déphasage qui est supérieur à la durée de traitement des OB de priorité supérieure.

Appel d'OB avec déphasage

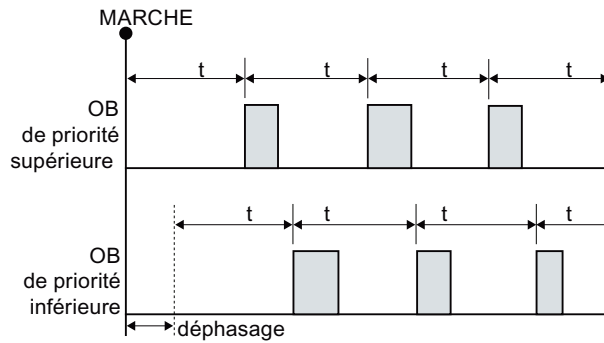


Tableau 8- 90 Codes d'erreur

RET_VAL (W#16#...)	Description
0000	Pas d'erreur
8090	L'OB n'existe pas ou a un type incorrect.
8091	Intervalle de temps invalide
8092	Déphasage invalide
80B2	Aucun événement n'est associé à l'OB.

8.4.2.2 QRY\_CINT (Interroger alarme cyclique)

Tableau 8- 91 QRY\_CINT (Interroger alarme cyclique)

CONT/LOG	SCL	Description
<pre> QRY_CINT - EN          END - OB_NR      RET_VAL               CYCLE               PHASE               STATUS                     </pre>	<pre> ret_val := QRY_CINT(   ob_nr:=_int_in_,   cycle=&gt;_udint_out_,   phase=&gt;_udint_out_,   status=&gt;_word_out_);                     </pre>	Permet d'obtenir les valeurs des paramètres et l'état d'exécution d'un OB d'alarme cyclique. Les valeurs renvoyées sont celles qui étaient en vigueur au moment de l'exécution de QRY_CINT.



Tableau 8- 92 Types de données pour les paramètres

Paramètre et type		Type de données	Description
OB_NR	IN	OB_CYCLIC	Numéro d'OB (noms symboliques comme OB_MyOBName acceptés)
RET_VAL	OUT	Int	Code d'erreur d'exécution
CYCLE	OUT	UDInt	Intervalle de temps, en microsecondes
PHASE	OUT	UDInt	Déphasage, en microsecondes
STATUS	OUT	Word	Code d'état de l'alarme cyclique : <ul style="list-style-type: none"> <li>• Bits 0 à 4 : voir le tableau pour STATUS ci-dessous</li> <li>• Autres bits : toujours 0</li> </ul>

Tableau 8- 93 Paramètre STATUS

Bit	Valeur	Description
0	0	A l'état MARCHE de la CPU
	1	Pendant le démarrage
1	0	L'alarme est validée.
	1	L'alarme est inhibée via l'instruction DIS_IRT.
2	0	L'alarme n'est pas active ou a expiré.
	1	L'alarme est active.
4	0	L'OB identifié par OB_NR n'existe pas.
	1	L'OB identifié par OB_NR existe.
Autres bits		Toujours 0

Si une erreur se produit, RET\_VAL contient le code d'erreur approprié et le paramètre STATUS est égal à 0.

Tableau 8- 94 Paramètre RET\_VAL

RET_VAL (W#16#...)	Description
0000	Pas d'erreur
8090	L'OB n'existe pas ou a un type incorrect.
80B2	Aucun événement n'est associé à l'OB.

### 8.4.3 Alarmes temporisées

Vous pouvez démarrer et annuler le traitement des alarmes temporisées à l'aide des instructions SRT\_DINT et CAN\_DINT et interroger l'état des alarmes à l'aide de l'instruction QRY\_DINT. Chaque alarme temporisée est un événement unique qui se produit après le temps de retard indiqué. Si l'événement d'alarme temporisée est annulé avant que le temps de retard n'ait expiré, l'interruption du programme ne se produit pas.

Tableau 8- 95 Instructions SRT\_DINT, CAN\_DINT et QRY\_DINT

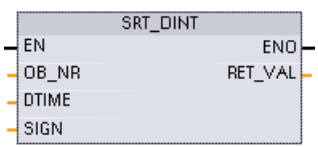
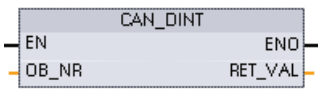
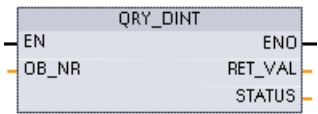
CONT/LOG	SCL	Description
	<pre>ret_val := SRT_DINT(   ob_nr:=_int_in_,   dtime:=_time_in_,   sign:=_word_in_);</pre>	SRT_DINT démarre une alarme temporisée qui exécute un OB lorsque le temps de retard indiqué par le paramètre DTIME s'est écoulé.
	<pre>ret_val := CAN_DINT(   ob_nr:=_int_in_);</pre>	CAN_DINT annule une alarme temporisée qui a déjà démarré. L'OB d'alarme temporisée n'est pas exécuté dans ce cas.
	<pre>ret_val := QRY_DINT(   ob_nr:=_int_in_,   status=&gt;_word_out_);</pre>	QRY_DINT interroge l'état de l'alarme temporisée indiquée par le paramètre OB_NR.

Tableau 8- 96 Types de données pour les paramètres

Paramètre et type		Type de données	Description
OB_NR	IN	OB_DELAY	Bloc d'organisation (OB) à démarrer après un temps de retard. Faites votre choix parmi les OB d'alarme temporisée disponibles qui ont été créés via la fonction "Ajouter nouveau bloc" de l'arborescence de projet. Double-cliquez sur le champ de paramètre, puis cliquez sur l'icône d'aide pour voir les OB disponibles.
DTIME <sup>1</sup>	IN	Time	Valeur du temps de retard (1 à 60000 ms)
SIGN <sup>1</sup>	IN	Word	Non utilisé par le S7-1200 : Toute valeur est acceptée. Il faut indiquer une valeur pour éviter les erreurs.
RET_VAL	OUT	Int	Code d'erreur d'exécution
STATUS	OUT	Word	Instruction QRY_DINT : Etat de l'OB d'alarme temporisée indiqué. Voir le tableau ci-dessous.

<sup>1</sup> Uniquement pour SRT\_DINT

## Fonctionnement

L'instruction SRT\_DINT indique un temps de retard, démarre la temporisation de temps de retard interne et associe un sous-programme d'OB d'alarme temporisée à l'événement d'expiration du temps de retard. Lorsque le temps de retard indiqué s'est écoulé, une interruption du programme est générée, ce qui déclenche l'exécution de l'OB d'alarme temporisée associé. Vous pouvez annuler une alarme temporisée en cours de traitement avant que le temps de retard indiqué n'ait expiré en exécutant l'instruction CAN\_DINT. Le nombre total d'événements d'alarme temporisée et d'alarme cyclique actifs ne doit pas dépasser quatre.

## Ajout de sous-programmes d'OB d'alarme temporisée à votre projet

Seuls des OB d'alarme temporisée peuvent être affectés aux instructions SRT\_DINT et CAN\_DINT. Aucun OB d'alarme temporisée n'existe dans un nouveau projet. Vous devez ajouter ces OB d'alarme temporisée à votre projet. Procédez comme suit pour créer un OB d'alarme temporisée :

1. Double-cliquez sur "Ajouter nouveau bloc" dans la branche "Blocs de programme" de l'arborescence de projet, sélectionnez "Bloc d'organisation (OB)", puis choisissez "Alarme temporisée".
2. Vous avez la possibilité de renommer l'OB, de sélectionner le langage de programmation ou de sélectionner le numéro de bloc. Commutez en numérotation manuelle si vous voulez utiliser un numéro de bloc différent de celui affecté automatiquement.
3. Editez le sous-programme de l'OB d'alarme temporisée et créez une réaction programmée que vous voulez exécuter lorsque l'événement d'alarme temporisée se produit. Vous pouvez appeler des FB et des FC à partir de l'OB d'alarme temporisée, avec une profondeur d'imbrication maximale de quatre.
4. Les noms des OB d'alarme temporisée nouvellement affectés seront disponibles lorsque vous éditez le paramètre OB\_NR des instructions SRT\_DINT et CAN\_DINT.

## Paramètre STATUS de QRY\_DINT :

Tableau 8- 97 En présence d'une erreur (REL\_VAL <> 0), STATUS = 0.

Bit	Valeur	Description
0	0	A l'état MARCHE
	1	Pendant le démarrage
1	0	L'alarme est validée.
	1	L'alarme est inhibée.
2	0	L'alarme n'est pas active ou a expiré.
	1	L'alarme est active.
4	0	Il n'existe pas d'OB de numéro OB_NR.
	1	Il existe un OB de numéro OB_NR.
Autres bits		Toujours 0

Codes d'erreur


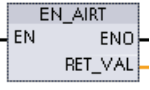
Tableau 8- 98 Codes d'erreur pour SRT\_DINT, CAN\_DINT et QRY\_DINT

RET_VAL (W#16#...)	Description
0000	Pas d'erreur
8090	Paramètre OB_NR incorrect
8091	Paramètre DTIME incorrect
80A0	L'alarme temporisée n'a pas démarré.

8.4.4 Alarmes asynchrones

Utilisez les instructions DIS\_AIRT et EN\_AIRT pour inhiber et valider le traitement des alarmes.

Tableau 8- 99 Instructions DIS\_AIRT et EN\_AIRT

CONT/LOG	SCL	Description
	DIS_AIRT ( ) ;	DIS_AIRT retarde le traitement de nouveaux événements d'alarme. Vous pouvez exécuter DIS_AIRT plus d'une fois dans un OB.
	EN_AIRT ( ) ;	EN_AIRT valide le traitement d'événements d'alarme que vous aviez préalablement inhibé à l'aide de l'instruction DIS_AIRT. Chaque exécution de DIS_AIRT doit être annulée par une exécution de EN_AIRT.  Les exécutions de EN_AIRT doivent se produire dans le même OB ou dans toute FC ou tout FB appelé par cet OB pour que les alarmes soient réactivées pour cet OB.

 ATTENTION

Si le temps de filtre pour une voie d'entrée TOR est modifié par rapport à un paramétrage précédent, une nouvelle valeur d'entrée de niveau "0" peut devoir être présente pendant une durée cumulée allant jusqu'à 20,0 ms pour que le filtre réagisse pleinement aux nouvelles entrées. Pendant ce temps, les événements d'impulsion "0" courts de moins de 20,0 ms peuvent ne pas être détectés ni comptés.

Cette modification des temps de filtre peut provoquer un fonctionnement inattendu des machines ou du processus, pouvant entraîner la mort, des blessures graves et des dommages matériels importants.

Pour garantir la prise en compte immédiate d'un nouveau temps de filtre, mettez la CPU hors tension puis sous tension.

Tableau 8- 100 Types de données pour les paramètres

Paramètre et type		Type de données	Description
RET_VAL	OUT	Int	Nombre de retards = nombre d'exécutions DIS_AIRT dans la file d'attente

Les exécutions de DIS\_AIRT sont comptabilisées par le système d'exploitation. Chacune d'elles reste en vigueur jusqu'à ce qu'elle soit expressément annulée par une instruction EN\_AIRT ou jusqu'à ce que l'OB en cours ait été intégralement traité. Si, par exemple, vous avez désactivé les alarmes cinq fois via cinq exécutions de DIS\_AIRT, vous devez annuler ces dernières avec cinq exécutions de EN\_AIRT pour que le traitement des alarmes soit à nouveau activé.

Une fois les événements d'alarme réactivés, les alarmes qui se sont produites alors que DIS\_AIRT était en vigueur sont traitées ou les alarmes sont traitées dès que l'OB en cours a été exécuté.

Le paramètre RET\_VAL indique le nombre de fois où le traitement des alarmes a été inhibé, ce nombre correspondant au nombre d'exécutions de DIS\_AIRT en file d'attente. Le traitement des alarmes n'est réactivé que lorsque le paramètre RET\_VAL est égal à 0.

## 8.5 Diagnostic (PROFINET ou PROFIBUS)

### 8.5.1 Opérations de diagnostic

Vous pouvez utiliser les instructions de diagnostic suivantes avec PROFINET ou PROFIBUS :

- Instruction GET\_DIAG (Page 322) : Vous pouvez lire les informations de diagnostic du périphérique spécifié.
- Instruction DeviceStates (Page 320) : Vous pouvez interroger les états de fonctionnement d'un périphérique IO dans un sous-réseau I/O.
- Instruction ModuleStates (Page 321) : Vous pouvez interroger les états de fonctionnement des modules d'un périphérique IO décentralisé.
- Instruction LED (Page 318) : Vous pouvez lire l'état des LED d'un périphérique IO décentralisé.

### 8.5.2 Événements de diagnostic d'une périphérie décentralisée

---

#### Remarque

Dans un réseau PROFIBUS IO, la CPU passe à l'état MARCHE après un chargement ou une mise hors tension puis sous tension à moins que la compatibilité matérielle ne soit configurée pour autoriser des modules de remplacement compatibles (Page 131) et qu'un ou plusieurs modules manquent ou ne soient pas compatibles avec le module configuré.

---

8.5 Diagnostic (PROFINET ou PROFIBUS)

Comme le montre le tableau suivant, la CPU prend en charge le diagnostic pouvant être configuré pour les composants du système de périphérie décentralisé. Chacune de ces erreurs génère une entrée dans le tampon de diagnostic.

Tableau 8- 101 Gestion des événements de diagnostic pour PROFINET et PROFIBUS

Type d'erreur	Information de diagnostic pour la station ?	Entrée dans le tampon de diagnostic ?	Etat de fonctionnement de la CPU
Erreur de diagnostic	Oui	Oui	reste à l'état Marche
Défaillance du châssis ou de la station	Oui	Oui	reste à l'état Marche
Erreur d'accès aux E/S <sup>1</sup>	Non	Oui	reste à l'état Marche
Erreur d'accès à la périphérie <sup>2</sup>	Non	Oui	reste à l'état Marche
Evénement débrogage/enfichage	Oui	Oui	reste à l'état Marche

- <sup>1</sup> Exemple d'erreur d'accès aux E/S : un module a été débrogé.
- <sup>2</sup> Exemple d'erreur d'accès à la périphérie : communication acyclique avec un sous-module qui ne transmet pas.

Utilisez l'instruction GET\_DIAG (Page 322) pour chaque station afin d'obtenir les informations de diagnostic. Ceci vous permet de traiter les erreurs/défauts constatés sur l'appareil via le programme et de commuter, si vous le souhaitez, la CPU sur Arrêt. Cette méthode requiert la spécification de l'appareil à partir duquel les informations d'état doivent être lues.

L'instruction GET\_DIAG utilise l'adresse "L" (LADDR) de la station pour lire le diagnostic de toute la station. Vous trouvez cette adresse L dans la vue de configuration de réseau après avoir sélectionné le châssis entier de la station (zone grise dans sa totalité), l'adresse L est affichée dans l'onglet Propriétés de la station. Vous trouvez le paramètre LADDR pour chaque module individuel soit dans les propriétés du module (dans la configuration de l'appareil), soit dans la table de variables par défaut de la CPU.

8.5.3 Instruction LED

Tableau 8- 102 Instruction LED

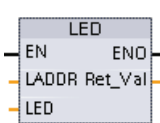
CONT/LOG	SCL	Description
	<pre>ret_val := LED(     laddr:=_word_in_,     LED:=_uint_in_);</pre>	<p>Utilisez l'instruction LED pour lire l'état des DEL sur une CPU ou une interface. L'état de la DEL indiquée est renvoyé dans la sortie RET_VAL.</p>

Tableau 8- 103 Types de données pour les paramètres

Paramètre et type		Type de données	Description		
LADDR	IN	HW_IO	Numéro d'identification de la CPU ou de l'interface <sup>1</sup>		
LED	IN	UInt	Numéro identificateur de la DEL		
			1	RUN/STOP	Couleur 1 = vert, couleur 2 = jaune
			2	Défaut	Couleur 1 = rouge
			3	Maintenance	Couleur 1 = jaune
			4	Redondance	Sans objet
			5	Lien	Couleur 1 = vert
6	Tx/Rx	Couleur 1 = jaune			
RET_VAL	OUT	Int	Etat de la DEL		

<sup>1</sup> Vous pouvez, par exemple, sélectionner la CPU ("PLC\_1", par exemple) ou l'interface PROFINET dans la liste déroulante du paramètre.

Tableau 8- 104 Etat de RET\_VAL

RET_VAL (W#16#...)	Description	
Etat de DEL 0 à 9	0	La DEL n'existe pas.
	1	La DEL est éteinte.
	2	La DEL est allumée en couleur 1 (feu fixe).
	3	La DEL est allumée en couleur 2 (feu fixe).
	4	La DEL clignote en couleur 1 à 2 Hz.
	5	La DEL clignote en couleur 2 à 2 Hz.
	6	La DEL clignote alternativement en couleur 1 et 2 à 2 Hz.
	7	La DEL est allumée en couleur 1 (Tx/Rx).
	8	La DEL est allumée en couleur 2 (Tx/Rx).
9	L'état de la DEL n'est pas disponible.	
8091	L'appareil identifié par LADDR n'existe pas.	
8092	L'appareil identifié par LADDR ne comporte pas de DEL.	
8093	L'identificateur de DEL n'est pas défini.	
80Bx	La CPU identifiée par LADDR ne prend pas en charge l'instruction LED.	

### 8.5.4 Instruction DeviceStates

Tableau 8- 105 Instruction DeviceStates

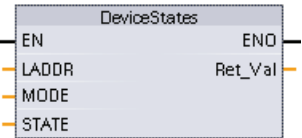
CONT/LOG	SCL	Description
	<pre>ret_val := DeviceStates(     laddr:=hw_io_in_,     mode:=_uint_in_,     state:=_variant_inout_);</pre>	DeviceStates permet d'extraire les états de fonctionnement des périphériques IO d'un sous-système IO. Après l'exécution, le paramètre STATE contient l'état d'erreur de chaque périphérique IO dans une liste de bits (pour les paramètres LADDR et MODE affectés). Cette information correspond à l'état des appareils présenté dans la vue de diagnostic de STEP 7.

Tableau 8- 106 Types de données pour les paramètres

Paramètre et type	Type de données	Description
LADDR      IN	HW_IOSYSTEM	Adresse logique (identificateur pour le système IO)
MODE	UInt	Type d'état : <ul style="list-style-type: none"> <li>• 1 : La configuration de l'appareil est active ou n'est pas encore achevée.</li> <li>• 2 : L'appareil est défectueux.</li> <li>• 3 : L'appareil est désactivé.</li> <li>• 4 : L'appareil existe.</li> </ul>
RET_VAL	Int	Code d'erreur d'exécution
STATE <sup>1</sup> InOut	VARIANT	Mémoire tampon qui reçoit l'état d'erreur de chaque appareil. Vous pouvez choisir n'importe quel type de données binaire (Bool, Byte, Word ou DWord) ou un tableau d'éléments de type binaire comme type de données pour le paramètre STATE. <ul style="list-style-type: none"> <li>• Bit récapitulatif : bit 0 =1 si l'un des bits d'état des périphériques IO est à 1.</li> <li>• Bit d'état : état du périphérique IO de numéro de station n en fonction du mode sélectionné. Par exemple, MODE = 2 et bit 3 = 1 signifient que la station 3 est défaillante.</li> </ul>

<sup>1</sup> Pour PROFIBUS DP, la longueur des informations d'état est de 128 bits. Pour PROFINET IO, cette longueur est de 1024 bits.

Après l'exécution, le paramètre STATE contient l'état d'erreur de chaque périphérique IO sous forme de liste de bits (pour les paramètres LADDR et MODE affectés).

Tableau 8- 107 Codes d'erreur

RET_VAL (W#16#...)	Description
0	Pas d'erreur
8091	LADDR n'existe pas.



RET_VAL (W#16#...)	Description
8092	LADDR ne correspond pas à un système IO.
8093	Type de données invalide affecté au paramètre STATE. Les types de données valides sont Bool, Byte, Word ou Dword ou un tableau de Bool, Byte, Word ou Dword.
80Bx	L'instruction DeviceStates n'est pas prise en charge par la CPU pour cette adresse LADDR.
8452	Les données d'état complètes sont trop grandes pour le paramètre STATE affecté. La mémoire tampon STATE contient un résultat partiel.

## 8.5.5 Instruction ModuleStates

Tableau 8- 108 Instruction ModuleStates

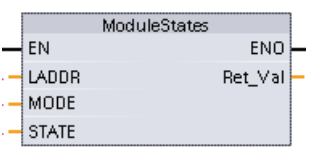
CONT/LOG	SCL	Description
	<pre>ret_val := ModuleStates(   laddr:=_word_in_,   mode:=_uint_in_,   state:=_variant_inout);</pre>	<p>ModuleStates permet d'extraire les états de fonctionnement des modules d'E/S. Après l'exécution, le paramètre STATE contient l'état d'erreur de chaque module d'E/S dans une liste de bits (pour les paramètres LADDR et MODE affectés). Cette information correspond à l'état des modules présenté dans la vue de diagnostic de STEP 7.</p>

Tableau 8- 109 Types de données pour les paramètres

Paramètre et type		Type de données	Description
LADDR	IN	HW_DEVICE	Adresse logique (identificateur pour les modules d'E/S)
MODE	IN	UInt	Type d'état : <ul style="list-style-type: none"> <li>• 1 : La configuration du module est active ou n'est pas encore achevée.</li> <li>• 2: Module défaillant</li> <li>• 3 : Le module est désactivé.</li> <li>• 4 : Le module existe.</li> </ul>
RET_VAL	OUT	Int	Etat (code d'erreur)
STATE <sup>1</sup>	InOut	Variant	Mémoire tampon qui reçoit l'état d'erreur de chaque module. Vous pouvez choisir n'importe quel type de données binaire (Bool, Byte, Word ou DWord) ou un tableau d'éléments de type binaire comme type de données pour le paramètre STATE. <ul style="list-style-type: none"> <li>• Bit récapitulatif : bit 0 =1 si l'un des bits d'état des modules d'E/S est à 1.</li> <li>• Bit d'état : état du module d'E/S de numéro d'emplacement <i>n</i> en fonction du mode sélectionné. Par exemple, MODE = 2 et bit 3 = 1 signifient que la station 3 est défaillante.</li> </ul>

<sup>1</sup> 128 bits au maximum peuvent être affectés. Le nombre de bits requis dépend de votre utilisation des modules d'E/S.

Tableau 8- 110 Codes d'erreur


RET_VAL ( W#16#...)	Description
0	Pas d'erreur
8091	Le module identifié par LADDR n'existe pas.
8092	Le module identifié par LADDR ne correspond pas à un périphérique IO.
8093	Type de données invalide pour le paramètre STATE. Les types de données valides sont Bool, Byte, Word ou Dword ou un tableau de Bool, Byte, Word ou Dword.
80Bx	L'instruction ModuleStates n'est pas prise en charge par cette CPU pour cette adresse LADDR.
8452	Les données d'état complètes sont trop grandes pour le paramètre STATE affecté. La mémoire tampon STATE contient un résultat partiel.

### 8.5.6 Instruction GET\_DIAG

#### Description

L'instruction GET\_DIAG permet de lire les informations de diagnostic d'un objet matériel. Vous sélectionnez l'objet matériel à l'aide du paramètre LADDR. Le paramètre MODE permet de sélectionner les informations de diagnostic à lire.

Tableau 8- 111 Instruction GET\_DIAG

CONT/LOG	SCL	Description
	<pre>ret_val := GET_DIAG(     mode:=_uint_in_,     laddr:=_word_in_,     cnt_diag=&gt;_uint_out_,     diag:=_variant_inout_,     detail:=_variant_inout_);</pre>	Permet de lire les informations de diagnostic du périphérique spécifié.

#### Paramètres

Le tableau suivant présente les paramètres de l'instruction GET\_DIAG :

Tableau 8- 112 Types de données pour les paramètres

Paramètre et type	Type de données	Description
MODE IN	UInt	Le paramètre MODE permet de sélectionner les données de diagnostic à transmettre.
LADDR IN	HW_ANY (Word)	ID matériel de l'appareil
RET_VAL OUT	Int	Etat de l'instruction

Paramètre et type		Type de données	Description
CNT_DIAG	OUT	UInt	Nombre de détails de diagnostic transmis
DIAG	InOut	Variant	Pointeur désignant la zone de données pour le stockage des informations de diagnostic du mode sélectionné
DETAILS	InOut	Variant	Pointeur désignant la zone de données pour le stockage des détails de diagnostic conformément au mode sélectionné

## Paramètre MODE

Selon la valeur du paramètre MODE, des données de diagnostic différentes sont transmises aux paramètres de sortie DIAG, CNT\_DIAG et DETAILS :

Tableau 8- 113 Paramètre MODE

MODE	Description	DIAG	CNT_DIAG	DETAILS
0	Transmission de toutes les informations de diagnostic prises en charge pour un module sous forme de double mot (DWord), bit X=1 indiquant que le mode X est pris en charge.	Chaîne de bits des modes pris en charge sous forme de double mot (DWord), bit X=1 indiquant que le mode X est pris en charge.	0	-
1	Transmission de l'état inhérent de l'objet matériel adressé.	Etat de diagnostic : Transmission conformément à la structure DIS. (Remarque : voir les informations sur la structure DIS ci-après et l'exemple d'instruction GET_DIAG à la fin du paragraphe)	0	-
2	Transmission de l'état de tous les modules subordonnés de l'objet matériel adressé.	Transmission des données de diagnostic conformément à la structure DNN. (Remarque : voir les informations sur la structure DNN ci-après et l'exemple d'instruction GET_DIAG à la fin du paragraphe)	0	Informations d'état du module conformément à la structure DiagnosticsDetails.

### Structure DIS

Lorsque le paramètre MODE a la valeur 1, les informations de diagnostic sont transmises conformément à la structure DIS. Le tableau suivant donne la signification des différents paramètres :

Tableau 8- 114 Structure DIS (source d'informations de diagnostic)

Paramètre	Type de données	Valeur	Description
MaintenanceState	DWord	Enum	
		0	Pas de maintenance nécessaire
		1	Le module ou l'appareil est désactivé.
		2	-
		3	-
		4	-
		5	Maintenance nécessaire
		6	Requête de maintenance
		7	Erreur
		8	Etat inconnu / erreur dans le module subordonné
		9	-
Componentstate Detail	DWord	Tableau de bits	Etat des sous-modules du module : <ul style="list-style-type: none"> <li>• Bits 0 à 15 : Message d'état du module</li> <li>• Bits 16 à 31 : Message d'état de la CPU</li> </ul>
		0 à 2 (enum)	Information supplémentaire : <ul style="list-style-type: none"> <li>• Bit 0 : Pas d'information supplémentaire</li> <li>• Bit 1 : Transfert interdit</li> </ul>
		3	Bit 3 = 1 : Au moins une voie prend en charge des qualifieurs pour le diagnostic.
		4	Bit 4 = 1: Maintenance nécessaire pour au moins une voie ou un composant
		5	Bit 5 = 1: Requête de maintenance pour au moins une voie ou un composant
		6	Bit 6 = 1: Erreur dans au moins une voie ou un composant
		7 à 10	-
		11 à 14	Bit 11 = 1 : PNIO, sous-module correct Bit 12 = 1 : PNIO, module de remplacement Bit 13 = 1 : PNIO, module incorrect Bit 14 = 1 : PNIO, module déconnecté
15	-		

Paramètre	Type de données	Valeur	Description
		16 à 31	Information d'état pour modules générée par la CPU : Bit 16 = 1 : Module désactivé Bit 17 = 1 : Opération CiR active Bit 18 = 1 : Entrée non disponible Bit 19 = 1 : Sortie non disponible Bit 20 = 1 : Débordement de la mémoire tampon de diagnostic Bit 21 = 1 : Diagnostic non disponible Bits 22 - 31 : Réservés (toujours = 0)
OwnState	Uint16	Enum	La valeur du paramètre OwnState décrit l'état de maintenance du module.
		0	Pas d'erreur
		1	Le module ou l'appareil est désactivé.
		2	Maintenance nécessaire
		3	Requête de maintenance
		4	Erreur
		5	Le module ou l'appareil n'est pas accessible depuis la CPU (vaut pour les modules et appareils en dessous d'une CPU).
		6	Les entrées/sorties ne sont pas disponibles.
IO State	Uint16	Tableau de bits	Etat d'E/S du module
		0	Bit 0 = 1 : Pas de maintenance nécessaire
		1	Bit 1 = 1 : Le module ou l'appareil est désactivé.
		2	Bit 2 = 1 : Maintenance nécessaire
		3	Bit 3 = 1 : Requête de maintenance
		4	Bit 4 = 1 : Erreur
		5	Bit 5 = 1 : Le module ou l'appareil n'est pas accessible depuis la CPU (vaut pour les modules et appareils en dessous d'une CPU).
		6	Les entrées/sorties ne sont pas disponibles.
		7	Qualifieur ; bit 7 = 1 si les bits 0, 2 ou 3 sont à 1
8 à 15	Réservés (toujours = 0)		
OperatingState	UInt16	Enum	
		0	-
		1	A l'état ARRET / mise à jour du firmware
		2	A l'état ARRET / effacement général de la mémoire
		3	A l'état ARRET / auto-démarrage
		4	A l'état ARRET
		5	Effacement général de la mémoire
		6	A l'état MISE EN ROUTE
		7	A l'état MARCHE
		8	-
9	A l'état ATTENTE		

Paramètre	Type de données	Valeur	Description
		10	-
		11	-
		12	Module défaillant
		13	-
		14	Absence d'alimentation
		15	CiR
		16	A l'état ARRET / sans DIS
		17	A l'état
		18	
		19	
		20	

### Structure DiagnosticsDetail

Lorsque le paramètre MODE a la valeur 2, les détails de diagnostic sont transmis conformément à la structure DiagnosticsDetail. Le tableau suivant donne la signification des différents paramètres :

Tableau 8- 115 Structure DiagnosticsDetail

Paramètre	Type de données	Description
ChannelNumber	UInt	Numéro de voie
Properties	Word	
ALID	UInt	Identification de l'alarme
Qualifier	DWord	Qualifieur des données de diagnostic
ErrorType	UDInt	Type d'erreur de voie
ExtErrorType	UDInt	Type d'erreur de voie étendu
AddValue_1	UInt	Valeur supplémentaire
AddValue_2	UInt	Valeur supplémentaire
AddValue_3	UInt	Valeur supplémentaire
AddValue_4	UInt	Valeur supplémentaire

## Structure DNN

Lorsque le paramètre MODE a la valeur 2, les détails de diagnostic sont transmis conformément à la structure DNN. Le tableau suivant donne la signification des différents paramètres :

Tableau 8- 116 Structure DNN (noeud de navigation de diagnostic)

Paramètre	Type de données	Valeur	Description
SubordinateState	UINT	Enum	Etat du module subordonné (voir paramètre OwnState de la structure DIS)
SubordinateIOState	WORD	Bitarray	Etat des entrées et sorties du module subordonné (voir paramètre IO State de la structure DIS)
DNNmode	WORD	Bitarray	<ul style="list-style-type: none"> <li>• Bit 0 = 0 : Diagnostic activé</li> <li>• Bit 0 = 1 : Diagnostic désactivé</li> <li>• Bits 1 à 15 : Réservés</li> </ul>

## Paramètre RET\_VAL

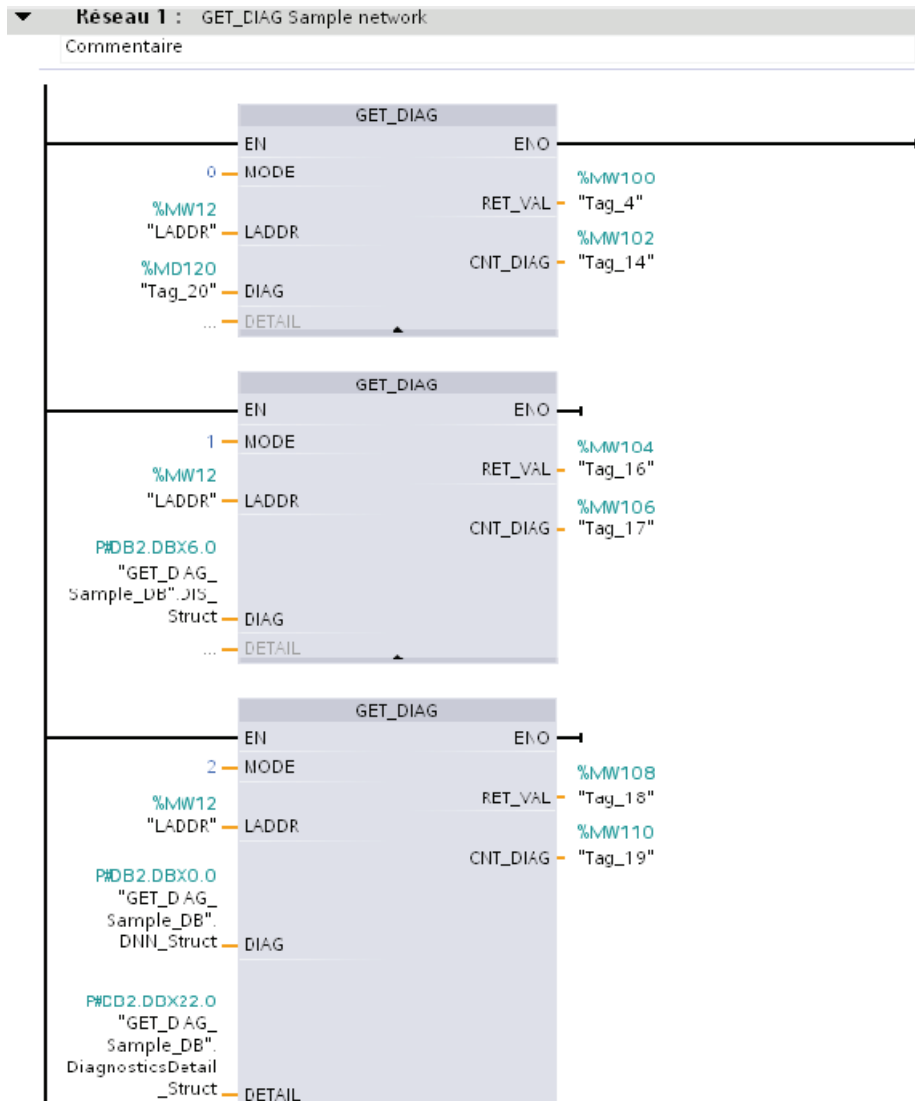
Tableau 8- 117 Codes d'erreur du paramètre RET\_VAL

Code d'erreur (W#16#...)	Description
0	Pas d'erreur
n	La zone de données dans le paramètre DETAILS est trop petite. Les détails des données de diagnostic ne peuvent pas tous être transmis.
8080	La valeur dans le paramètre MODE n'est pas prise en charge.
8081	Le type dans le paramètre DIAG n'est pas pris en charge pour le mode sélectionné (paramètre MODE).
8082	Le type dans le paramètre DETAILS n'est pas pris en charge pour le mode sélectionné (paramètre MODE).
8090	LADDR n'existe pas.
8091	La voie sélectionnée dans le paramètre CHANNEL n'existe pas.
80C1	Ressources insuffisantes pour une exécution en parallèle

Exemple

Le réseau CONT et le DB ci-après montrent comment utiliser les trois modes avec les trois structures :

- DIS
- DiagnosticsDetail
- DNN





GET_DIAG_Sample_DB							
	Nom	Type de données	Déc...	Vale... de...	Rém...	Visible dans...	Commentaire
1	Static						
2	DNN_Struct	DNN	0.0		<input type="checkbox"/>	<input checked="" type="checkbox"/>	
3	SubordinateState	UInt	0.0	0	<input type="checkbox"/>	<input checked="" type="checkbox"/>	
4	SubordinateIOState	Word	2.0	0	<input type="checkbox"/>	<input checked="" type="checkbox"/>	
5	DNNmode	Word	4.0	0	<input type="checkbox"/>	<input checked="" type="checkbox"/>	
6	DIS_Struct	DIS	6.0		<input type="checkbox"/>	<input checked="" type="checkbox"/>	
7	MaintenanceState	DWord	0.0	0	<input type="checkbox"/>	<input checked="" type="checkbox"/>	
8	ComponentStateDetail	DWord	4.0	0	<input type="checkbox"/>	<input checked="" type="checkbox"/>	
9	OwnState	UInt	8.0	0	<input type="checkbox"/>	<input checked="" type="checkbox"/>	
10	IOState	Word	10.0	0	<input type="checkbox"/>	<input checked="" type="checkbox"/>	
11	OperatingState	UInt	12.0	0	<input type="checkbox"/>	<input checked="" type="checkbox"/>	
12	DiagnosticsDetail_Struct	DiagnosticsDetail	22.0		<input type="checkbox"/>	<input checked="" type="checkbox"/>	
13	ChannelNumber	UInt	0.0	0	<input type="checkbox"/>	<input checked="" type="checkbox"/>	Channel number
14	Properties	Word	2.0	0	<input type="checkbox"/>	<input checked="" type="checkbox"/>	
15	ALID	UInt	4.0	0	<input type="checkbox"/>	<input checked="" type="checkbox"/>	Alarm identifier
16	Qualifier	DWord	8.0	0	<input type="checkbox"/>	<input checked="" type="checkbox"/>	
17	ErrorType	UDInt	12.0	0	<input type="checkbox"/>	<input checked="" type="checkbox"/>	Channel error type
18	ExtErrorType	UDInt	16.0	0	<input type="checkbox"/>	<input checked="" type="checkbox"/>	extended channel error type
19	AddValue_1	UInt	20.0	0	<input type="checkbox"/>	<input checked="" type="checkbox"/>	additional value
20	AddValue_2	UInt	22.0	0	<input type="checkbox"/>	<input checked="" type="checkbox"/>	additional value
21	AddValue_3	UInt	24.0	0	<input type="checkbox"/>	<input checked="" type="checkbox"/>	additional value
22	AddValue_4	UInt	26.0	0	<input type="checkbox"/>	<input checked="" type="checkbox"/>	additional value

- ① DNN
- ② DIS
- ③ DiagnosticsDetail

**Remarque**


Dans le DB, vous devez saisir manuellement le type de données pour accéder à chacune des trois structures ; il n'y a pas de sélection par liste déroulante. Saisissez les types de données exactement comme indiqué ci-dessous :

- DNN
- DIS
- DiagnosticsDetail

## 8.6 Impulsion

### 8.6.1 Instruction CTRL\_PWM

Tableau 8- 118 Instruction CTRL\_PWM (Modulation de largeur d'impulsion)

CONT/LOG	SCL	Description
 <p>Diagramme de l'instruction CTRL_PWM. Les entrées sont EN, PWM et ENABLE. Les sorties sont END, BUSY et STATUS.</p>	<pre>"CTRL_PWM_DB" (     PWM:=_word_in_,     enable:=_bool_in_,     busy=&gt;_bool_out_,     status=&gt;_word_out_);</pre>	<p>Fournit une sortie à période fixe avec un rapport cyclique variable. La sortie PWM s'exécute en continu une fois qu'elle a été lancée à la fréquence indiquée (période). On fait varier la largeur d'impulsion de la manière nécessaire pour influencer sur la commande souhaitée.</p>

1 STEP 7 crée automatiquement le DB lorsque vous insérez l'instruction.

2 Dans l'exemple SCL, "CTRL\_PWM\_DB" est le nom du DB d'instance.

#### CTRL\_HSC

Tableau 8- 119 Types de données pour les paramètres

Paramètre et type		Type de données	Description
PWM	IN	HW_PWM (Word)	Identificateur PWM : les noms des générateurs d'impulsions activés deviennent des variables dans la table de variables "Constantes" et sont disponibles pour utilisation comme paramètre PWM. (valeur par défaut : 0) 0)
ENABLE	IN	Bool	1=démarrer le générateur d'impulsions 0 = arrêter le générateur d'impulsions
BUSY	OUT	Bool	Fonction occupée (valeur par défaut : 0) 0)
STATUS	OUT	Word	Code d'erreur d'exécution (valeur par défaut : 0) 0)

L'instruction CTRL\_PWM stocke les informations de paramètres dans le DB. L'utilisateur ne peut pas modifier séparément les paramètres du bloc de données qui sont gérés par l'instruction CTRL\_PWM.

Indiquez le générateur d'impulsions activé à utiliser en spécifiant son nom de variable comme paramètre PWM.

Lorsque l'entrée EN est à VRAI, l'instruction PWM\_CTRL démarre ou arrête le générateur d'impulsions PWM identifié conformément à la valeur de l'entrée ENABLE. La durée de l'impulsion est précisée par la valeur dans l'adresse de mot de sortie Q associée.

Etant donné que la CPU traite la requête une fois que l'instruction CTRL\_PWM est exécutée, le paramètre BUSY signalera toujours FAUX. Si une erreur est détectée, ENO est mis à FAUX et le paramètre STATUS contient un code d'erreur.

La largeur d'impulsion est définie à la valeur initiale paramétrée dans la configuration des appareils lorsque la CPU passe à l'état MARCHE. Vous écrivez des valeurs dans l'adresse de mot Q indiquée dans la configuration des appareils (Adresses de sortie / Adresse de départ) selon vos besoins pour modifier la durée d'impulsion. Vous utilisez une instruction de transfert, de conversion, mathématique ou une boîte PID pour écrire la durée d'impulsion désirée dans le mot Q approprié. Vous devez utiliser la plage valide de la valeur de mot Q (pourcentage, millièmes, dix millièmes ou format analogique S7).

#### Remarque

#### Les E/S TOR affectées à PWM et PTO ne peuvent pas être forcées

Les E/S TOR utilisées par la modulation de largeur d'impulsion (PWM) et la sortie de trains d'impulsions (PTO) sont affectées pendant la configuration des appareils. Lorsque des adresses d'E/S TOR sont affectées à ces appareils, les valeurs dans les adresses d'E/S affectées ne peuvent pas être forcées par la fonction de forçage permanent de la table de visualisation.

Tableau 8- 120 Valeurs du paramètre STATUS

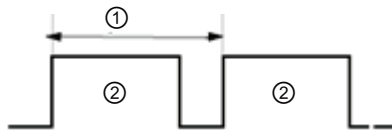
STATUS	Description
0	Pas d'erreur
80A1	L'identificateur PWM n'accède pas à un nom de générateur d'impulsions valide.

Tableau 8- 121 Codes d'erreur communs

Code d'erreur <sup>1</sup>	Description
8022	Zone trop petite pour l'entrée
8023	Zone trop petite pour la sortie
8024	Zone d'entrée illicite
8025	Zone de sortie illicite
8028	Affectation illicite de bit d'entrée
8029	Affectation illicite de bit de sortie
8030	La zone de sortie est un DB en lecture seule.
803A	DB inexistant

<sup>1</sup> Si l'une des erreurs communes se produit pendant l'exécution d'un bloc de code, la CPU passe à l'état ARRET à moins que vous n'ayez utilisé l'instruction GetError ou GetErrorID dans ce bloc de code pour créer une réaction programmée à l'erreur.

### 8.6.2 Fonctionnement des sorties d'impulsions



- ① Temps de cycle
- ② Largeur d'impulsion

La largeur d'impulsion peut être exprimée sous forme de centièmes de la période (0 à 100), de millièmes (0 à 1000), de dix millièmes (0 à 10000) ou de format analogique S7.

La largeur d'impulsion peut varier de 0 (pas d'impulsion, toujours désactivée) à la pleine échelle (pas d'impulsion, toujours activée).

Comme la sortie PWM peut varier de 0 à la pleine échelle, elle fournit une sortie TOR qui est, en de nombreuses façons, identique à une sortie analogique. Vous pouvez, par exemple, utiliser cette sortie PWM pour commander la vitesse d'un moteur de l'arrêt à la pleine vitesse ou pour commander la position d'une soupape de "fermée" à "complètement ouverte".

Deux générateurs d'impulsions sont disponibles pour commander les fonctions de sortie d'impulsions rapides : PWM et PTO (Sortie de trains d'impulsions). PTO est utilisé par les instructions de commande de mouvement. Vous pouvez affecter chaque générateur d'impulsions à PWM ou à PTO, mais pas aux deux en même temps.

Les deux générateurs d'impulsions sont mappés sur des sorties TOR spécifiques comme décrit dans le tableau suivant. Vous pouvez utiliser les sorties CPU intégrées ou les sorties du Signal Board optionnel. Les numéros des sorties sont indiqués dans le tableau suivant (sur la base de la configuration par défaut des sorties). Si vous avez modifié la numérotation des sorties, les numéros des sorties seront ceux que vous avez définis. Quoi qu'il en soit, PTO1/PWM1 utilise les deux premières sorties TOR et PTO2/PWM2 utilise les deux sorties TOR suivantes, sur la CPU ou sur le Signal Board enfiché. Notez que PWM ne nécessite qu'une sortie alors que PTO peut optionnellement utiliser deux sorties par voie. Si une sortie n'est pas utilisée pour une fonction d'impulsion, elle est disponible pour d'autres usages.

**IMPORTANT**

**Les sorties de trains d'impulsions ne peuvent pas être utilisées par d'autres instructions dans le programme utilisateur**

Lorsque vous configurez les sorties de la CPU ou du Signal Board en tant que générateurs d'impulsions (pour les instructions PWM ou de commande de mouvement), les adresses des sorties correspondantes (Q0.0, Q0.1, Q4.0 et Q4.1) sont supprimées de la mémoire Q et ne peuvent pas être utilisées à d'autres fins dans le programme utilisateur. Si votre programme utilisateur écrit une valeur dans une sortie utilisée comme générateur d'impulsions, la CPU n'écrit pas cette valeur dans la sortie physique.

Tableau 8- 122 Affectations par défaut des sorties aux générateurs d'impulsions

Description	Impulsion	Sens
PTO 0		
E/S intégrées	Q0.0	Q0.1
E/S du SB	Q4.0	Q4.1
PWM 0		
Sorties intégrées	Q0.0	-
Sorties du SB	Q4.0	-
PTO 1		
E/S intégrées	Q0.2	Q0.3
E/S du SB	Q4.2	Q4.3
PWM 1		
Sorties intégrées	Q0.2	-
Sorties du SB	Q4.2	-
PTO 2		
E/S intégrées	Q0.4 <sup>1</sup>	Q0.5 <sup>1</sup>
E/S du SB	Q4.0	Q4.1
PWM 2		
Sorties intégrées	Q0.4 <sup>1</sup>	-
Sorties du SB	Q4.1	-
PTO 3		
E/S intégrées	Q0.6 <sup>2</sup>	Q0.7 <sup>2</sup>
E/S du SB	Q4.2	Q4.3
PWM 3		
Sorties intégrées	Q0.6 <sup>2</sup>	-
Sorties du SB	Q4.3	-

<sup>1</sup> La CPU 1211C ne comporte pas de sorties Q0.4, Q0.5, Q0.6 et Q0.7. Ces sorties ne peuvent donc pas être utilisées dans la CPU 1211C.

<sup>2</sup> La CPU 1212C ne comporte pas de sorties Q0.6 et Q0.7. Ces sorties ne peuvent donc pas être utilisées dans la CPU 1212C.

<sup>3</sup> Ce tableau s'applique aux fonctions PTO/PWM des CPU 1211C, CPU 1212C, CPU 1214C et CPU 1215C.

### 8.6.3 Configuration d'une voie d'impulsion pour PWM

Pour préparer le fonctionnement de PWM, configurez d'abord une voie d'impulsion dans la configuration des appareils en sélectionnant la CPU, le générateur d'impulsions (PTO/PWM) et soit PWM1 soit PWM2. Activez le générateur d'impulsions (case à cocher). Si un générateur d'impulsions est activé, un nom par défaut unique est affecté à ce générateur d'impulsions particulier. Vous pouvez modifier ce nom en l'éditant dans la boîte d'édition "Nom :", mais ce doit être un nom unique. Les noms des générateurs d'impulsions activés deviennent des variables dans la table de variables "Constantes" et sont disponibles pour utilisation comme paramètre PWM de l'instruction CTRL\_PWM.

**IMPORTANT**

La fréquence d'impulsion maximale des générateurs d'impulsions pour les sorties TOR est de 100 kHz (pour la CPU), de 20 kHz (pour un SB) et de 200 kHz (pour un SB rapide). Toutefois, STEP 7 ne vous avertit pas si vous configurez un axe avec une vitesse ou une fréquence maximale qui dépasse cette limitation matérielle. Afin d'éviter tout problème avec votre application, assurez-vous toujours de ne pas dépasser la fréquence d'impulsion maximale du matériel.

Vous pouvez renommer le générateur d'impulsions, ajouter un commentaire et affecter des paramètres comme suit.

- Générateur d'impulsions utilisé comme suit : PWM ou PTO (choisissez PWM)
- Source de sortie : Intégrée à la CPU ou SB
- Unité de temps : millisecondes ou microsecondes
- Format de durée d'impulsion :
  - Centièmes (0 à 100)
  - Millièmes (0 à 1000)
  - Dix millièmes (0 à 10000)
  - Format analogique S7 (0 à 27648)
- Période : Entrez la valeur de la période. Cette valeur ne peut être modifiée que dans la Configuration des appareils.
- Durée d'impulsion initiale : Entrez la valeur initiale de la durée d'impulsion. La valeur de la durée d'impulsion peut être modifiée pendant l'exécution.

Entrez l'adresse de début pour configurer les adresses de sortie. Entrez l'adresse de mot Q dans laquelle vous voulez placer la valeur de durée d'impulsion.

**IMPORTANT**

**Les sorties de trains d'impulsions ne peuvent pas être utilisées par d'autres instructions dans le programme utilisateur**

Lorsque vous configurez les sorties de la CPU ou du Signal Board en tant que générateurs d'impulsions (pour les instructions PWM ou de commande de mouvement), les adresses des sorties correspondantes (Q0.0, Q0.1, Q4.0 et Q4.1) sont supprimées de la mémoire Q et ne peuvent pas être utilisées à d'autres fins dans le programme utilisateur. Si votre programme utilisateur écrit une valeur dans une sortie utilisée comme générateur d'impulsions, la CPU n'écrit pas cette valeur dans la sortie physique.

L'adresse par défaut est QW1000 pour PWM1 et QW1002 pour PWM2. La valeur à cette adresse commande la durée de l'impulsion et est initialisée à la valeur "Durée d'impulsion initiale" indiquée ci-avant à chaque fois que la CPU passe de l'état ARRET à l'état MARCHE. Vous modifiez cette valeur de mot Q pendant l'exécution pour modifier la largeur d'impulsion.

## 8.7 Consignation de données

Votre programme de commande peut utiliser les instructions Data log pour stocker des valeurs de données d'exécution dans des fichiers journaux permanents. Les fichiers journaux sont stockés en mémoire flash (CPU ou carte mémoire). Les données des journaux sont stockées en format CSV standard (valeurs séparées par une virgule). Les enregistrements sont organisés sous forme de fichier journal circulaire d'une dimension prédéterminée.

Vous utiliserez les instructions Data log dans votre programme pour créer, ouvrir, écrire un enregistrement et fermer les fichiers journaux. Vous décidez des valeurs du programme qui seront consignées en créant une mémoire tampon qui définit un enregistrement de journal unique. Votre mémoire tampon de données sert de stockage temporaire pour un nouvel enregistrement de journal. Les nouvelles valeurs en cours doivent être copiées par programme dans la mémoire tampon pendant l'exécution. Une fois toutes les valeurs de données en cours actualisées, vous pouvez exécuter l'instruction DataLogWrite pour transférer les données de la mémoire tampon dans un enregistrement de journal.

Utilisez le serveur Web API intégré pour gérer vos fichiers journaux. Téléchargez des enregistrements récents ou tous les enregistrements, effacez des enregistrements ou supprimez des fichiers journaux à l'aide de la page Web standard pour les journaux de données. Lorsqu'un fichier journal a été transféré dans votre PC, vous pouvez en analyser les données à l'aide de tableurs standard tels que Excel.

### 8.7.1 Structure des enregistrements de journaux

Les paramètres DATA et HEADER de l'instruction DataLogCreate définissent le type de données et la description d'en-tête de colonne de tous les éléments de données dans un enregistrement de journal.

#### Paramètre DATA pour l'instruction DataLogCreate

Le paramètre DATA désigne de la mémoire utilisée comme tampon temporaire pour un nouvel enregistrement de journal et doit correspondre à une adresse M ou DB.

Vous pouvez affecter un DB entier (dérivé d'un type de données API que vous affectez à la création du DB) ou une partie d'un DB (l'élément de DB indiqué peut être tout type de données, structure de données, type de données API ou tableau de données).

Les structures sont limitées à un niveau d'imbrication unique. Le nombre total d'éléments de données déclarés doit correspondre au nombre de colonnes indiqué dans le paramètre HEADER. Le nombre maximum d'éléments de données que vous pouvez affecter est 253 (avec horodatage) ou 255 (sans horodatage). Cette limitation maintient votre enregistrement en deçà de la limite de colonne de 256 d'une feuille Excel.

Le paramètre DATA peut indiquer des éléments de données rémanents ou non rémanents dans un DB de type "standard" (compatible avec S7-300/400) ou "optimisé".

Pour écrire un enregistrement de journal DATA, vous devez d'abord charger de nouvelles valeurs du processus dans l'enregistrement DATA temporaire, puis exécuter l'instruction DataLogWrite qui enregistre les nouvelles valeurs de l'enregistrement dans le fichier journal.

#### Paramètre HEADER pour l'instruction DataLogCreate

Le paramètre HEADER désigne les noms d'en-tête de colonne dans la ligne supérieure de la matrice de données codée dans le fichier CSV. Les données HEADER doivent se situer en mémoire DB ou M et les caractères doivent respecter les règles du format CSV standard avec des virgules séparant chaque nom de colonne. Les types de données possibles sont la chaîne, le tableau d'octets ou le tableau de caractères. Les tableaux de caractères ou d'octets permettent d'avoir une taille plus importante, les chaînes étant limitées à 255 octets au maximum. Le paramètre HEADER est facultatif. Si HEADER n'est pas défini, aucune ligne d'en-tête n'est créée dans le fichier journal.



## 8.7.2 Instructions de gestion des journaux de données

### 8.7.2.1 DataLogCreate

Tableau 8- 123 Instruction DataLogCreate

CONT/LOG	SCL	Description
	<pre>"DataLogCreate_DB" (     req:=_bool_in_,     records:=_udint_in_,     format:=_uint_in_,     timestamp:=_uint_in_,     done=&gt;_bool_out_,     busy=&gt;_bool_out_,     error=&gt;_bool_out_,     status=&gt;_word_out_,     name:=_string_inout_,     ID:=_dword_inout_,     header:=_variant_inout_,     data:=_variant_inout_);</pre>	<p>Crée et initialise un fichier journal. Le fichier est créé dans le répertoire \DataLogs de l'API sous le nom indiqué par le paramètre NAME et les opérations d'écriture y sont implicitement autorisées. Vous pouvez utiliser les instructions Data log pour stocker par programme les données d'exécution du processus dans la mémoire flash de la CPU.</p> <p>STEP 7 crée automatiquement le DB d'instance associé lorsque vous insérez l'instruction.</p>

<sup>1</sup> Dans l'exemple SCL, "DataLogCreate\_DB" est le nom du DB d'instance.

Tableau 8- 124 Types de données pour les paramètres

Paramètre et type		Type de données	Description
REQ	IN	Bool	Un front montant déclenche l'opération. (valeur par défaut : faux)
RECORDS	IN	UDint	Nombre maximum d'enregistrements que le journal circulaire peut contenir avant que l'entrée la plus ancienne ne soit écrasée. L'enregistrement d'en-tête n'est pas inclus. L'API doit disposer d'une mémoire de chargement suffisante pour que le journal soit créé avec succès. (valeur par défaut - 1)
FORMAT	IN	UInt	Format du journal de données : <ul style="list-style-type: none"> <li>0 : Format interne (non pris en charge)</li> <li>1 : Valeurs séparées par des virgules "csv-eng" (valeur par défaut)</li> </ul>
TIMESTAMP	IN	UInt	Format d'horodatage des données. Des en-têtes de colonne pour les champs date et heure ne sont pas nécessaires. L'horodatage utilise l'heure système (UTC : temps universel coordonné) et non l'heure locale. <ul style="list-style-type: none"> <li>0 : Pas d'horodatage</li> <li>1 : Horodatage (valeur par défaut)</li> </ul>

8.7 Consignation de données

Paramètre et type		Type de données	Description
NAME	IN	Variant	Nom du journal de données. Vous indiquez ce nom. Cette variante accepte seulement le type de données String et peut uniquement se situer en mémoire locale, DB ou M. (valeur par défaut : '') Cette référence chaîne est également utilisée comme nom du fichier journal. Vous devez respecter les restrictions concernant les noms dans le système de fichiers Windows. Ainsi, les caractères \ / : * ? " < >   et l'espace ne sont pas autorisés.
ID	IN_OUT	DWord	Identificateur numérique du journal de données. Vous mémorisez cette valeur générée dont vous aurez besoin avec les autres instructions de journaux de données. Le paramètre ID est uniquement utilisé comme sortie dans l'instruction DataLogCreate. (valeur par défaut : 0) L'accès symbolique à ce paramètre n'est pas autorisé.
HEADER	IN_OUT	Variant	Pointeur désignant les noms d'en-tête de colonne du journal dans la ligne supérieure de la matrice de données codée dans le fichier CSV. (Valeur par défaut : nul). Les données HEADER doivent se situer en mémoire DB ou M. Les caractères doivent respecter les règles du format CSV standard avec des virgules séparant chaque nom de colonne. Les types de données possibles sont la chaîne, le tableau d'octets ou le tableau de caractères. Les tableaux de caractères ou d'octets permettent d'avoir une taille plus importante, les chaînes étant limitées à 255 octets au maximum. Le paramètre HEADER est facultatif. Si HEADER n'est pas paramétré, aucune ligne d'en-tête n'est créée dans le fichier journal.
DATA	IN_OUT	Variant	Pointeur désignant une structure, un type de données utilisateur (UDT) ou un tableau pour les données des enregistrements. Les données des enregistrements doivent se situer en mémoire DB ou M. Le paramètre DATA indique les éléments de données individuels (colonnes) d'un enregistrement de journal et leur type de données. Les structures sont limitées à un niveau d'imbrication unique. Le nombre d'éléments de données déclarés doit correspondre au nombre de colonnes indiqué dans le paramètre HEADER. Le nombre maximum d'éléments de données que vous pouvez affecter est 253 (avec horodatage) ou 255 (sans horodatage). Cette limitation maintient votre enregistrement en deçà de la limite de colonne de 256 d'une feuille Excel.
DONE	OUT	Bool	Le bit DONE est VRAI pour un cycle lorsque la dernière demande s'est achevée sans erreur. (valeur par défaut : faux)
BUSY	OUT	Bool	<ul style="list-style-type: none"> <li>• 0 : Pas d'opération en cours</li> <li>• 1 : Opération en cours</li> </ul>
ERROR	OUT	Bool	Le bit ERROR est VRAI pour un cycle lorsque la dernière demande s'est achevée avec une erreur. La valeur de code d'erreur dans le paramètre STATUS ne vaut que pendant le cycle où ERROR est à VRAI.
STATUS	OUT	Word	Code d'erreur d'exécution (valeur par défaut : 0)

Le fichier journal est créé avec une taille fixe prédéterminée basée sur les paramètres RECORDS et DATA. Les enregistrements sont organisés sous forme de fichier journal circulaire. De nouveaux enregistrements sont ajoutés au fichier journal jusqu'à ce que le nombre maximum d'enregistrements indiqué dans le paramètre RECORDS soit sauvegardé. L'enregistrement suivant se substituera alors à l'enregistrement le plus ancien. Une nouvelle opération d'écriture écrasera l'enregistrement le plus ancien suivant et ainsi de suite.

---

**Remarque**

Si vous voulez éviter l'écrasement d'enregistrements, vous pouvez créer un nouveau journal sur la base du journal en cours avec l'instruction DataLogNewFile lorsque le journal en cours a atteint le nombre maximum d'enregistrements. Les nouveaux enregistrements seront alors sauvegardés dans le nouveau fichier journal. L'ancien fichier journal et ses données enregistrées sont conservés dans la mémoire flash.

---

Utilisation de la mémoire :

- Les journaux ne consomment que de la mémoire de chargement.
- Aucune limite n'est définie pour le nombre total de journaux. La taille de tous les journaux combinés est limitée par les ressources de mémoire de chargement disponibles. Huit journaux au maximum peuvent être ouverts simultanément.
- Le nombre maximum possible pour le paramètre RECORDS est la limite des nombres UDint (4 294 967 295). La limite effective pour le paramètre RECORD dépend de la taille d'un enregistrement individuel, de la taille des autres journaux et des ressources de mémoire de chargement disponibles. En outre, Excel limite le nombre de lignes autorisées dans une feuille Excel.

---

**Remarque**

L'exécution d'une instruction DataLogCreate s'étend sur de nombreux cycles du programme. La durée réelle nécessaire à la création d'un fichier journal dépend de la structure d'un enregistrement et du nombre d'enregistrements. La logique de votre programme doit surveiller et capturer la transition à l'état VRAI du bit DONE de DataLogCreate avant que le nouveau journal ne puisse être utilisé pour d'autres opérations sur les journaux.

---

Tableau 8- 125 Valeurs de ERROR et STATUS

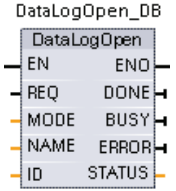
ERROR	STATUS (W#16#...)	Description
0	0000	Pas d'erreur
0	7000	Appel sans front de REQ : BUSY = 0, DONE = 0
0	7001	Premier appel avec front de REQ (fonctionne) : BUSY = 1, DONE = 0
0	7002	n-ième appel (fonctionne) : BUSY = 1, DONE = 0
1	8070	Toute la mémoire d'instance interne est utilisée.
1	807F	Erreur interne
1	8090	Nom de fichier invalide
1	8091	Le paramètre NAME ne fait pas référence à une chaîne de caractères.
1	8093	Le journal de données existe déjà.

8.7 Consignation de données

ERROR	STATUS (W#16#...)	Description
1	8097	La longueur de fichier demandée dépasse le maximum du système de fichiers.
1	80B3	Mémoire de chargement disponible insuffisante
1	80B4	Cartouche mémoire protégée en écriture
1	80C1	Trop de fichiers ouverts : huit fichiers journaux ouverts sont autorisés au maximum.
1	8253	Nombre d'enregistrements invalide
1	8353	Format sélectionné invalide
1	8453	Horodatage sélectionné invalide
1	8B24	Allocation de zone HEADER invalide : Pointe par exemple sur la mémoire locale
1	8B51	Type de données du paramètre HEADER invalide
1	8B52	Trop d'éléments de données dans le paramètre HEADER
1	8C24	Affectation de zone DATA invalide : Pointe par exemple sur la mémoire locale
1	8C51	Type de données du paramètre DATA invalide
1	8C52	Trop d'éléments de données dans le paramètre DATA

8.7.2.2 DataLogOpen

Tableau 8- 126 Instruction DataLogOpen

CONT/LOG	SCL	Description
	<pre>"DataLogOpen_DB" (     req:= _bool_in_,     mode:= _uint_in_,     done=&gt; _bool_out_,     busy=&gt; _bool_out_,     error=&gt; _bool_out_,     status=&gt; _word_out_,     name:= _string_inout_,     ID:= _dword_inout_ );</pre>	<p>Ouvre un fichier journal pré-existant. Un journal doit être ouvert pour que vous puissiez y écrire de nouveaux enregistrements. Les journaux peuvent être ouverts et fermés de façon individuelle. Huit journaux au maximum peuvent être ouverts en même temps.</p> <p>STEP 7 crée automatiquement le DB d'instance associé lorsque vous insérez l'instruction.</p>

<sup>2</sup> Dans l'exemple SCL, "DataLogOpen\_DB" est le nom du DB d'instance.

Tableau 8- 127 Types de données pour les paramètres

Paramètre et type	Type de données	Description
REQ	IN	Bool Un front montant déclenche l'opération. (valeur par défaut : faux)
MODE	IN	UInt Mode de fonctionnement : <ul style="list-style-type: none"> <li>0 : Ajouter aux données existantes (valeur par défaut)</li> <li>1 : Effacer tous les enregistrements existants</li> </ul>
NAME	IN	Variant Nom d'un journal existant. Cette variante accepte seulement le type de données String et peut uniquement se situer en mémoire locale, DB ou M. (valeur par défaut : '')

Paramètre et type		Type de données	Description
ID	IN_OUT	DWord	Identificateur numérique d'un journal de données (valeur par défaut : 0) <b>Remarque</b> : L'accès symbolique à ce paramètre n'est pas autorisé.
DONE	OUT	Bool	Le bit DONE est VRAI pour un cycle lorsque la dernière demande s'est achevée sans erreur. (valeur par défaut : faux)
BUSY	OUT	Bool	<ul style="list-style-type: none"> <li>• 0 : Pas d'opération en cours</li> <li>• 1 : Opération en cours</li> </ul>
ERROR	OUT	Bool	Le bit ERROR est VRAI pour un cycle lorsque la dernière demande s'est achevée avec une erreur. La valeur de code d'erreur dans le paramètre STATUS ne vaut que pendant le cycle où ERROR est à VRAI.
STATUS	OUT	Word	Code d'erreur d'exécution (valeur par défaut : 0)

Vous pouvez indiquer soit le nom NAME, soit l'identificateur numérique ID (ID en tant que paramètre d'entrée) d'un journal pré-existant. Si vous indiquez les deux paramètres et que l'ID valide indiqué correspond bien au journal désigné par NAME, l'ID est utilisé et le nom NAME n'est pas pris en compte.

Le nom indiqué dans NAME doit être le nom d'un journal créé avec l'instruction DataLogCreate. Si vous indiquez uniquement le paramètre NAME et que NAME désigne un journal valide, l'ID correspondant est renvoyé (ID en tant que paramètre de sortie).

#### Remarque

##### Utilisation générale des fichiers journaux

- Les fichiers journaux sont automatiquement ouverts après les instructions DataLogCreate et DataLogNewFile.
- Les fichiers journaux sont automatiquement fermés lors d'un passage de l'API de l'état MARCHE à l'état ARRET ou lors d'une mise hors tension puis sous tension de l'API.
- Un fichier journal doit être ouvert pour qu'une nouvelle opération DataLogWrite soit possible.
- Huit fichiers journaux au plus peuvent être ouverts à un moment donné. Il peut exister plus de huit fichiers journaux, mais certains d'entre eux doivent être fermés pour qu'il n'y ait pas plus de huit ouverts.

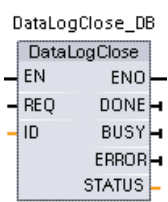
Tableau 8- 128 Valeurs de ERROR et STATUS

ERROR	STATUS (W#16#)	Description
0	0000	Pas d'erreur
0	0002	Avertissement : Fichier journal déjà ouvert par ce programme d'application
0	7000	Appel sans front de REQ : BUSY = 0, DONE = 0
0	7001	Premier appel avec front de REQ (fonctionne) : BUSY = 1, DONE = 0
0	7002	n-ième appel (fonctionne) : BUSY = 1, DONE = 0
1	8070	Toute la mémoire d'instance interne est utilisée.
1	8090	La définition de journal est incompatible avec le fichier journal existant.
1	8091	Le paramètre NAME ne fait pas référence à une chaîne de caractères.

ERROR	STATUS (W#16#)	Description
1	8092	Le journal n'existe pas.
1	80C0	Le fichier journal est verrouillé.
1	80C1	Trop de fichiers ouverts : huit fichiers journaux ouverts sont autorisés au maximum.

### 8.7.2.3 DataLogClose

Tableau 8- 129 Instruction DataLogClose

CONT/LOG	SCL	Description
	<pre>"DataLogClose_DB" (     req:=_bool_in_,     done=&gt;_bool_out_,     busy=&gt;_bool_out_,     error=&gt;_bool_out_,     status=&gt;_word_out_,     ID:=_dword_inout_);</pre>	<p>Ferme un fichier journal ouvert. Les opérations DataLogWrite sur un fichier journal fermé provoquent une erreur. Aucune opération d'écriture dans ce journal n'est autorisée tant qu'une nouvelle opération DataLogOpen n'est pas exécutée.</p> <p>Un passage à l'état ARRET ferme tous les fichiers journaux ouverts.</p> <p>STEP 7 crée automatiquement le DB d'instance associé lorsque vous insérez l'instruction.</p>

<sup>2</sup> Dans l'exemple SCL, "DataLogClose\_DB" est le nom du DB d'instance.

Tableau 8- 130 Types de données pour les paramètres

Paramètre et type		Type de données	Description
REQ	IN	Bool	Un front montant déclenche l'opération. (valeur par défaut : faux)
ID	IN_OUT	DWord	Identificateur numérique d'un journal de données. Utilisé uniquement comme entrée pour l'instruction DataLogClose (valeur par défaut : 0) <b>Remarque :</b> L'accès symbolique à ce paramètre n'est pas autorisé.
DONE	OUT	Bool	Le bit DONE est VRAI pour un cycle lorsque la dernière demande s'est achevée sans erreur.
BUSY	OUT	Bool	<ul style="list-style-type: none"> <li>0 : Pas d'opération en cours</li> <li>1 : Opération en cours</li> </ul>
ERROR	OUT	Bool	Le bit ERROR est VRAI pour un cycle lorsque la dernière demande s'est achevée avec une erreur. La valeur de code d'erreur dans le paramètre STATUS ne vaut que pendant le cycle où ERROR est à VRAI.
STATUS	OUT	Word	Code d'erreur d'exécution (valeur par défaut : 0)

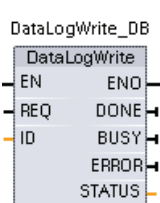
Tableau 8- 131 Valeurs de ERROR et STATUS

ERROR	STATUS (W#16#)	Description
0	0000	Pas d'erreur
0	0001	Le journal n'est pas ouvert.

ERROR	STATUS (W#16#)	Description
0	7000	Appel sans front de REQ : BUSY = 0, DONE = 0
0	7001	Premier appel avec front de REQ (fonctionne) : BUSY = 1, DONE = 0
0	7002	n-ième appel (fonctionne) : BUSY = 1, DONE = 0
1	8092	Le journal n'existe pas.

### 8.7.2.4 DataLogWrite

Tableau 8- 132 Instruction DataLogWrite

CONT/LOG	SCL	Description
	<pre>"DataLogWrite_DB" (     req:=_bool_in_,     done=&gt;_bool_out_,     busy=&gt;_bool_out_,     error=&gt;_bool_out_,     status=&gt;_word_out_,     ID:=_dword_inout_);</pre>	<p>Ecrit un enregistrement dans le journal spécifié. Le journal cible pré-existant doit être ouvert pour qu'une opération DataLogWrite soit autorisée.</p> <p>STEP 7 crée automatiquement le DB d'instance associé lorsque vous insérez l'instruction.</p>

<sup>2</sup> Dans l'exemple SCL, "DataLogWrite\_DB" est le nom du DB d'instance.

Tableau 8- 133 Types de données pour les paramètres

Paramètre et type		Type de données	Description
REQ	IN	Bool	Un front montant déclenche l'opération. (valeur par défaut : faux)
ID	IN_OUT	DWord	Identificateur numérique du journal de données. Utilisé uniquement comme entrée pour l'instruction DataLogWrite (valeur par défaut : 0) <b>Remarque :</b> L'accès symbolique à ce paramètre n'est pas autorisé.
DONE	OUT	Bool	Le bit DONE est VRAI pour un cycle lorsque la dernière demande s'est achevée sans erreur.
BUSY	OUT	Bool	<ul style="list-style-type: none"> <li>• 0 : Pas d'opération en cours</li> <li>• 1 : Opération en cours</li> </ul>
ERROR	OUT	Bool	Le bit ERROR est VRAI pour un cycle lorsque la dernière demande s'est achevée avec une erreur. La valeur de code d'erreur dans le paramètre STATUS ne vaut que pendant le cycle où ERROR est à VRAI.
STATUS	OUT	Word	Code d'erreur d'exécution (valeur par défaut : 0)

L'adresse et la structure de données de la mémoire tampon d'enregistrements sont définies par le paramètre DATA d'une instruction DataLogCreate. Vous devez charger par programme les valeurs du processus en cours d'exécution dans la mémoire tampon, puis exécuter l'instruction DataLogWrite pour copier les nouvelles données de la mémoire tampon dans le journal.

Le paramètre ID identifie un journal et une configuration d'enregistrement. Le nombre ID est généré à la création d'un journal.

8.7 Consignation de données

S'il y a des enregistrements vides dans le fichier journal circulaire, l'écriture se fait dans l'enregistrement vide disponible suivant. Si tous les enregistrements sont pleins, l'enregistrement le plus ancien est écrasé.

**PRUDENCE**

**Perte potentielle de données de journaux pendant une coupure de courant affectant la CPU**

Si une coupure de courant se produit pendant une opération DataLogWrite qui n'est pas encore achevée, l'enregistrement en cours de transfert dans le journal peut être perdu.

Tableau 8- 134 Valeurs de ERROR et STATUS

ERROR	STATUS (W#16#)	Description
0	0000	Pas d'erreur
0	0001	Indique que le journal est plein. Chaque journal est créé avec un nombre maximum défini d'enregistrements. Le dernier enregistrement du nombre maximum a été écrit. La prochaine opération d'écriture écrasera donc l'enregistrement le plus ancien.
0	7000	Appel sans front de REQ : BUSY = 0, DONE = 0
0	7001	Premier appel avec front de REQ (fonctionne) : BUSY = 1, DONE = 0
0	7002	n-ième appel (fonctionne) : BUSY = 1, DONE = 0
1	8070	Toute la mémoire d'instance interne est utilisée.
1	8092	Le journal n'existe pas.
1	80B0	Le fichier journal n'est pas ouvert (uniquement pour le mode d'ouverture explicite).

8.7.2.5 DataLogNewFile

Tableau 8- 135 Instruction DataLogNewFile

CONT/LOG	SCL	Description
	<pre>"DataLogNewFile_DB" (   req:= _bool_in_,   records:= _udint_in_,   done=&gt; _bool_out_,   busy=&gt; _bool_out_,   error=&gt; _bool_out_,   status=&gt; _word_out_,   name:= _DataLog_out_,   ID:= dword inout );</pre>	<p>Permet à votre programme de créer un nouveau fichier journal sur la base d'un fichier journal existant.</p> <p>STEP 7 crée automatiquement le DB d'instance associé lorsque vous insérez l'instruction.</p>

<sup>2</sup> Dans l'exemple SCL, "DataLogNewFile\_DB" est le nom du DB d'instance.



Tableau 8- 136 Types de données pour les paramètres

Paramètre et type		Type de données	Description
REQ	IN	Bool	Un front montant déclenche l'opération. (valeur par défaut : faux)
RECORDS	IN	UDInt	Nombre maximum d'enregistrements que le journal circulaire peut contenir avant que l'entrée la plus ancienne ne soit écrasée. (valeur par défaut : 1) 1) L'enregistrement d'en-tête n'est pas inclus. La CPU doit disposer d'une mémoire de chargement suffisante pour que le journal soit créé avec succès.
NAME	IN	Variant	Nom du journal de données. Vous indiquez ce nom. Cette variante accepte seulement le type de données String et peut uniquement se situer en mémoire locale, DB ou M. (valeur par défaut : ' ') ' Cette référence chaîne est également utilisée comme nom du fichier journal. Vous devez respecter les restrictions concernant les noms dans le système de fichiers Windows. Ainsi, les caractères \ / : * ? " < >   et l'espace ne sont pas autorisés.
ID	IN_OUT	DWord	Identificateur numérique du journal de données (valeur par défaut : 0): <ul style="list-style-type: none"> <li>• Au moment de l'exécution, l'entrée ID identifie un journal de données valide. La configuration du nouveau journal est reprise de ce journal.</li> <li>• Après l'exécution, le paramètre ID devient une sortie qui renvoie l'ID du fichier journal nouvellement créé.</li> </ul> <b>Remarque :</b> L'accès symbolique à ce paramètre n'est pas autorisé.
DONE	OUT	Bool	Le bit DONE est VRAI pour un cycle lorsque la dernière demande s'est achevée sans erreur.
BUSY	OUT	Bool	<ul style="list-style-type: none"> <li>• 0 : Pas d'opération en cours</li> <li>• 1 : Opération en cours</li> </ul>
ERROR	OUT	Bool	Le bit ERROR est VRAI pour un cycle lorsque la dernière demande s'est achevée avec une erreur. La valeur de code d'erreur dans le paramètre STATUS ne vaut que pendant le cycle où ERROR est à VRAI.
STATUS	OUT	Word	Code d'erreur d'exécution (valeur par défaut : 0) 0)

Vous pouvez exécuter l'instruction DataLogNewFile lorsqu'un journal est rempli ou est considéré comme achevé et que vous ne voulez perdre aucune des données qui y sont contenues. Un nouveau fichier journal vide ayant la structure du fichier journal plein peut être créé. L'enregistrement d'en-tête est repris dans le journal d'origine, ainsi que les propriétés du journal d'origine (mémoire tampon d'enregistrements DATA, format des données et horodatage). Le fichier journal d'origine est implicitement fermé et le nouveau fichier journal est implicitement ouvert.

Déclenchement via les paramètres de DataLogWrite : Votre programme doit surveiller les paramètres ERROR et STATUS de chaque opération DataLogWrite. Lorsque le dernier enregistrement est écrit et qu'un journal est plein, le bit ERROR de DataLogWrite est égal à 1 et le mot DataLogWrite STATUS est égal à 1. Ces valeurs de ERROR et de STATUS ne sont valables que pendant un cycle, aussi votre logique de surveillance doit-elle utiliser ERROR = 1 comme déclencheur pour capturer la valeur de STATUS puis tester que STATUS est égal à 1 (journal plein).

Fonctionnement de DataLogNewFile : Lorsque la logique de votre programme reçoit le signal "journal plein", elle active une opération DataLogNewFile. Vous devez exécuter DataLogNewFile avec l'ID d'un journal existant et ouvert (déjà plein) mais avec un nouveau paramètre NAME unique. Une fois l'opération DataLogNewFile exécutée, une nouvelle valeur ID correspondant au nom du nouveau journal est renvoyée (en tant que paramètre de sortie). Le nouveau fichier journal est implicitement ouvert et est prêt à recevoir de nouveaux enregistrements. Les nouvelles opérations DataLogWrite destinées au nouveau fichier journal doivent utiliser la valeur ID renvoyée par l'opération DataLogNewFile.

**Remarque**

L'exécution d'une instruction DataLogNewFile s'étend sur de nombreux cycles du programme. La durée réelle nécessaire à la création d'un fichier journal dépend de la structure d'un enregistrement et du nombre d'enregistrements. La logique de votre programme doit surveiller et capturer la transition à l'état VRAI du bit DONE de DataLogNewFile avant que le nouveau journal ne puisse être utilisé pour d'autres opérations sur les journaux.

Tableau 8- 137 Valeurs de ERROR et STATUS

ERROR	STATUS (W#16#)	Description
0	0000	Pas d'erreur
0	7000	Appel sans front de REQ : BUSY = 0, DONE = 0
0	7001	Premier appel avec front de REQ (fonctionne) : BUSY = 1, DONE = 0
0	7002	n-ième appel (fonctionne) : BUSY = 1, DONE = 0
1	8070	Toute la mémoire d'instance interne est utilisée.
1	8090	Nom de fichier invalide
1	8091	Le paramètre NAME ne fait pas référence à une chaîne de caractères.
1	8092	Le journal n'existe pas.
1	8093	Le journal de données existe déjà.
1	8097	La longueur de fichier demandée dépasse le maximum du système de fichiers.
1	80B3	Mémoire de chargement disponible insuffisante
1	80B4	La cartouche mémoire est protégée en écriture.
1	80C1	Trop de fichiers ouverts

**8.7.3 Utilisation des journaux de données**

Les fichiers journaux sont stockés en format "valeurs séparées par des virgules" (\*.csv) en mémoire flash permanente. Vous pouvez visualiser les journaux de données à l'aide du serveur Web de l'API ou en retirant la carte mémoire de l'API et en l'insérant dans un lecteur de carte PC standard.

### Visualisation de journaux de données avec le serveur Web API

Si le port PROFINET de l'API et un PC sont connectés à un réseau, vous pouvez utiliser un navigateur Web PC tel que Microsoft Internet Explorer ou Mozilla Firefox pour accéder au serveur Web API intégré. L'API peut être à l'état MARCHE ou ARRET lorsque vous utilisez le serveur Web API. Si l'API est à l'état MARCHE, votre programme de commande continue à s'exécuter pendant que le serveur Web API transfère les données de journaux via le réseau.

Accès par le serveur Web :

1. Activez le serveur Web dans la configuration d'appareil pour la CPU cible (Page 534).
2. Connectez votre PC à l'API via le réseau PROFINET (Page 535).
3. Ouvrez une session sur le serveur Web intégré (Page 537).
4. Téléchargez les enregistrements récents ou tous les enregistrements, effacez des enregistrements ou supprimez des fichiers journaux à l'aide de la page Web standard pour les journaux de données (Page 548).
5. Une fois qu'une copie d'un fichier journal a été chargée dans votre PC, vous pouvez ouvrir le fichier .csv avec un tableur tel que Excel.

### Visualisation de journaux de données sur une carte mémoire API

Si une carte mémoire S7-1200 de type "programme" est insérée dans la CPU S7-1200, vous pouvez enlever la carte mémoire et l'insérer dans une fente de carte SD (Secure Digital) ou MMC (MultiMediaCard) standard sur un PC ou une PG. L'API est à l'état ARRET lorsque la carte mémoire est retirée et le programme de commande ne s'exécute pas.

Servez-vous de l'explorateur de fichiers Windows pour naviguer jusqu'au répertoire \DataLog sur la carte mémoire. Tous vos fichiers journaux \\*.csv se trouvent dans ce répertoire.

Faites une copie des fichiers journaux que vous placerez sur un lecteur local de votre PC. Puis, utilisez Excel pour ouvrir une copie locale d'un fichier \*.csv et non le fichier original stocké sur la carte mémoire.

<b>PRUDENCE</b>
-----------------

<b>Vous pouvez copier des fichiers journaux sur une carte mémoire S7-1200 à l'aide d'un lecteur PC, mais ne modifiez pas ni ne supprimez ces fichiers.</b>
------------------------------------------------------------------------------------------------------------------------------------------------------------

La page standard du serveur Web pour les journaux de données est l'outil recommandé pour la visualisation, le téléchargement (copie), l'effacement (des données) et la suppression des fichiers journaux. Le serveur Web gère les fichiers en carte mémoire pour vous et permet d'éviter les modifications ou suppressions accidentelles de données.
------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

La navigation directe dans le système de fichiers de la carte mémoire via l'explorateur Windows présente un risque car vous pouvez supprimer ou modifier accidentellement les journaux de données ou d'autres fichiers système, ce qui pourrait altérer des fichiers ou rendre la carte mémoire inutilisable.
---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

### Visualisation de journaux dans un navigateur Web

Même si vous n'utilisez pas le serveur Web, vous pouvez visualiser les journaux directement dans un navigateur Web, tel qu'Internet Explorer ou Mozilla Firefox. Il suffit d'entrer le texte suivant dans la barre d'adresse de votre navigateur avec l'adresse IP de votre CPU et le nom réel du fichier journal que vous avez indiqué dans STEP 7 au lieu de "MyDataLog" :

`http://192.168.0.1/DataLog.html?FileName=MyDataLog.csv`

Les adresses fixes des fichiers journaux permettent également d'accéder à ces derniers par le biais d'outils de collecte de fichiers de tiers.

#### 8.7.4 Limites de taille des fichiers journaux

Les fichiers journaux partagent la mémoire de chargement de l'API avec le programme, les données de programme, les données de configuration, les pages Web personnalisées et les données système API. Un grand programme utilisant la mémoire de chargement interne requiert une part importante de la mémoire de chargement, les fichiers journaux pouvant alors manquer d'espace. Dans ce cas, vous pouvez utiliser une carte programme pour augmenter la taille de la mémoire de chargement. Les CPU S7-1200 peuvent utiliser soit la mémoire de chargement interne, soit la mémoire de chargement externe, mais pas les deux simultanément.

Reportez-vous au chapitre sur les cartes mémoire pour plus de détails sur la création d'une carte programme (Page 120).

#### Règle de la taille maximale pour un fichier journal

La taille maximale d'un fichier journal ne doit pas dépasser 25% de la taille de la mémoire de chargement (interne ou externe). Si votre application a besoin de plus d'entrées de journal, utilisez l'instruction "DataLogNewFile" pour créer un nouveau fichier une fois tous les enregistrements du premier fichier remplis. Le tableau suivant présente les tailles maximales pour un fichier journal.

Tableau 8- 138 Taille de la mémoire de chargement et taille maximale pour un fichier journal

Zone de données	CPU 1211C	CPU 1212C	CPU 1214C	CPU 1215C	Stockage des données
<b>Mémoire de chargement interne</b> mémoire flash	1 Mo (250 Ko max. pour un fichier journal)	1 Mo (250 Ko max. pour un fichier journal)	4 Mo (500 Ko max. pour un fichier journal)	4 Mo (500 Ko max. pour un fichier journal)	Programme utilisateur et données de programme, données de configuration, journaux (Data logs), pages Web personnalisées plus données système API
<b>Mémoire de chargement externe</b> cartes mémoire flash "programme" optionnelles	2 Mo, 12 Mo ou 24 Mo selon la taille de la carte SD (500 Ko max. pour un fichier journal avec une carte 2 Mo) (6 Mo max. pour un fichier journal avec une carte 24 Mo)				

## Détermination de la taille de l'espace libre en mémoire de chargement

1. Etablissez une liaison en ligne entre STEP 7 et l'API S7-1200 cible.
2. Chargez le programme auquel vous voulez ajouter des opérations liées aux journaux de données.
3. Créez les pages Web personnalisées optionnelles dont vous avez besoin (les pages Web standard qui donnent accès aux journaux de données sont stockées dans le firmware de l'API et n'utilisent pas la mémoire de chargement).
4. Utilisez les outils en ligne et de diagnostic pour connaître la taille de la mémoire de chargement et le pourcentage d'espace libre en mémoire de chargement (Page 728).
5. Multipliez la taille de la mémoire de chargement par le pourcentage d'espace libre pour connaître l'espace actuellement libre en mémoire de chargement.

## Règle de la taille maximale pour l'ensemble des journaux de données (data logs)

La quantité d'espace libre en mémoire de chargement varie pendant le fonctionnement normal au fur et à mesure que le système d'exploitation utilise et libère de la mémoire. Nous vous conseillons de limiter la taille combinée de tous les fichiers journaux à la moitié de l'espace libre disponible.

## Calcul des besoins de mémoire pour un enregistrement de journal unique

Les données des journaux sont stockées sous forme d'octets caractères dans le format de fichier CSV (valeurs séparées par des virgules). Le tableau suivant montre le nombre d'octets nécessaires pour le stockage de chaque type de données.

Tableau 8- 139            Tailles de données CSV

Type de données	Nombre d'octets (octets de données plus octet de virgule de séparation)
Bool	2
Byte	5
Word	7
DWord	12
Char	4
String	<b>Exemple 1 :</b> MaChaîne[10] La taille de chaîne maximale est définie à 10 caractères. caractères de texte + remplissage automatique avec des espaces = 10 octets guillemets ouvrant et fermant + virgule = 3 octets 10 + 3 = 13 octets au total <b>Exemple 2 :</b> MaChaîne2 Si aucune taille n'est indiquée entre crochets, 254 octets est pris par défaut. caractères de texte + remplissage automatique avec des espaces = 254 octets guillemets ouvrant et fermant + virgule = 3 octets 254 + 3 = 257 octets au total
USInt	5
UInt	7

Type de données	Nombre d'octets (octets de données plus octet de virgule de séparation)
UDInt	12
SInt	5
Int	7
DInt	12
Real	16
LReal	25
Time	15
DTL	24

Le paramètre DATA de DataLogCreate désigne une structure qui indique le nombre de champs de données et le type de données de chaque champ pour un enregistrement de données. Le tableau ci-dessus indique les octets nécessaires dans le fichier CSV pour chaque type de données. Multipliez le nombre d'occurrences d'un type de données précis par le nombre d'octets qu'il nécessite. Répétez cette opération pour chaque type de données dans l'enregistrement et additionnez les nombres d'octets obtenus pour avoir la taille totale de l'enregistrement. Ajoutez un octet comme caractère de fin de ligne.

taille d'un enregistrement de journal = somme des octets nécessaires pour tous les champs de données + 1 (caractère de fin de ligne)

### Calcul des besoins de mémoire pour la totalité du fichier journal

Le paramètre RECORDS de l'instruction DataLogCreate définit le nombre maximum d'enregistrements dans un fichier journal. Lors de la création d'un fichier journal, la taille de mémoire maximale est allouée.

taille du fichier journal = (nombre d'octets dans un enregistrement) x (nombre d'enregistrements)

### 8.7.5 Exemple de programme pour les journaux de données

Cet exemple de programme pour les journaux de données ne montre pas toute la logique de programme nécessaire pour récupérer des valeurs échantillons d'un processus dynamique, mais présente les opérations clés des instructions de journaux de données. La structure et le nombre de fichiers journaux que vous utilisez dépendent des exigences de commande de votre processus.

---

#### Remarque

##### Utilisation générale des fichiers journaux

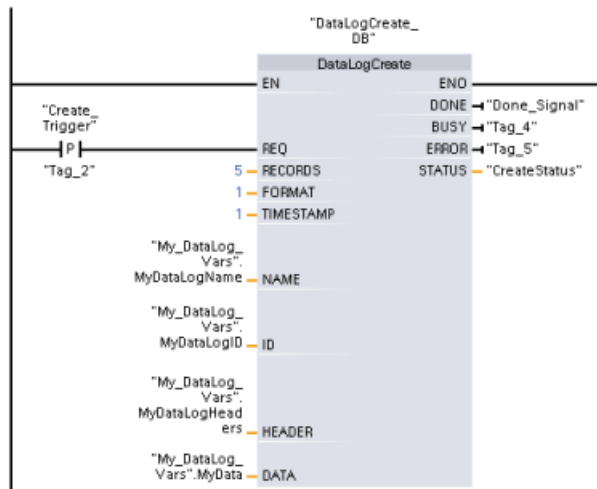
- Les fichiers journaux sont automatiquement ouverts après les instructions DataLogCreate et DataLogNew File.
  - Les fichiers journaux sont automatiquement fermés lors d'un passage de l'API de l'état MARCHE à l'état ARRET ou lors d'une mise hors tension puis sous tension de l'API.
  - Un fichier journal doit être ouvert pour qu'une opération DataLogWrite soit possible.
  - Huit fichiers journaux au plus peuvent être ouverts à un moment donné. Il peut exister plus de huit fichiers journaux, mais certains d'entre eux doivent être fermés pour qu'il n'y en ait pas plus de huit ouverts.
- 

### Exemple de programme de journal de données

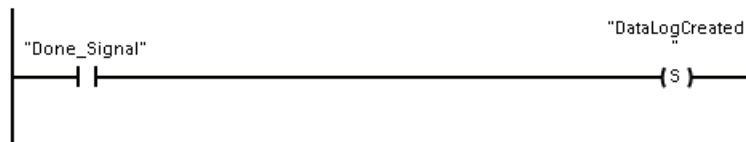
Des noms de journaux, un texte d'en-tête et la structure MyData sont créés à titre d'exemple dans un bloc de données. Les trois variables MyData stockent temporairement de nouvelles valeurs échantillons. On transfère les valeurs échantillons du processus situées à ces adresses de DB dans un fichier journal en exécutant l'instruction DataLogWrite.

My_Datalog_Vars			
	Nom	Type de données	Valeur initiale
1	Static		
2	MyNEWDatalogName	String	'MyNEWDatalog'
3	MyDataLogName	String	'MyDataLog'
4	MyDataLogID	DWord	0
5	MyDataLogHeaders	String	'Count, Temperature, Pressure'
6	MyData	Struct	
7	MyCount	Int	0
8	MyTemperature	Real	0.0
9	MyPressure	Real	0.0

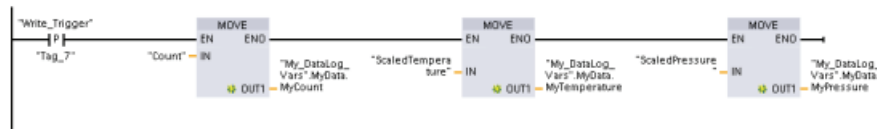
**Réseau 1** Un front montant de REQ démarre l'opération de création du journal.



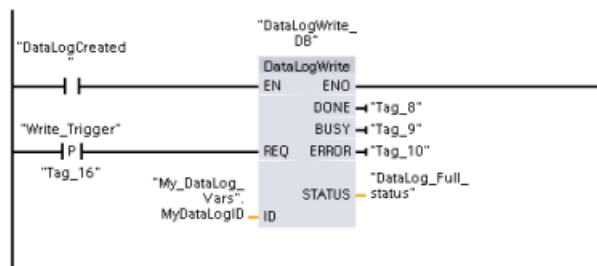
**Réseau 2** Capturer la sortie DONE de DataLogCreate car elle n'est valable que pendant un cycle.



**Réseau 3** Un front montant déclenche le stockage de nouvelles valeurs du processus dans la structure MyData.

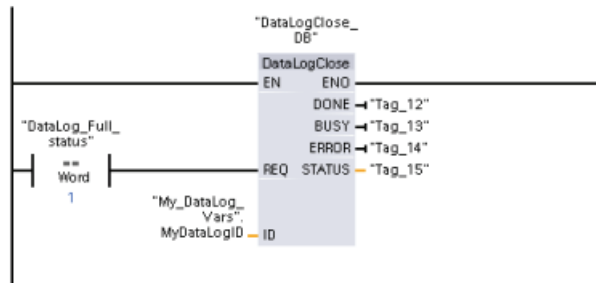


**Réseau 4** L'état de l'entrée EN indique lorsque l'opération DataLogCreate est achevée. Une opération de création s'étend sur plusieurs cycles et doit être achevée avant l'exécution d'une opération d'écriture. Un front montant à l'entrée REQ est l'événement qui déclenche une opération d'écriture validée.

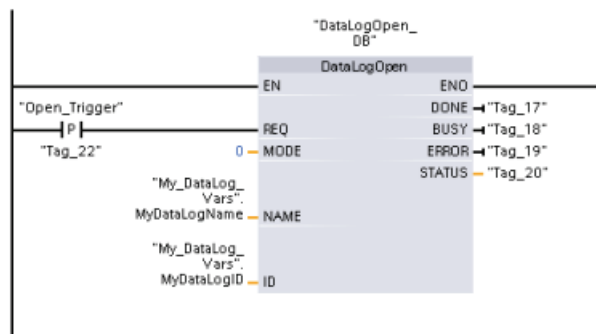




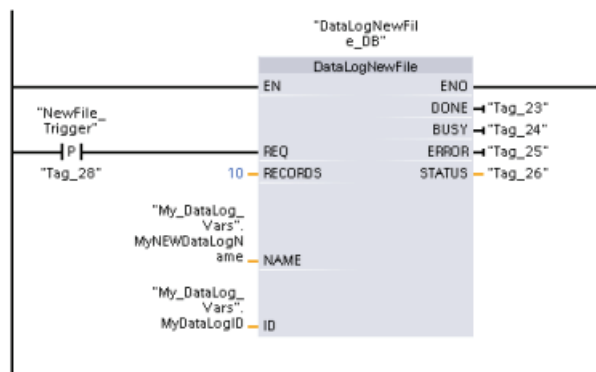
**Réseau 5** Fermer le journal de données une fois le dernier enregistrement écrit. Après l'exécution de l'opération DataLogWrite qui écrit le dernier enregistrement, l'état plein du fichier journal est signalé par la mise à 1 de la sortie STATUS de DataLogWrite.



**Réseau 6** Un front montant à l'entrée REQ de DataLogOpen simule l'enfoncement par un utilisateur d'un bouton sur une IHM qui ouvre un fichier journal. Si vous ouvrez un fichier journal dont tous les enregistrements sont pleins, l'opération DataLogWrite suivante écrasera l'enregistrement le plus ancien. Vous voudrez peut-être conserver l'ancien journal et, pour ce faire, créer un nouveau journal comme illustré au réseau 7.



**Réseau 7** Le paramètre ID est un paramètre d'entrée/sortie (IN/OUT). Vous fournissez d'abord la valeur ID du journal existant dont vous voulez copier la structure. Une fois l'opération DataLogNewFile achevée, une nouvelle valeur ID unique pour le nouveau journal est renvoyée dans l'adresse de référence ID. La capture nécessaire du bit DONE = VRAI n'est pas montrée (voir les réseaux 1, 2 et 4 pour un exemple de logique pour le bit DONE).



Fichiers journaux créés par l'exemple de programme visualisés via le serveur Web de la CPU S7-1200

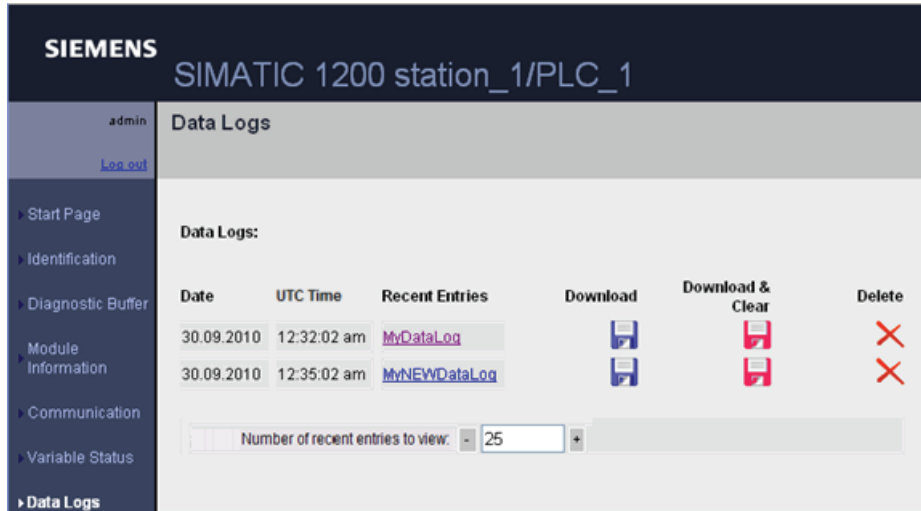


Tableau 8- 140 Exemples de fichiers .csv téléchargés visualisés dans Excel

<p>Deux enregistrements écrits dans un fichier de cinq enregistrements au maximum</p>	<table border="1"> <thead> <tr> <th></th> <th>A</th> <th>B</th> <th>C</th> <th>D</th> <th>E</th> <th>F</th> </tr> </thead> <tbody> <tr> <td>1</td> <td>Record</td> <td>Date</td> <td>UTC Time</td> <td>Count</td> <td>Temperature</td> <td>Pressure</td> </tr> <tr> <td>2</td> <td>1</td> <td>9/29/2010</td> <td>21:01:46</td> <td>5</td> <td>5.00E+00</td> <td>5.00E+00</td> </tr> <tr> <td>3</td> <td>2</td> <td>9/29/2010</td> <td>21:01:47</td> <td>5</td> <td>5.00E+00</td> <td>5.00E+00</td> </tr> <tr> <td>4</td> <td colspan="6">//END</td> </tr> <tr> <td>5</td> <td colspan="6"></td> </tr> </tbody> </table>		A	B	C	D	E	F	1	Record	Date	UTC Time	Count	Temperature	Pressure	2	1	9/29/2010	21:01:46	5	5.00E+00	5.00E+00	3	2	9/29/2010	21:01:47	5	5.00E+00	5.00E+00	4	//END						5																				
	A	B	C	D	E	F																																																			
1	Record	Date	UTC Time	Count	Temperature	Pressure																																																			
2	1	9/29/2010	21:01:46	5	5.00E+00	5.00E+00																																																			
3	2	9/29/2010	21:01:47	5	5.00E+00	5.00E+00																																																			
4	//END																																																								
5																																																									
<p>Cinq enregistrements dans un fichier journal de cinq enregistrements au maximum</p>	<table border="1"> <thead> <tr> <th></th> <th>A</th> <th>B</th> <th>C</th> <th>D</th> <th>E</th> <th>F</th> </tr> </thead> <tbody> <tr> <td>1</td> <td>Record</td> <td>Date</td> <td>UTC Time</td> <td>Count</td> <td>Temperature</td> <td>Pressure</td> </tr> <tr> <td>2</td> <td>1</td> <td>9/30/2010</td> <td>20:26:56</td> <td>1</td> <td>9.86E+01</td> <td>3.52E+01</td> </tr> <tr> <td>3</td> <td>2</td> <td>9/30/2010</td> <td>20:28:43</td> <td>2</td> <td>1.00E+02</td> <td>3.73E+01</td> </tr> <tr> <td>4</td> <td>3</td> <td>9/30/2010</td> <td>20:29:03</td> <td>3</td> <td>9.99E+01</td> <td>3.68E+01</td> </tr> <tr> <td>5</td> <td>4</td> <td>9/30/2010</td> <td>20:29:21</td> <td>4</td> <td>9.95E+01</td> <td>3.64E+01</td> </tr> <tr> <td>6</td> <td>5</td> <td>9/30/2010</td> <td>20:30:19</td> <td>5</td> <td>9.92E+01</td> <td>3.74E+01</td> </tr> <tr> <td>7</td> <td colspan="6"></td> </tr> </tbody> </table>		A	B	C	D	E	F	1	Record	Date	UTC Time	Count	Temperature	Pressure	2	1	9/30/2010	20:26:56	1	9.86E+01	3.52E+01	3	2	9/30/2010	20:28:43	2	1.00E+02	3.73E+01	4	3	9/30/2010	20:29:03	3	9.99E+01	3.68E+01	5	4	9/30/2010	20:29:21	4	9.95E+01	3.64E+01	6	5	9/30/2010	20:30:19	5	9.92E+01	3.74E+01	7						
	A	B	C	D	E	F																																																			
1	Record	Date	UTC Time	Count	Temperature	Pressure																																																			
2	1	9/30/2010	20:26:56	1	9.86E+01	3.52E+01																																																			
3	2	9/30/2010	20:28:43	2	1.00E+02	3.73E+01																																																			
4	3	9/30/2010	20:29:03	3	9.99E+01	3.68E+01																																																			
5	4	9/30/2010	20:29:21	4	9.95E+01	3.64E+01																																																			
6	5	9/30/2010	20:30:19	5	9.92E+01	3.74E+01																																																			
7																																																									
<p>Une sixième opération d'écriture pour l'écriture d'un enregistrement supplémentaire dans le fichier ci-dessus qui est plein provoque l'écrasement de l'enregistrement 1 le plus ancien par l'enregistrement 6. Une nouvelle opération d'écriture provoquera l'écrasement de l'enregistrement 2 par l'enregistrement 7, et ainsi de suite.</p>	<table border="1"> <thead> <tr> <th></th> <th>A</th> <th>B</th> <th>C</th> <th>D</th> <th>E</th> <th>F</th> </tr> </thead> <tbody> <tr> <td>1</td> <td>Record</td> <td>Date</td> <td>UTC Time</td> <td>Count</td> <td>Temperature</td> <td>Pressure</td> </tr> <tr> <td>2</td> <td>6</td> <td>9/30/2010</td> <td>20:32:03</td> <td>6</td> <td>9.86E+01</td> <td>3.58E+01</td> </tr> <tr> <td>3</td> <td>2</td> <td>9/30/2010</td> <td>20:28:43</td> <td>2</td> <td>1.00E+02</td> <td>3.73E+01</td> </tr> <tr> <td>4</td> <td>3</td> <td>9/30/2010</td> <td>20:29:03</td> <td>3</td> <td>9.99E+01</td> <td>3.68E+01</td> </tr> <tr> <td>5</td> <td>4</td> <td>9/30/2010</td> <td>20:29:21</td> <td>4</td> <td>9.95E+01</td> <td>3.64E+01</td> </tr> <tr> <td>6</td> <td>5</td> <td>9/30/2010</td> <td>20:30:19</td> <td>5</td> <td>9.92E+01</td> <td>3.74E+01</td> </tr> <tr> <td>7</td> <td colspan="6"></td> </tr> </tbody> </table>		A	B	C	D	E	F	1	Record	Date	UTC Time	Count	Temperature	Pressure	2	6	9/30/2010	20:32:03	6	9.86E+01	3.58E+01	3	2	9/30/2010	20:28:43	2	1.00E+02	3.73E+01	4	3	9/30/2010	20:29:03	3	9.99E+01	3.68E+01	5	4	9/30/2010	20:29:21	4	9.95E+01	3.64E+01	6	5	9/30/2010	20:30:19	5	9.92E+01	3.74E+01	7						
	A	B	C	D	E	F																																																			
1	Record	Date	UTC Time	Count	Temperature	Pressure																																																			
2	6	9/30/2010	20:32:03	6	9.86E+01	3.58E+01																																																			
3	2	9/30/2010	20:28:43	2	1.00E+02	3.73E+01																																																			
4	3	9/30/2010	20:29:03	3	9.99E+01	3.68E+01																																																			
5	4	9/30/2010	20:29:21	4	9.95E+01	3.64E+01																																																			
6	5	9/30/2010	20:30:19	5	9.92E+01	3.74E+01																																																			
7																																																									

## 8.8 Gestion des blocs de données

### 8.8.1 READ\_DBL, WRIT\_DBL (Lecture ou écriture dans un DB en mémoire de chargement)

Tableau 8- 141 Instructions READ\_DBL et WRIT\_DBL

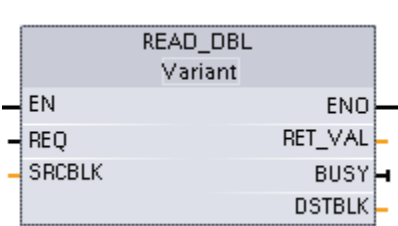
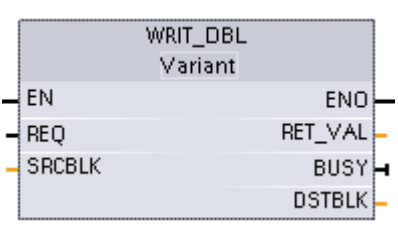
CONT/LOG		Description
	<pre>READ_DBL (   req:=_bool_in_,   srcblk:=_variant_in_,   busy=&gt;_bool_out_,   dstblk=&gt;_variant_out_);</pre>	<p>Permet de copier les valeurs initiales d'un DB ou une partie de ces valeurs de la mémoire de chargement dans un DB cible en mémoire de travail.</p> <p>Le contenu de la mémoire de chargement n'est pas modifié pendant l'opération de copie.</p>
	<pre>WRIT_DBL (   req:=_bool_in_,   srcblk:=_variant_in_,   busy=&gt;_bool_out_,   dstblk=&gt;_variant_out_);</pre>	<p>Permet de copier les valeurs en cours d'un DB ou une partie de ces valeurs de la mémoire de travail dans un DB cible en mémoire de chargement.</p> <p>Le contenu de la mémoire de travail n'est pas modifié pendant l'opération de copie.</p>

Tableau 8- 142 Types de données pour les paramètres

Paramètre et type	Type de données	Description
REQ IN	BOOL	Un signal de niveau haut démarre l'opération si BUSY = 0.
SRCBLK IN	VARIANT	READ_DBL : Pointeur désignant le bloc de données source en mémoire de chargement WRIT_DBL : Pointeur désignant le bloc de données source en mémoire de travail
RET_VAL OUT	INT	Code d'erreur d'exécution
BUSY OUT	BOOL	BUSY = 1 signale que l'opération de lecture/écriture n'est pas achevée.
DSTBLK OUT	VARIANT	READ_DBL : Pointeur désignant le bloc de données de destination en mémoire de travail WRIT_DBL : Pointeur désignant le bloc de données de destination en mémoire de chargement

Typiquement, un DB est stocké à la fois dans la mémoire de chargement (mémoire flash) et dans la mémoire de travail (mémoire vive). Les valeurs initiales sont toujours stockées en mémoire de chargement et les valeurs en cours sont toujours stockées en mémoire de travail. On peut utiliser READ\_DBL pour copier un ensemble de valeurs initiales de la mémoire de chargement dans les valeurs en cours d'un DB en mémoire de travail auquel votre programme fait référence. Vous pouvez utiliser WRIT\_DBL pour actualiser les valeurs initiales stockées en mémoire de chargement interne ou dans une carte mémoire avec les valeurs en cours dans la mémoire de travail.

---

**Remarque**

**Limitez les opérations d'écriture en mémoire flash via WRIT\_DBL**

L'instruction WRIT\_DBL effectue des opérations d'écriture en mémoire flash (mémoire de chargement interne ou carte mémoire). Il convient d'utiliser cette instruction pour des mises à jour exceptionnelles, telles qu'une modification du processus de production.

---

Les blocs de données utilisés par READ\_DBL et WRIT\_DBL doivent avoir préalablement été créés par STEP 7 pour que vous puissiez utiliser ces instructions. Si le DB source est créé avec le type "standard", le DB de destination doit également être de type "standard". Si le bloc de données source est créé avec le type "optimisé", le bloc de données de destination doit également être de type "optimisé".

S'il s'agit de DB standard, vous pouvez indiquer un nom de variable ou une valeur P#. La valeur P# vous permet d'indiquer et de copier un nombre quelconque d'éléments de la taille précisée (octet, mot ou double mot). Vous pouvez ainsi copier tout ou partie d'un DB. S'il s'agit de DB optimisés, vous pouvez uniquement indiquer un nom de variable ; vous ne pouvez pas utiliser l'opérateur P#. Si vous indiquez un nom de variable pour des DB standard ou optimisés (ou pour d'autres types de mémoire de travail), c'est ce qui est désigné par cette variable qui est copié. Il peut s'agir d'un type utilisateur, d'un tableau ou d'un élément de base. Le type Struct peut uniquement être utilisé par ces instructions si le DB est standard, pas s'il est optimisé. Vous devez utiliser un type de données utilisateur (UDT) pour une structure en mémoire optimisée. Seul un type de données utilisateur garantit que les "types de données" sont exactement identiques à la fois pour la structure source et la structure cible.

---

**Remarque**

**Utilisation d'une structure (type de données Struct) dans un DB "optimisé"**

Lorsque vous utilisez un type de données Struct avec des DB "optimisés", vous devez d'abord créer un type de données utilisateur (UDT) pour la structure Struct. Vous configurez alors les deux DB - source et destination - avec le type UDT. Le type UDT garantit que les types de données à l'intérieur de la structure Struct restent cohérents pour les deux DB.

Pour les DB "standard", vous utilisez la structure Struct sans créer d'UDT.

---

READ\_DBL et WRIT\_DBL s'exécutent de manière asynchrone par rapport au cycle du programme. Le traitement s'étend sur plusieurs appels de READ\_DBL et WRIT\_DBL. Vous démarrez la tâche de transfert de DB en appelant l'instruction avec REQ = 1, puis vous surveillez les sorties BUSY et RET\_VAL pour déterminer si le transfert de données est achevé et correct.

Pour garantir la cohérence des données, ne modifiez pas la zone de destination pendant le traitement de READ\_DBL ou la zone source pendant le traitement de WRIT\_DBL (c'est-à-dire tant que le paramètre BUSY est VRAI).

Restrictions concernant les paramètres SRCBLK et DSTBLK :

- Un bloc de données doit avoir été créé avant de pouvoir être référencé.
- La longueur d'un pointeur VARIANT de type BOOL doit être divisible par 8.
- La longueur d'un pointeur VARIANT de type STRING doit être identique dans les pointeurs source et destination.

### Recettes et informations de configuration de machine

Vous pouvez utiliser les instructions READ\_DBL et WRIT\_DBL pour gérer les recettes et les informations de configuration de machine. Vous disposez ainsi d'une autre méthode pour assurer la rémanence de valeurs qui ne changent pas souvent, bien que vous deviez veiller à limiter le nombre d'écritures pour éviter l'usure prématurée de la mémoire flash. Dans les faits, cela permet d'augmenter la quantité de mémoire rémanente au-delà de la quantité normale prise en charge pour les mises hors tension, au moins pour les valeurs qui ne changent pas souvent. Vous pouvez, par exemple, sauvegarder des informations de recette ou de configuration de machine de la mémoire de travail dans la mémoire de chargement à l'aide de l'instruction WRIT\_DBL et transférer ce type d'informations de la mémoire de chargement dans la mémoire de travail à l'aide de l'instruction READ\_DBL.

Tableau 8- 143 Codes d'erreur

RET_VAL (W#16#...)	Description
0000	Pas d'erreur
0081	Avertissement : La zone source est plus petite que la zone de destination. Les données source sont copiées dans leur intégralité, les octets surnuméraires dans la zone de destination restant inchangés.
7000	Appel avec REQ = 0 : BUSY = 0
7001	Premier appel avec REQ = 1 (fonctionne) : BUSY = 1
7002	n-ième appel (fonctionne) : BUSY = 1
8051	Erreur de type de bloc de données
8081	La zone source est plus grande que la zone de destination. La zone de destination est complètement remplie et les octets restants de la source ne sont pas pris en compte.
8251	Erreur de type de bloc de données source
82B1	Bloc de données source manquant
82C0	Le DB source est en cours d'édition par une autre instruction ou une fonction de communication.
8551	Erreur de type de bloc de données de destination
85B1	Bloc de données de destination manquant
85C0	Le DB de destination est en cours d'édition par une autre instruction ou une fonction de communication.
80C3	Plus de 50 instructions READ_DBL ou 50 instructions WRIT_DBL sont actuellement en file d'attente pour exécution.

## 8.9 Codes d'erreur communs pour les instructions avancées

Tableau 8- 144 Codes d'erreur communs pour les instructions avancées

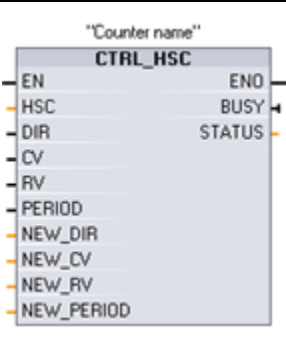
Code d'erreur (W#16#...) <sup>1</sup>	Description
8022	Zone trop petite pour l'entrée
8023	Zone trop petite pour la sortie
8024	Zone d'entrée illicite
8025	Zone de sortie illicite
8028	Affectation illicite de bit d'entrée
8029	Affectation illicite de bit de sortie
8030	La zone de sortie est un DB en lecture seule.
803A	DB inexistant

<sup>1</sup> Si l'une de ces erreurs se produit pendant l'exécution d'un bloc de code, la CPU passe à l'état ARRET à moins que vous n'ayez utilisé l'instruction GetError ou GetErrorID dans ce bloc de code et créé une réaction programmée à l'erreur.

## Instructions technologiques

### 9.1 Compteur rapide

Tableau 9- 1 Instruction CTRL\_HSC

CONT/LOG	SCL	Description
	<pre>"CTRL_HSC_0_DB" (   hsc:=_hw_hsc_in_,   dir:=_bool_in_,   cv:=_bool_in_,   rv:=_bool_in_,   period:=_bool_in_,   new_dir:=_int_in_,   new_cv:=_int_in_,   new_rv:=_dint_in_,   new_period:=_int_in_,   busy:=_bool_out_,   status:=_word_out_);</pre>	<p>Chaque instruction CTRL_HSC utilise une structure sauvegardée dans un DB afin de conserver les données. Vous affectez le DB lors du placement de l'instruction CTRL_HSC dans l'éditeur.</p>

- 1 STEP 7 crée automatiquement le DB lorsque vous insérez l'instruction.
- 2 Dans l'exemple SCL, "CTRL\_HSC\_0\_DB" est le nom du DB d'instance.

Tableau 9- 2 Types de données pour les paramètres

Paramètre et type	Type de données	Description	
HSC	IN	HW_HSC	Identificateur de compteur rapide
DIR <sup>1, 2</sup>	IN	Bool	1 = Demande de nouveau sens de comptage
CV <sup>1</sup>	IN	Bool	1 = Demande de paramétrage d'une nouvelle valeur de compteur
RV <sup>1</sup>	IN	Bool	1 = Demande de paramétrage d'une nouvelle valeur de référence
PERIOD <sup>1</sup>	IN	Bool	1 = Demande de paramétrage d'une nouvelle valeur de période (uniquement pour le mode de mesure de fréquence)
NEW_DIR	IN	Int	Nouveau sens de comptage : 1= incrémentation, -1= décrémentation
NEW_CV	IN	DInt	Nouvelle valeur de compteur
NEW_RV	IN	DInt	Nouvelle valeur de référence
NEW_PERIOD	IN	Int	Nouvelle valeur de période en secondes : 0.01, 0.1 ou 1 (uniquement pour le mode de mesure de fréquence)

9.1 Compteur rapide

Paramètre et type		Type de données	Description
BUSY <sup>3</sup>	OUT	Bool	Fonction occupée
STATUS	OUT	Word	Code d'erreur d'exécution

- <sup>1</sup> Si vous ne demandez pas l'actualisation d'une valeur de paramètre, les valeurs d'entrée correspondantes ne sont pas prises en compte.
- <sup>2</sup> Le paramètre DIR n'est valide que si le sens de comptage configuré a la valeur "Programme utilisat. (commande de sens interne)". Vous déterminez comment utiliser ce paramètre dans la configuration d'appareil HSC.
- <sup>3</sup> Pour un compteur rapide sur la CPU ou le Signal Board, le paramètre BUSY a toujours la valeur 0.

Vous configurez les paramètres pour chaque HSC dans la configuration d'appareil pour la CPU : mode de comptage, connexions d'E/S, affectation d'alarme et fonctionnement en tant que compteur rapide ou en tant qu'appareil de mesure de la fréquence d'impulsion.

Certains des paramètres du HSC peuvent être modifiés par votre programme utilisateur pour fournir une commande par programme du processus de comptage :

- Définir le sens de comptage à une valeur NEW\_DIR
- Définir la valeur de comptage en cours à une valeur NEW\_CV
- Définir la valeur de référence à une valeur NEW\_RV
- Définir la valeur de la période (pour le mode de mesure de fréquence) à une valeur NEW\_PERIOD

Si les valeurs des mémentos booléens suivants sont mises à 1 lors de l'exécution de l'instruction CTRL\_HSC, la valeur NEW\_xxx correspondante est chargée dans le compteur. Plusieurs demandes (plus d'un memento est mis à 1 au même moment) sont traitées lors d'une exécution unique de l'instruction CTRL\_HSC.

- DIR = 1 constitue une demande de chargement d'une valeur NEW\_DIR ; 0 = pas de modification
- CV = 1 constitue une demande de chargement d'une valeur NEW\_CV ; 0 = pas de modification
- RV = 1 constitue une demande de chargement d'une valeur NEW\_RV ; 0 = pas de modification
- PERIOD = 1 constitue une demande de chargement d'une valeur NEW\_PERIOD ; 0 = pas de modification

L'instruction CTRL\_HSC se place typiquement dans un OB d'alarme de processus qui est exécuté lorsque l'événement d'alarme de processus de compteur se produit. Par exemple, si un événement CV=RV déclenche l'alarme de compteur, un OB d'alarme de processus exécute l'instruction CTRL\_HSC et peut modifier la valeur de référence en chargeant une valeur NEW\_RV.

La valeur de comptage en cours n'est pas disponible dans les paramètres CTRL\_HSC. L'adresse de mémoire image qui contient la valeur de comptage en cours est définie pendant la configuration matérielle du compteur rapide. Vous pouvez vous servir de logique de programme pour lire directement la valeur de comptage. La valeur renvoyée à votre programme correspondra au comptage correct à l'instant où le compteur a été lu. Le compteur continuera à compter les événements rapides. Ainsi, la valeur de comptage en cours pourrait changer avant que votre programme n'ait terminé un processus basé sur une ancienne valeur en cours.



**Codes d'erreur :** En cas d'erreur, ENO est mis à 0 et la sortie STATUS contient un code d'erreur.

Tableau 9- 3 Valeurs de STATUS (W#16#)

STATUS	Description
0	Pas d'erreur
80A1	L'identificateur HSC n'accède pas à un compteur rapide.
80B1	Valeur incorrecte dans NEW_DIR
80B2	Valeur incorrecte dans NEW_CV
80B3	Valeur incorrecte dans NEW_RV
80B4	Valeur incorrecte dans NEW_PERIOD
80C0	Plusieurs accès au compteur rapide
80D0	Compteur rapide (HSC) non activé dans la configuration matérielle de la CPU

### 9.1.1 Fonctionnement du compteur rapide

Les compteurs rapides (HSC) comptent des événements se produisant plus rapidement que la vitesse d'exécution de l'OB. Si les événements à compter surviennent en deçà de la vitesse d'exécution de l'OB, vous pouvez utiliser des compteurs CTU, CTD ou CTUD. Si les événements surviennent plus rapidement que la vitesse d'exécution de l'OB, utilisez un compteur rapide. L'instruction CTRL\_HSC autorise la modification de certains des paramètres HSC par le programme utilisateur.

Vous pouvez, par exemple, utiliser un compteur rapide comme entrée pour un codeur angulaire incrémental. Le codeur angulaire fournit un nombre défini de valeurs de comptage par tour et une impulsion de réinitialisation par tour. Les horloges et l'impulsion de réinitialisation du codeur angulaire constituent les entrées du compteur rapide.

La première de plusieurs valeurs prédéfinies est chargée dans le compteur rapide et les sorties sont activées pour la durée où la valeur de comptage en cours est inférieure à la valeur prédéfinie en vigueur. Le compteur rapide fournit une alarme lorsque la valeur en cours est égale à la valeur prédéfinie, lorsqu'une réinitialisation se produit et lorsque le sens de comptage change.

Une nouvelle valeur prédéfinie est chargée et l'état suivant des sorties est activé à chaque fois que l'événement d'alarme "Valeur en cours égale à valeur prédéfinie" se produit. Lorsque l'événement d'alarme "Réinitialisation" survient, la première valeur prédéfinie ainsi que le premier état des sorties sont activés et le cycle se répète.

Comme les alarmes se produisent à une fréquence bien inférieure à la vitesse de comptage du compteur rapide, il est possible de réaliser une commande précise des opérations rapides qui a un impact relativement mineur sur le cycle de la CPU. La méthode d'association d'alarmes à des programmes d'alarme permet d'effectuer chaque chargement d'une nouvelle valeur prédéfinie dans un programme d'alarme distinct, ce qui facilite la commande. Mais vous pouvez aussi traiter tous les événements d'alarme dans un programme d'alarme unique.

Tableau 9- 4 Fréquence maximale (kHz)

HSC		Monophase	Biphase et quadrature de phase A/B
HSC1	CPU	100 kHz	80 kHz
	SB rapide	200 kHz	160 kHz
	SB	30 kHz	20 kHz
HSC2	CPU	100 kHz	80 kHz
	SB rapide	200 kHz	160 kHz
	SB	30 kHz	20 kHz
HSC3	CPU	100 kHz	80 kHz
HSC4	CPU	30 kHz	20 kHz
HSC5	CPU	30 kHz	20 kHz
	SB rapide	200 kHz	160 kHz
	SB	30 kHz	20 kHz
HSC6	CPU	30 kHz	20 kHz
	SB rapide	200 kHz	160 kHz
	SB	30 kHz	20 kHz

### Sélection des fonctions du HSC

Tous les compteurs rapides fonctionnent de la même manière dans un mode donné. Il existe quatre types fondamentaux de compteurs rapides :

- Compteur monophasé avec gestion interne du sens de comptage
- Compteur monophasé avec gestion externe du sens de comptage
- Compteur biphasé avec deux entrées d'horloge
- Compteur en quadrature de phase A/B

Vous pouvez utiliser chaque type de compteur rapide avec ou sans entrée de réinitialisation. Si vous activez l'entrée de réinitialisation (avec certaines restrictions, voir le tableau suivant), la valeur en cours est effacée et le reste jusqu'à ce que vous désactiviez l'entrée de réinitialisation.

- **Fonction fréquence** : Certains modes HSC permettent de configurer (type de comptage) le compteur rapide afin qu'il renvoie la fréquence et non une valeur de comptage d'impulsions en cours. Trois périodes de mesure de fréquence différentes sont disponibles : 0,01, 0,1 ou 1,0 seconde.

La période de mesure de fréquence détermine à quel rythme le compteur rapide calcule et renvoie une nouvelle valeur de fréquence. La fréquence renvoyée est une valeur moyenne déterminée par le nombre total de comptages pendant la dernière période de mesure. Si la fréquence change rapidement, la valeur renvoyée est une valeur intermédiaire entre la fréquence la plus élevée et la fréquence la plus faible apparaissant pendant la période de mesure. La fréquence est toujours indiquée en hertz (impulsions par seconde) quel que soit le paramétrage de la période de mesure de fréquence.

- **Modes et entrées du compteur** : Le tableau suivant montre les entrées utilisées pour les fonctions d'horloge, de gestion du sens de comptage et de réinitialisation associées aux compteurs rapides.

Une entrée ne peut servir pour deux fonctions différentes, mais toute entrée qui n'est pas utilisée pour le mode activé de son compteur rapide peut être utilisée à d'autres fins. Par exemple, si vous utilisez HSC1 dans un mode qui utilise des entrées intégrées mais n'utilise pas la réinitialisation externe (I0.3), vous pouvez utiliser I0.3 pour des alarmes sur front ou pour HSC2.

Tableau 9- 5 Modes de comptage des compteurs rapides

Type	Entrée 1	Entrée 2	Entrée 3	Fonction
Compteur monophasé avec gestion interne du sens de comptage	Horloge	(en option : sens de comptage)	-	Comptage ou fréquence
			Réinitialisation	Comptage
Compteur monophasé avec gestion externe du sens de comptage	Horloge	Sens	-	Comptage ou fréquence
			Réinitialisation	Comptage
Compteur biphasé avec deux entrées d'horloge	Horloge, incrémentation	Horloge, décrémentation	-	Comptage ou fréquence
			Réinitialisation	Comptage
Compteur en quadrature de phase A/B	Phase A	Phase B	-	Comptage ou fréquence
			Réinitialisation <sup>1</sup>	Comptage

<sup>1</sup> Pour un codeur : phase Z, position de référence

Adresses d'entrée pour les compteurs rapides

**Remarque**

Les E/S TOR utilisées par les compteurs rapides sont affectées pendant la configuration des appareils. Lorsque des adresses d'E/S TOR sont affectées à ces appareils, les valeurs dans les adresses d'E/S affectées ne peuvent pas être forcées par la fonction de forçage permanent dans une table de visualisation.

Lorsque vous configurez la CPU, vous avez la possibilité d'activer et de configurer chaque HSC. La CPU affecte automatiquement les adresses d'entrée pour chaque HSC selon sa configuration. Certains des HSC vous permettent de choisir soit l'utilisation des entrées intégrées de la CPU, soit l'utilisation des entrées d'un SB.

**IMPORTANT**

Comme on le voit dans les tableaux suivants, les affectations par défaut pour les signaux optionnels des différents HSC se recoupent. Par exemple, la réinitialisation externe optionnelle pour le HSC 1 utilise la même entrée que l'une des entrées pour le HSC 2.

Vérifiez toujours que vous avez configuré vos compteurs rapides de sorte qu'une entrée **n'est pas** utilisée par deux compteurs.

Le tableau suivant montre les affectations des entrées HSC pour les E/S intégrées de la CPU 1211C et pour un SB (si le SB a uniquement 2 entrées, seules les entrées 4.0 et 4.1 sont disponibles).

- Pour un compteur monophasé : C est l'entrée d'horloge, [d] est l'entrée de sens de comptage optionnelle et [R] est une entrée de réinitialisation externe optionnelle (la réinitialisation est disponible uniquement pour le mode "comptage").
- Pour un compteur biphasé : Cu est l'entrée d'incrément, CD est l'entrée de décrémentation et [R] est une entrée de réinitialisation externe optionnelle (la réinitialisation est disponible uniquement pour le mode "comptage").
- Pour un compteur en quadrature de phase A/B : A est l'entrée d'horloge A, B est l'entrée d'horloge B et [R] est une entrée de réinitialisation externe optionnelle (la réinitialisation est disponible uniquement pour le mode "comptage").

Tableau 9- 6 Affectations des entrées HSC pour la CPU 1211C

HSC		Entrée intégrée à la CPU (0.x)						Entrée SB (4.x par défaut) <sup>3</sup>			
		0	1	2	3	4	5	0	1	2	3
HSC 1 <sup>1</sup>	Monophasé	C	[d]		[R]			C	[d]		[R]
	Biphasé	CU	CD		[R]			CU	CD		[R]
	Phase A/B	A	B		[R]			A	B		[R]
HSC 2 <sup>1</sup>	Monophasé		[R]	C	[d]				[R]	C	[d]
	Biphasé		[R]	CU	CD				[R]	CU	CD
	Phase A/B		[R]	A	B				[R]	A	B
HSC 3	Monophasé					C	[d]				

HSC		Entrée intégrée à la CPU (0.x)						Entrée SB (4.x par défaut) <sup>3</sup>			
		0	1	2	3	4	5	0	1	2	3
	Biphase					CU	CD				
	Phase A/B					A	B				
HSC 5 <sup>2</sup>	Monophasé							C	[d]		[R]
	Biphase							CU	CD		[R]
	Phase A/B							A	B		[R]
HSC 6 <sup>2</sup>	Monophasé								[R]	C	[d]
	Biphase								[R]	CU	CD
	Phase A/B								[R]	A	B

<sup>1</sup> Les HSC 1 et HSC 2 peuvent être configurés soit pour les entrées intégrées, soit pour un SB.

<sup>2</sup> Les HSC 5 et HSC 6 sont disponibles uniquement avec un SB. Le HSC 6 est disponible uniquement avec un SB à 4 entrées.

<sup>3</sup> Un SB avec seulement 2 entrées TOR comporte uniquement les entrées 4.0 et 4.1.

Le tableau suivant montre les affectations des entrées HSC pour les E/S intégrées de la CPU 1212C et pour un SB (si le SB a uniquement 2 entrées, seules les entrées 4.0 et 4.1 sont disponibles).

- Pour un compteur monophasé : C est l'entrée d'horloge, [d] est l'entrée de sens de comptage optionnelle et [R] est une entrée de réinitialisation externe optionnelle (la réinitialisation est disponible uniquement pour le mode "comptage").
- Pour un compteur biphase : Cu est l'entrée d'incrément, CD est l'entrée de décrémentation et [R] est une entrée de réinitialisation externe optionnelle (la réinitialisation est disponible uniquement pour le mode "comptage").
- Pour un compteur en quadrature de phase A/B : A est l'entrée d'horloge A, B est l'entrée d'horloge B et [R] est une entrée de réinitialisation externe optionnelle (la réinitialisation est disponible uniquement pour le mode "comptage").

Tableau 9-7 Affectations des entrées HSC pour la CPU 1212C

HSC		Entrée intégrée à la CPU (0.x)								Entrée SB (4.x) <sup>3</sup>			
		0	1	2	3	4	5	6	7	0	1	2	3
HSC 1 <sup>1</sup>	Monophasé	C	[d]		[R]					C	[d]		[R]
	Biphase	CU	CD		[R]					CU	CD		[R]
	Phase A/B	A	B		[R]					A	B		[R]
HSC 2 <sup>1</sup>	Monophasé		[R]	C	[d]						[R]	C	[d]
	Biphase		[R]	CU	CD						[R]	CU	CD
	Phase A/B		[R]	A	B						[R]	A	B
HSC 3	Monophasé					C	[d]		[R]				
	Biphase					CU	CD		[R]				
	Phase A/B					A	B		[R]				
HSC 4	Monophasé						[R]	C	[d]				
	Biphase						[R]	CU	CD				
	Phase A/B						[R]	A	B				

9.1 Compteur rapide

HSC		Entrée intégrée à la CPU (0.x)								Entrée SB (4.x) <sup>3</sup>			
		0	1	2	3	4	5	6	7	0	1	2	3
HSC 5 <sup>2</sup>	Monophasé									C	[d]		[R]
	Biphase									CU	CD		[R]
	Phase A/B									A	B		[R]
HSC 6 <sup>2</sup>	Monophasé										[R]	C	[d]
	Biphase										[R]	CU	CD
	Phase A/B										[R]	A	B

- <sup>1</sup> Les HSC 1 et HSC 2 peuvent être configurés soit pour les entrées intégrées, soit pour un SB.
- <sup>2</sup> Les HSC 5 et HSC 6 sont disponibles uniquement avec un SB. Le HSC 6 est disponible uniquement avec un SB à 4 entrées.
- <sup>3</sup> Un SB avec seulement 2 entrées TOR comporte uniquement les entrées 4.0 et 4.1.

Les deux tableaux suivants montrent les affectations des entrées HSC pour les E/S intégrées de la CPU 1214C et pour un SB optionnel, le cas échéant.

- Pour un compteur monophasé : C est l'entrée d'horloge, [d] est l'entrée de sens de comptage optionnelle et [R] est une entrée de réinitialisation externe optionnelle (la réinitialisation est disponible uniquement pour le mode "comptage").
- Pour un compteur biphase : Cu est l'entrée d'incrément, CD est l'entrée de décrémentation et [R] est une entrée de réinitialisation externe optionnelle (la réinitialisation est disponible uniquement pour le mode "comptage").
- Pour un compteur en quadrature de phase A/B : A est l'entrée d'horloge A, B est l'entrée d'horloge B et [R] est une entrée de réinitialisation externe optionnelle (la réinitialisation est disponible uniquement pour le mode "comptage").

Tableau 9- 8 Affectations des entrées HSC pour les CPU 1214C et CPU 1215C (entrées intégrées uniquement)

HSC		Entrée TOR 0 (par défaut : 0.x)								Entrée TOR 1 (par défaut : 1.x)					
		0	1	2	3	4	5	6	7	0	1	2	3	4	5
HSC 1 <sup>1</sup>	Monophasé	C	[d]		[R]										
	Biphase	CU	CD		[R]										
	Phase A/B	A	B		[R]										
HSC 2 <sup>1</sup>	Monophasé		[R]	C	[d]										
	Biphase		[R]	CU	CD										
	Phase A/B		[R]	A	B										
HSC 3	Monophasé					C	[d]		[R]						
	Biphase					CU	CD		[R]						
	Phase A/B					A	B		[R]						
HSC 4	Monophasé						[R]	C	[d]						
	Biphase						[R]	CU	CD						
	Phase A/B						[R]	A	B						
HSC 5 <sup>1</sup>	Monophasé									C	[d]	[R]			
	Biphase									CU	CD	[R]			
	Phase A/B									A	B	[R]			

HSC		Entrée TOR 0 (par défaut : 0.x)							Entrée TOR 1 (par défaut : 1.x)						
		0	1	2	3	4	5	6	7	0	1	2	3	4	5
HSC 6 <sup>1</sup>	Monophase												C	[d]	[R]
	Biphase												CU	CD	[R]
	Phase A/B												A	B	[R]

<sup>1</sup> Les HSC 1, HSC 2, HSC 5 et HSC 6 peuvent être configurés soit pour les entrées intégrées, soit pour un SB.

Tableau 9- 9 Affectations des entrées HSC pour des SB

HSC <sup>1</sup>		Entrées de SB (par défaut : 4.x) <sup>2</sup>			
		0	1	2	3
HSC 1	Monophase	C	[d]		[R]
	Biphase	CU	CD		[R]
	Phase A/B	A	B		[R]
HSC 2	Monophase		[R]	C	[d]
	Biphase		[R]	CU	CD
	Phase A/B		[R]	A	B
HSC 5	Monophase	C	[d]		[R]
	Biphase	CU	CD		[R]
	Phase A/B	A	B		[R]
HSC 6	Monophase		[R]	C	[d]
	Biphase		[R]	CU	CD
	Phase A/B		[R]	A	B

<sup>1</sup> Pour la CPU 1214C : Les HSC 1, HSC 2, HSC 5 et HSC 6 peuvent être configurés soit pour les entrées intégrées, soit pour un SB.

<sup>2</sup> Un SB avec seulement 2 entrées TOR comporte uniquement les entrées 4.0 et 4.1.

## Accès à la valeur en cours du compteur rapide

### Remarque

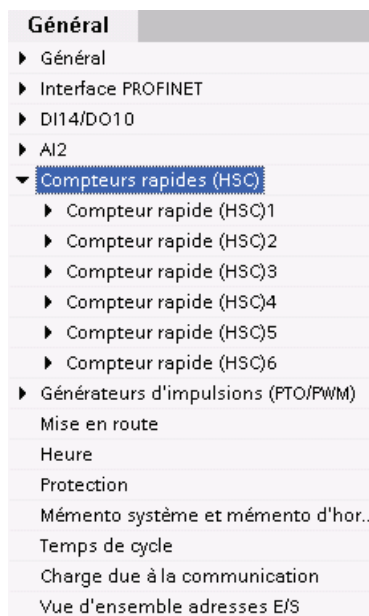
Lorsque vous activez un générateur d'impulsions pour utilisation en tant que PTO, un compteur rapide correspondant est affecté à cette fonction PTO. Le HSC1 est affecté à PTO1 et le HSC2 est affecté à PTO2. Le compteur rapide affecté appartient entièrement à la voie PTO et l'entrée ordinaire du compteur est désactivée. La valeur du compteur rapide est uniquement utilisée pour la fonctionnalité interne. Vous ne pouvez pas visualiser la valeur en cours (par exemple, dans ID1000) lorsque des impulsions surviennent.

La CPU sauvegarde la valeur en cours de chaque compteur rapide dans une adresse d'entrée (I). Le tableau suivant montre les adresses par défaut affectées à la valeur en cours pour chaque HSC. Vous pouvez modifier l'adresse I pour la valeur en cours en modifiant les propriétés de la CPU dans la configuration des appareils.

Tableau 9- 10 Valeur en cours du compteur rapide

HSC	Type de données	Adresse par défaut
HSC1	DInt	ID1000
HSC2	DInt	ID1004
HSC3	DInt	ID1008
HSC4	DInt	ID1012
HSC5	DInt	ID1016
HSC6	DInt	ID1020

### 9.1.2 Configuration du HSC



La CPU vous permet de configurer 6 compteurs rapides au maximum. Vous éditez les "Propriétés" de la CPU pour configurer les paramètres de chaque compteur rapide individuel.

Utilisez l'opération CTRL\_HSC dans votre programme utilisateur pour contrôler le fonctionnement du HSC.

Activez le HSC spécifique en sélectionnant l'option "Activer" pour cet HSC.



#### Remarque

Lorsque vous activez le compteur rapide et que vous lui associez des entrées, le filtre pour ces entrées est paramétré à 800 ns. Chaque entrée présente une configuration de filtre unique qui s'applique à toutes les utilisations : entrées du processus, alarmes, capture d'impulsions et entrées HSC.



**! ATTENTION**

Si le temps de filtre pour une voie d'entrée TOR est modifié par rapport à un paramétrage précédent, une nouvelle valeur d'entrée de niveau "0" peut devoir être présente pendant une durée cumulée allant jusqu'à 20,0 ms pour que le filtre réagisse pleinement aux nouvelles entrées. Pendant ce temps, les événements d'impulsion "0" courts de moins de 20,0 ms peuvent ne pas être détectés ni comptés.

Cette modification des temps de filtre peut provoquer un fonctionnement inattendu des machines ou du processus, pouvant entraîner la mort, des blessures graves et des dommages matériels importants.

Pour garantir la prise en compte immédiate d'un nouveau temps de filtre, mettez la CPU hors tension puis sous tension.

Une fois le HSC activé, configurez les autres paramètres, tels que la fonction de comptage, les valeurs initiales, les options de réinitialisation et les événements d'alarme.

The screenshot shows a configuration window for HSC with the following settings:

- Type de comptage: Comptage
- Phase de fonctionnement: Monophasé
- Source d'entrée: Entrée de CPU intégrée
- Sens de comptage donné par: Programme utilisat. (commande)
- Sens de comptage initial: Incrémental
- Période de mesure de fréquence: +/- sec

Pour des informations sur la configuration du HSC, reportez-vous à la section configuration de la CPU (Page 131).

## 9.2 Régulation PID

STEP 7 fournit les instructions PID suivantes pour la CPU S7-1200 :

- L'instruction PID\_Compact sert à la régulation de processus industriels à grandeurs d'entrée et de sortie continues.
- L'instruction PID\_3Step sert à la régulation de dispositifs actionnés par moteur, tels que des vannes, qui nécessitent des signaux discrets pour la commande d'ouverture et de fermeture.

---

**Remarque**

Les modifications que vous apportez à la configuration PID et que vous chargez à l'état MARCHE ne prennent effet qu'après un passage de la CPU de l'état ARRET à l'état MARCHE.

---

Ces deux instructions PID (PID\_3Step et PID\_Compact) peuvent calculer les actions P, I et D lors de la mise en route (si "l'optimisation préalable" est configurée). Vous pouvez également configurer l'instruction pour une "optimisation fine" qui vous permet d'optimiser les paramètres. Vous n'avez pas besoin de déterminer les paramètres manuellement.

---

**Remarque**

**Exécutez l'instruction PID à intervalles constants de la période d'échantillonnage (de préférence dans un OB d'alarme cyclique).**

Comme la boucle PID a besoin d'un certain temps pour réagir aux variations de la valeur de réglage, ne calculez pas la valeur de réglage à chaque cycle. N'exécutez pas l'instruction PID dans l'OB de cycle de programme principal (tel que l'OB 1).

---

La période d'échantillonnage de l'algorithme PID représente le temps entre deux calculs de la valeur de réglage. La valeur de réglage est calculée pendant l'auto-régulation et arrondie à un multiple de la période. Toutes les autres fonctions de l'instruction PID sont exécutées à chaque appel.

## Algorithme PID

Le régulateur PID (actions proportionnelle/intégrale/dérivée) mesure l'intervalle de temps entre deux appels, puis évalue les résultats pour surveiller la période d'échantillonnage. Une valeur moyenne de la période d'échantillonnage est générée à chaque changement de mode ainsi qu'à la mise en route initiale. Cette valeur sert de référence à la fonction de surveillance et est utilisée pour les calculs. La surveillance porte sur le temps de mesure actuel entre deux appels et sur la valeur moyenne de la période d'échantillonnage définie du régulateur.

La valeur de réglage pour le régulateur PID est constituée de trois actions :

- Action proportionnelle P : La valeur de réglage calculée par l'action P est proportionnelle à l'écart entre la consigne et la mesure.
- Action intégrale I : La valeur de réglage calculée par l'action I augmente proportionnellement à la durée de l'écart entre la consigne et la mesure pour finalement compenser l'écart.
- Action dérivée D : La valeur de réglage calculée par l'action D augmente avec l'accroissement de la vitesse de modification de l'écart entre la consigne et la mesure. La valeur de réglage est synchronisée sur la consigne le plus rapidement possible.

Le régulateur PID utilise la formule suivante pour calculer la valeur de réglage pour l'instruction PID\_Compact.

$$y = K_p \left[ (b \cdot w - x) + \frac{1}{T_i \cdot s} (w - x) + \frac{T_d \cdot s}{a \cdot T_d \cdot s + 1} (c \cdot w - x) \right]$$

y	Valeur de réglage	x	Mesure
w	Consigne	s	Opérateur de Laplace
K <sub>p</sub>	Gain proportionnel (action P)	a	Coefficient de retard de l'action dérivée (action D)
T <sub>i</sub>	Temps d'intégration (action I)	b	Pondération de l'action proportionnelle (action P)
T <sub>d</sub>	Temps de dérivation (action D)	c	Pondération de l'action dérivée (action D)

Le régulateur PID utilise la formule suivante pour calculer la valeur de réglage pour l'instruction PID\_3Step.

$$\Delta y = K_p \cdot s \cdot \left[ (b \cdot w - x) + \frac{1}{T_i \cdot s} (w - x) + \frac{T_d \cdot s}{a \cdot T_d \cdot s + 1} (c \cdot w - x) \right]$$

y	Valeur de réglage	x	Mesure
w	Consigne	s	Opérateur de Laplace
K <sub>p</sub>	Gain proportionnel (action P)	a	Coefficient de retard de l'action dérivée (action D)
T <sub>i</sub>	Temps d'intégration (action I)	b	Pondération de l'action proportionnelle (action P)
T <sub>d</sub>	Temps de dérivation (action D)	c	Pondération de l'action dérivée (action D)

## 9.2.1 Insertion de l'instruction PID et de l'objet technologique

STEP 7 fournit deux instructions pour la régulation PID :

- L'instruction PID\_Compact et son objet technologique associé fournissent un régulateur PID universel optimisable. L'objet technologique contient tous les paramètres pour la boucle de régulation.
- L'instruction PID\_3Step et son objet technologique associé fournissent un régulateur PID avec des paramètres spécifiques pour les vannes commandées par moteur. L'objet technologique contient tous les paramètres pour la boucle de régulation. Le régulateur PID\_3Step comporte deux sorties booléennes supplémentaires.

Une fois l'objet technologique créé, vous devez configurer les paramètres (Page 388). Vous ajustez également les paramètres d'autooptimisation ("optimisation préalable" à la mise en route ou "optimisation fine" manuelle) pour mettre en service le régulateur PID (Page 390).

Tableau 9- 11 Insertion de l'instruction PID et de l'objet technologique

Lorsque vous insérez une instruction PID dans votre programme utilisateur, STEP 7 crée automatiquement un objet technologique et un DB d'instance pour l'instruction. Le DB d'instance contient tous les paramètres utilisés par l'instruction PID. Chaque instruction PID doit avoir son propre DB d'instance unique pour fonctionner correctement.

Après l'insertion de l'instruction PID et la création de l'objet technologique et du DB d'instance, vous configurez les paramètres pour l'objet technologique (Page 388).

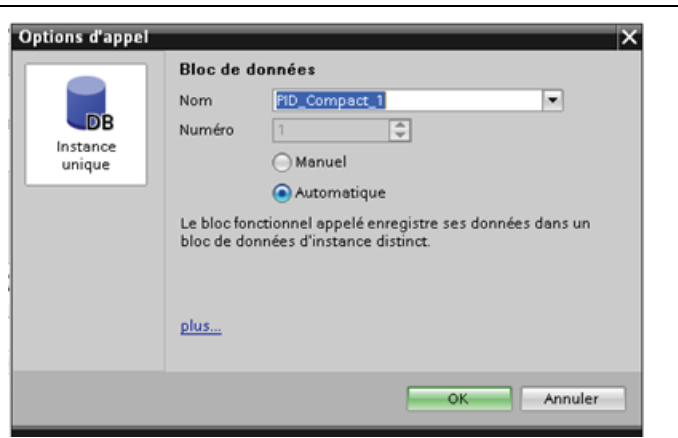
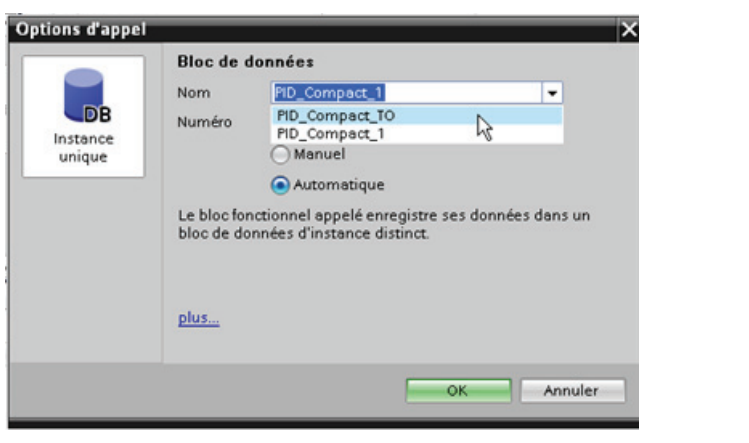

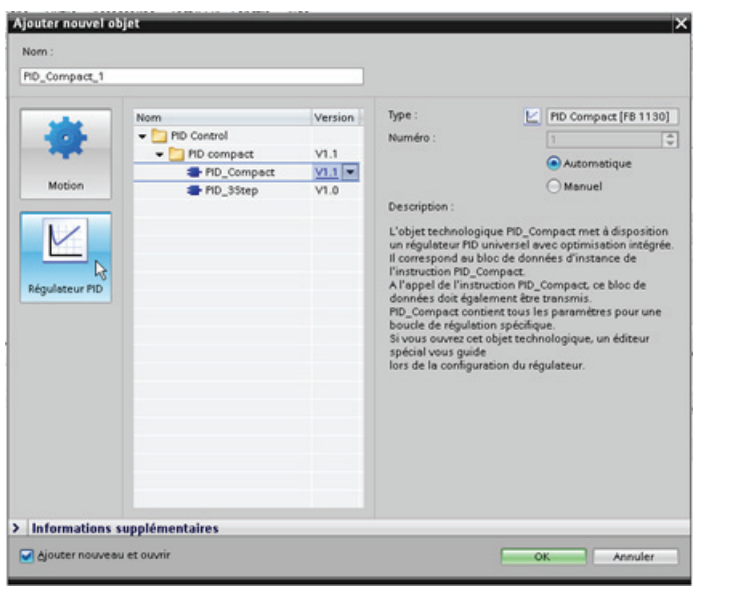


Tableau 9- 12 (Facultatif) Création d'un objet technologique dans le navigateur du projet

<p>Vous pouvez également créer des objets technologiques pour votre projet <b>avant</b> d'insérer l'instruction PID. Si vous avez créé l'objet technologique avant d'insérer une instruction PID dans votre programme utilisateur, vous pourrez sélectionner cet objet technologique à l'insertion de l'instruction PID.</p>	
<p>Pour créer un objet technologique, double-cliquez sur l'icône "Ajouter nouvel objet" dans le navigateur du projet.</p>	
<p>Cliquez sur l'icône "Régulation" et sélectionnez l'objet technologique correspondant au type de régulateur PID (PID_Compact ou PID_3Step). Vous pouvez créer un nom facultatif pour l'objet technologique. Cliquez sur "OK" pour créer l'objet technologique.</p>	

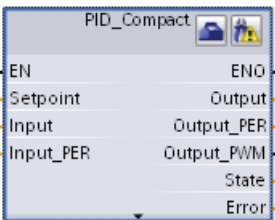
### 9.2.2 Instruction PID\_Compact

Le régulateur PID utilise la formule suivante pour calculer la valeur de réglage pour l'instruction PID\_Compact.

$$y = K_p \left[ (b \cdot w - x) + \frac{1}{T_i \cdot s} (w - x) + \frac{T_D \cdot s}{a \cdot T_D \cdot s + 1} (c \cdot w - x) \right]$$

y	Valeur de réglage	x	Mesure
w	Consigne	s	Opérateur de Laplace
K <sub>p</sub>	Gain proportionnel (action P)	a	Coefficient de retard de l'action dérivée (action D)
T <sub>i</sub>	Temps d'intégration (action I)	b	Pondération de l'action proportionnelle (action P)
T <sub>D</sub>	Temps de dérivation (action D)	c	Pondération de l'action dérivée (action D)

Tableau 9- 13 Instruction PID\_Compact

CONT/LOG	SCL	Description
	<pre>"PID_Compact_1" (   Setpoint:=_real_in_,   Input:=_real_in_,   Input_PER:=_word_in_,   ManualEnable:=_bool_in_,   ManualValue:=_real_in_,   Reset:=_bool_in_,   ScaledInput=&gt;_real_out_,   Output=&gt;_real_out_,   Output_PER=&gt;_word_out_,   Output_PWM=&gt;_bool_out_,   SetpointLimit_H=&gt;_bool_out_,   SetpointLimit_L=&gt;_bool_out_,   InputWarning_H=&gt;_bool_out_,   InputWarning_L=&gt;_bool_out_,   State=&gt;_int_out_,   Error=&gt;_dword_out_);</pre>	<p>PID_Compact fournit un régulateur PID à fonction d'autooptimisation pour les modes automatique et manuel. PID_Compact est un régulateur PIDT1 avec anti-emballement (anti-windup) et pondération des actions P et D.</p>

<sup>1</sup> STEP 7 crée automatiquement l'objet technologique et le DB d'instance lorsque vous insérez l'instruction. Le DB d'instance contient les paramètres de l'objet technologique.

<sup>2</sup> Dans l'exemple SCL, "PID\_Compact\_1" est le nom du DB d'instance.

Tableau 9- 14 Types de données pour les paramètres

Paramètre et type		Type de données	Description
Setpoint	IN	Real	Consigne du régulateur PID en mode automatique. Valeur par défaut : 0.0
Input	IN	Real	Mesure. Valeur par défaut : 0.0 Vous devez également paramétrer sPid_Cmpt.b_Input_PER_On à FAUX.
Input_PER	IN	Word	Mesure analogique (optionnelle). Valeur par défaut : W#16#0 Vous devez également paramétrer sPid_Cmpt.b_Input_PER_On à VRAI.
ManualEnable	IN	Bool	Active ou désactive le mode de fonctionnement manuel. Valeur par défaut : FALSE: <ul style="list-style-type: none"> <li>• PID_Compact V1.0 et V1.2 : Lorsque la CPU passe à l'état MARCHE, PID_Compact démarre en mode manuel si ManualEnable = TRUE. Une transition de FALSE à TRUE n'est pas nécessaire pour mettre PID_Compact en mode manuel.</li> <li>• PID_Compact V1.1 : Lorsque la CPU passe à l'état MARCHE, PID Compact démarre dans le dernier mode actif si ManualEnable = TRUE. Une transition de TRUE à FALSE et à TRUE est nécessaire pour mettre PID_Compact en mode manuel.</li> </ul>
ManualValue	IN	Real	Valeur de réglage pour le mode manuel. Valeur par défaut : 0.0
Reset	IN	Bool	Le paramètre Reset redémarre le régulateur. Valeur par défaut : FAUX Voir les diagrammes de réaction à la réinitialisation pour <b>PID_Compact</b> V1.1 et V1.0 dans le paragraphe "Réaction à la réinitialisation" ci-après.
ScaledInput	OUT	Real	Mesure mise à l'échelle. Valeur par défaut : 0.0
Output <sup>1</sup>	OUT	Real	Valeur de réglage. Valeur par défaut : 0.0
Output_PER <sup>1</sup>	OUT	Word	Valeur de réglage analogique. Valeur par défaut : W#16#0
Output_PWM <sup>1</sup>	OUT	Bool	Valeur de réglage pour la modulation de largeur d'impulsion. Valeur par défaut : FAUX
SetpointLimit_H	OUT	Bool	Limite supérieure de la consigne. Valeur par défaut : FAUX Si SetpointLimit_H est VRAI, la limite supérieure absolue de la consigne est atteinte. Valeur par défaut : FAUX
SetpointLimit_L	OUT	Bool	Limite inférieure de la consigne. Valeur par défaut : FAUX Si SetpointLimit_L est VRAI, la limite inférieure absolue de la consigne est atteinte. Valeur par défaut : FAUX
InputWarning_H	OUT	Bool	Si InputWarning_H est VRAI, la mesure a atteint ou dépassé la limite d'alerte supérieure. Valeur par défaut : FAUX
InputWarning_L	OUT	Bool	Si InputWarning_L est VRAI, la mesure a atteint ou dépassé la limite d'alerte inférieure. Valeur par défaut : FAUX

Paramètre et type		Type de données	Description
State	OUT	Int	Mode de fonctionnement en cours du régulateur PID. Valeur par défaut : 0 Utilisez sRet.i_Mode pour changer le mode. <ul style="list-style-type: none"> <li>• State = 0 : Inactif</li> <li>• State = 1 : Optimisation préalable</li> <li>• State = 2 : Optimisation fine manuelle</li> <li>• State = 3 : Mode automatique</li> <li>• State = 4 : Mode manuel</li> </ul>
ErrorBits	OUT	DWord	Le tableau des paramètres (Page 379) ErrorBits de l'instruction PID_Compact présente les messages d'erreur. Valeur par défaut : DW#16#0000 (pas d'erreur)

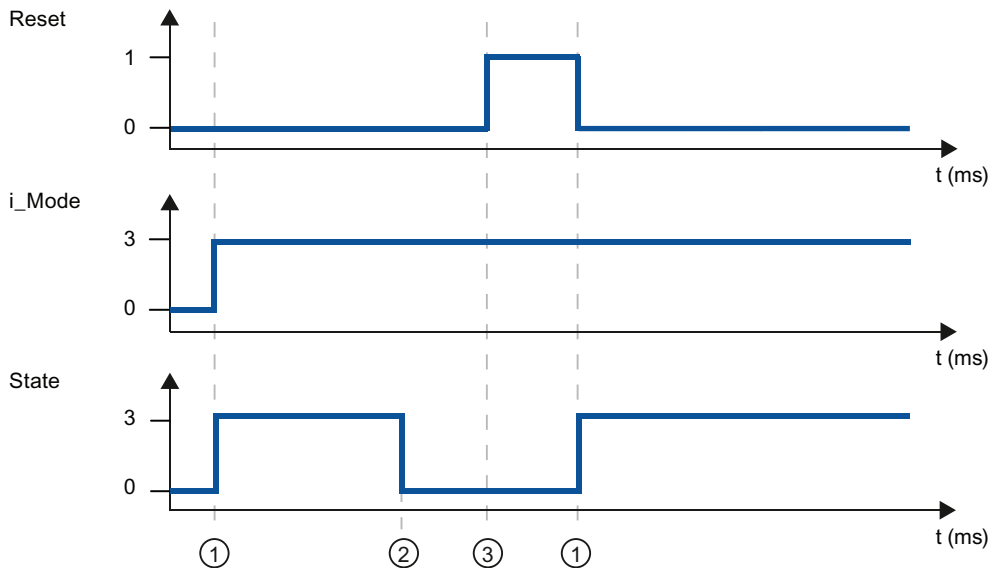
<sup>1</sup> Les paramètres de sortie Output, Output\_PER et Output\_PWM peuvent être utilisés en parallèle.

### Réaction à la réinitialisation

La réaction à Reset = TRUE dépend de la version de l'instruction PID\_Compact.

#### Réaction à la réinitialisation de PID\_Compact V1.1

Un front montant du paramètre Reset réinitialise les erreurs et alertes et efface l'action intégrale. Un front descendant du paramètre Reset déclenche le passage au dernier mode de fonctionnement actif.

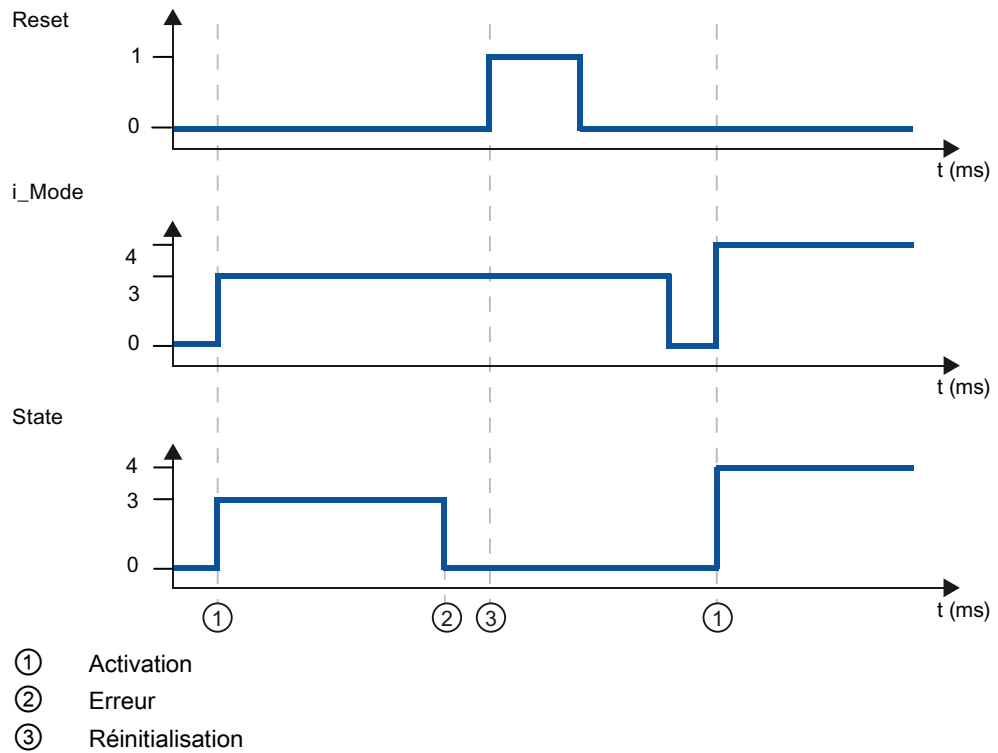


- ① Activation
- ② Erreur
- ③ Réinitialisation



### Réaction à la réinitialisation de PID\_Compact V1.0

Un front montant du paramètre Reset réinitialise les erreurs et alertes et efface l'action intégrale. Le régulateur n'est réactivé qu'au front suivant sur i\_Mode.



Fonctionnement du régulateur PID\_Compact

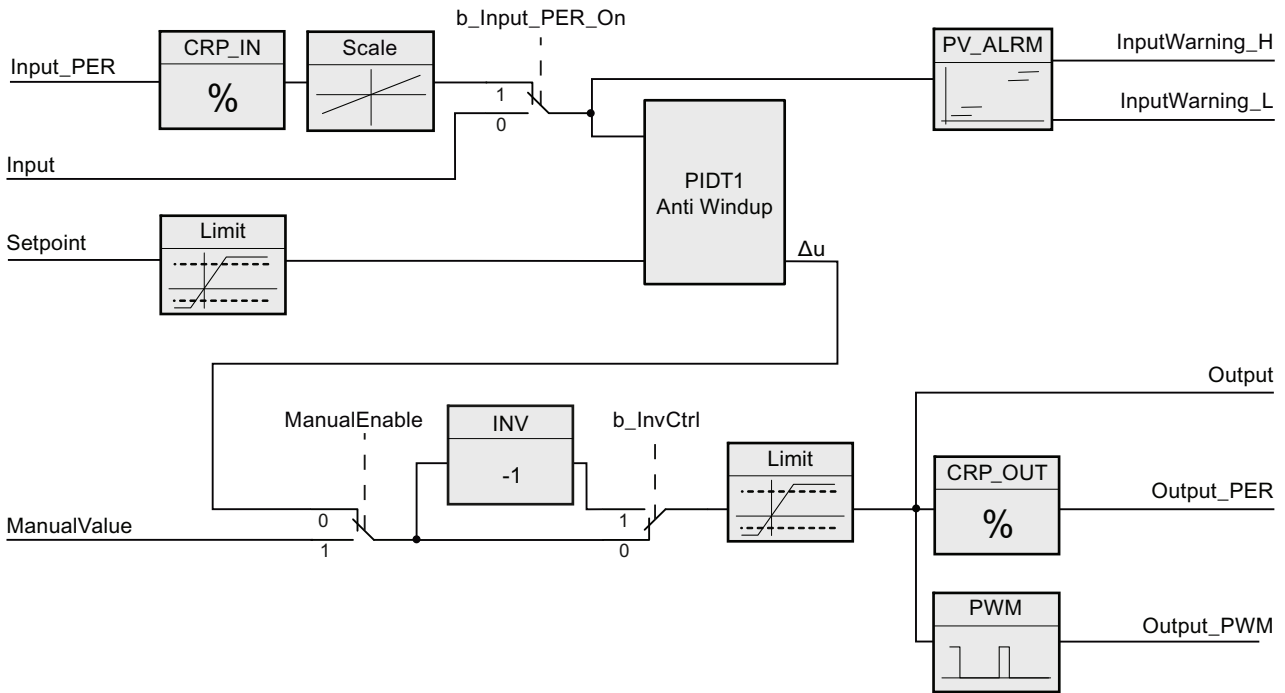


Figure 9-1 Fonctionnement du régulateur PID\_Compact

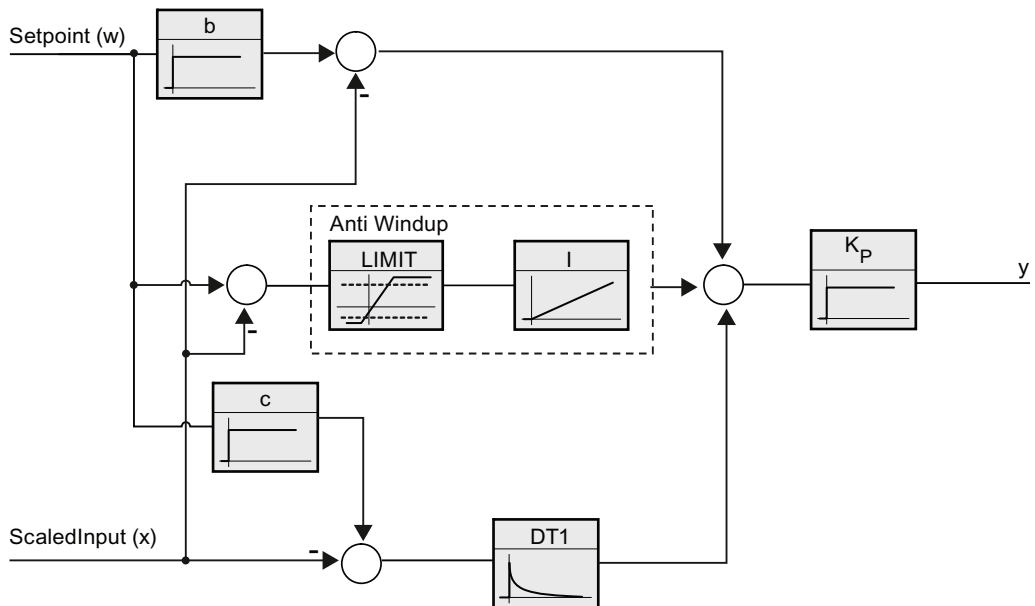


Figure 9-2 Fonctionnement du régulateur PID\_Compact en tant que régulateur PIDT1 avec anti-emballement

### 9.2.3 Paramètre ErrorBits de l'instruction PID\_Compact

Si plusieurs erreurs sont en attente, les valeurs des codes d'erreur sont données au moyen d'une addition binaire. Ainsi, l'affichage du code d'erreur 0003 signifie que les erreurs 0001 et 0002 sont également en attente.

Tableau 9- 15 Paramètre ErrorBits de l'instruction PID\_Compact

ErrorBit (DW#16#...)	Description
0000	Pas d'erreur
0001	Le paramètre "Input" se situe en dehors des limites de la mesure. Input > sPid_Cmpt.r_Pv_Hlm ou Input < sPid_Cmpt.r_Pv_Llm Vous ne pourrez redémarrer l'actionneur qu'après élimination de l'erreur.
0002	Valeur invalide dans le paramètre "Input_PER". Vérifiez s'il y a une erreur en attente au niveau de l'entrée analogique.
0004	Erreur pendant l'optimisation fine. L'oscillation de la mesure n'a pas pu être conservée.
0008	Erreur au démarrage de l'optimisation préalable. La mesure est trop proche de la consigne. Démarrez l'optimisation fine.
0010	La consigne a été modifiée pendant l'optimisation du régulateur.
0020	Impossible d'exécuter une optimisation préalable en mode automatique ou pendant l'optimisation fine.
0040	Erreur pendant l'optimisation fine. La consigne est trop proche des limites de la consigne.
0080	Configuration incorrecte des limites de la valeur de réglage. Vérifiez que les limites de la valeur de réglage sont configurées correctement et correspondent au sens dans lequel se fait la commande.
0100	Une erreur pendant l'optimisation du régulateur a entraîné des paramètres invalides.
0200	Valeur invalide dans le paramètre "Input" : Le format numérique de la valeur est invalide.
0400	Le calcul de la valeur de réglage a échoué. Vérifiez les paramètres PID.
0800	Erreur de période d'échantillonnage : PID_Compact n'est pas appelé dans la période d'échantillonnage de l'OB d'alarme cyclique.
1000	Valeur invalide dans le paramètre "Setpoint" : Le format numérique de la valeur est invalide.

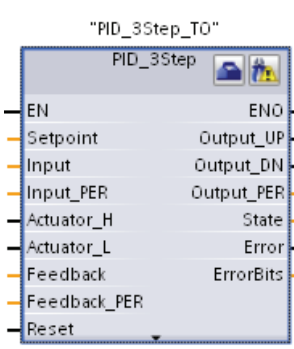
### 9.2.4 Instruction PID\_3STEP

Le régulateur PID utilise la formule suivante pour calculer la valeur de réglage pour l'instruction PID\_3Step.

$$\Delta y = K_p \cdot s \cdot \left[ (b \cdot w - x) + \frac{1}{T_i \cdot s} (w - x) + \frac{T_D \cdot s}{a \cdot T_D \cdot s + 1} (c \cdot w - x) \right]$$

y	Valeur de réglage	x	Mesure
w	Consigne	s	Opérateur de Laplace
K <sub>p</sub>	Gain proportionnel (action P)	a	Coefficient de retard de l'action dérivée (action D)
T <sub>i</sub>	Temps d'intégration (action I)	b	Pondération de l'action proportionnelle (action P)
T <sub>D</sub>	Temps de dérivation (action D)	c	Pondération de l'action dérivée (action D)

Tableau 9- 16 Instruction PID\_3Step

CONT/LOG	SCL	Description
	<pre>"PID_3Step_1" (   SetpoInt:=_real_in_,   Input:=_real_in_,   ManualValue:=_real_in_,   Feedback:=_real_in_,   InputPer:=_word_in_,   FeedbackPer:=_word_in_,   ManualEnable:=_bool_in_,   ManualUP:=_bool_in_,   ManualDN:=_bool_in_,   ActuatorH:=_bool_in_,   ActuatorL:=_bool_in_,   Reset:=_bool_in_,   ScaledInput=&gt;_real_out_,   ScaledFeedback=&gt;_real_out_,   ErrorBits=&gt;_dword_out_,   OutputPer=&gt;_word_out_,   State=&gt;_int_out_,   OutputUP=&gt;_bool_out_,   OutputDN=&gt;_bool_out_,   SetpoIntLimitH=&gt;_bool_out_,   SetpoIntLimitL=&gt;_bool_out_,   InputWarningH=&gt;_bool_out_,   InputWarningL=&gt;_bool_out_,   Error=&gt;_bool_out_);</pre>	<p>PID_3Step configure un régulateur PID à fonction d'autooptimisation qui a été optimisé pour des vannes et actionneurs commandés par moteur. Cette instruction fournit deux sorties booléennes.</p> <p>PID_3Step est un régulateur PIDT1 avec anti-emballement (anti-windup) et pondération des actions P et D.</p>

- 1 STEP 7 crée automatiquement l'objet technologique et le DB d'instance lorsque vous insérez l'instruction. Le DB d'instance contient les paramètres de l'objet technologique.
- 2 Dans l'exemple SCL, "PID\_3Step\_1" est le nom du DB d'instance.

Tableau 9- 17 Types de données pour les paramètres

Paramètre et type		Type de données	Description
Setpoint	IN	Real	Consigne du régulateur PID en mode automatique. Valeur par défaut : 0.0
Input	IN	Real	Mesure. Valeur par défaut : 0.0 Vous devez également paramétrer Config.InputPEROn à FALSE.
Input_PER	IN	Word	Mesure analogique (optionnelle). Valeur par défaut : W#16#0 Vous devez également paramétrer Config.InputPEROn à TRUE.
ManualEnable	IN	Bool	Active ou désactive le mode de fonctionnement manuel. Valeur par défaut : FALSE <ul style="list-style-type: none"> <li>En cas de front montant de FALSE à TRUE, le régulateur PID passe en mode manuel (State = 4) et Retain.Mode reste inchangé.</li> <li>En cas de front descendant de TRUE à FALSE, le régulateur PID passe au dernier mode de fonctionnement actif et State = Retain.Mode.</li> </ul>
ManualUP	IN	Bool	En mode manuel, chaque front montant ouvre la vanne de 5 % de la plage d'actionnement totale ou pour la durée du temps d'actionnement minimum du moteur. ManualUP est évalué uniquement si vous n'utilisez pas Output_PER et s'il n'y a pas de signalisation de position. Valeur par défaut : FALSE <ul style="list-style-type: none"> <li>Si Output_PER a la valeur FALSE, l'entrée manuelle active Output_UP pour une durée correspondant à un mouvement de 5% du dispositif.</li> <li>Si Config.ActuatorEndStopOn a la valeur TRUE, Output_UP n'est pas activé si Actuator_H a la valeur TRUE.</li> </ul>
ManualDN	IN	Bool	En mode manuel, chaque front montant ferme la vanne de 5 % de la plage d'actionnement totale ou pour la durée du temps d'actionnement minimum du moteur. ManualDN est évalué uniquement si vous n'utilisez pas Output_PER et s'il n'y a pas de signalisation de position. Valeur par défaut : FALSE <ul style="list-style-type: none"> <li>Si Output_PER a la valeur FALSE, l'entrée manuelle active Output_DN pour une durée correspondant à un mouvement de 5% du dispositif.</li> <li>Si Config.ActuatorEndStopOn a la valeur TRUE, Output_DN n'est pas activé si Actuator_L a la valeur TRUE.</li> </ul>
ManualValue	IN	Real	Valeur de réglage pour le mode manuel. Valeur par défaut : 0.0 En mode manuel, vous indiquez la position absolue de la vanne. ManualValue est évalué uniquement si vous utilisez OutputPer ou si la signalisation de position est disponible. Valeur par défaut : 0.0
Feedback	IN	Real	Signalisation de position de la vanne. Valeur par défaut : 0.0 Pour utiliser Feedback, paramétrez Config.FeedbackPerOn = FALSE.

Paramètre et type		Type de données	Description
Feedback_PER	IN	Word	Signalisation analogique de la position de la vanne. Valeur par défaut : W#16#0 Pour utiliser Feedback_PER, paramétrez Config.FeedbackPerOn = TRUE. Feedback_PER est mis à l'échelle à l'aide des paramètres suivants : <ul style="list-style-type: none"> <li>• Config.FeedbackScaling.LowerPointIn</li> <li>• Config.FeedbackScaling.UpperPointIn</li> <li>• Config.FeedbackScaling.LowerPointOut</li> <li>• Config.FeedbackScaling.UpperPointOut</li> </ul>
Actuator_H	IN	Bool	Si Actuator_H = TRUE, la vanne est à la butée supérieure et n'est plus déplacée dans ce sens. Valeur par défaut : FALSE
Actuator_L	IN	Bool	Si Actuator_L = TRUE, la vanne est à la butée inférieure et n'est plus déplacée dans ce sens. Valeur par défaut : FALSE
Reset	IN	Bool	Redémarre le régulateur PID. Valeur par défaut : FALSE En cas de front montant de FALSE à TRUE : <ul style="list-style-type: none"> <li>• Mode de fonctionnement "inactif"</li> <li>• Mesure = 0</li> <li>• Les valeurs intermédiaires du régulateur sont réinitialisées (les paramètres PID sont conservés).</li> </ul> En cas de front descendant de TRUE à FALSE, passage au dernier mode actif.
ScaledInput	OUT	Real	Mesure mise à l'échelle
ScaledFeedback	OUT	Real	Position de vanne mise à l'échelle
Output_PER	OUT	Word	Valeur de réglage analogique. Si Config.OutputPerOn a la valeur TRUE, le paramètre Output_PER est utilisé.
Output_UP	OUT	Bool	Valeur de réglage TOR pour l'ouverture de la vanne. Valeur par défaut : FALSE Si Config.OutputPerOn a la valeur FALSE, le paramètre Output_UP est utilisé.
Output_DN	OUT	Bool	Valeur de réglage TOR pour la fermeture de la vanne. Valeur par défaut : FALSE Si Config.OutputPerOn a la valeur FALSE, le paramètre Output_DN est utilisé.
SetpointLimitH	OUT	Bool	Limite supérieure de la consigne. Valeur par défaut : FALSE Si SetpointLimitH a la valeur TRUE, la limite supérieure absolue de la consigne est atteinte. Dans la CPU, la consigne est limitée à la limite supérieure absolue configurée de la mesure.
SetpointLimitL	OUT	Bool	Limite inférieure de la consigne. Valeur par défaut : FALSE Si SetpointLimitL a la valeur TRUE, la limite inférieure absolue de la consigne est atteinte. Dans la CPU, la consigne est limitée à la limite inférieure absolue configurée de la mesure.
InputWarningH	OUT	Bool	Si InputWarningH a la valeur TRUE, la mesure a atteint ou dépassé la limite d'alerte supérieure. Valeur par défaut : FALSE
InputWarningL	OUT	Bool	Si InputWarningL a la valeur TRUE, la mesure a atteint ou dépassé la limite d'alerte inférieure. Valeur par défaut : FALSE

Paramètre et type		Type de données	Description
State	OUT	Int	Mode de fonctionnement en cours du régulateur PID. Valeur par défaut : 0 Utilisez Retain.Mode pour changer le mode de fonctionnement : <ul style="list-style-type: none"> <li>• State = 0 : Inactif</li> <li>• State = 1 : Optimisation préalable</li> <li>• State = 2 : Optimisation fine manuelle</li> <li>• State = 3 : Mode automatique</li> <li>• State = 4 : Mode manuel</li> <li>• State = 5 : Accostage de la valeur de réglage de remplacement</li> <li>• State = 6 : Mesure du temps de transition</li> <li>• State = 7 : Accostage de la valeur de réglage de remplacement avec surveillance d'erreur</li> <li>• State = 8 : Surveillance d'erreur</li> </ul>
Error	OUT	Bool	Si Error a la valeur TRUE, au moins un message d'erreur est en attente. Valeur par défaut : FALSE
ErrorBits	OUT	DWord	Le tableau du paramètre (Page 386) ErrorBits de l'instruction PID_3STEP présente les messages d'erreur. Valeur par défaut : DW#16#0000 (pas d'erreur)

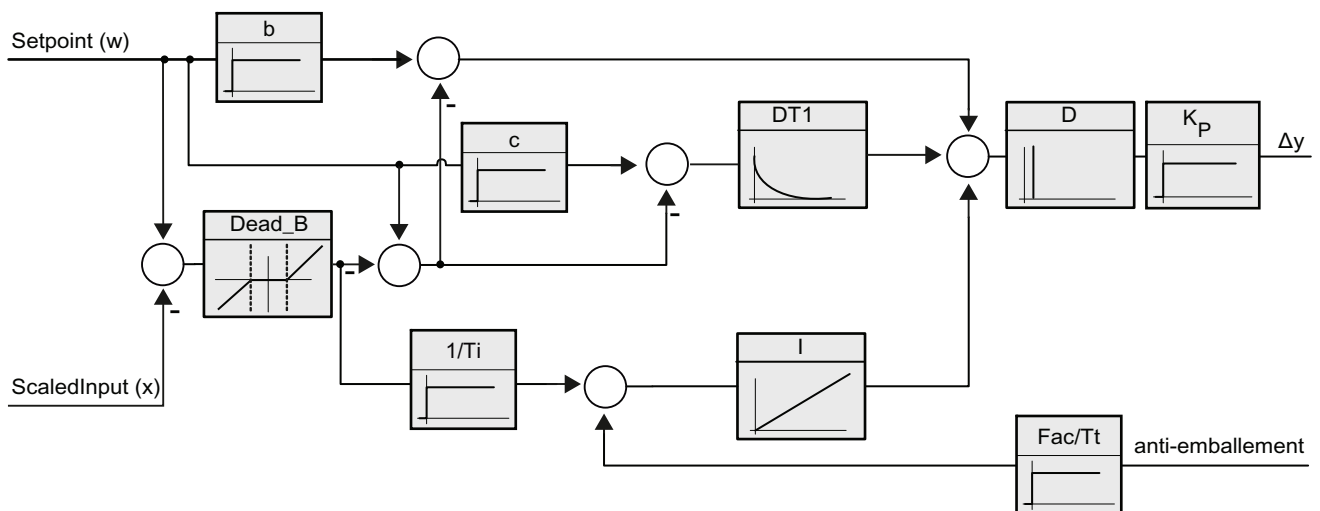


Figure 9-3 Fonctionnement du régulateur PID\_3Step en tant que régulateur PIDT1 avec anti-emballement

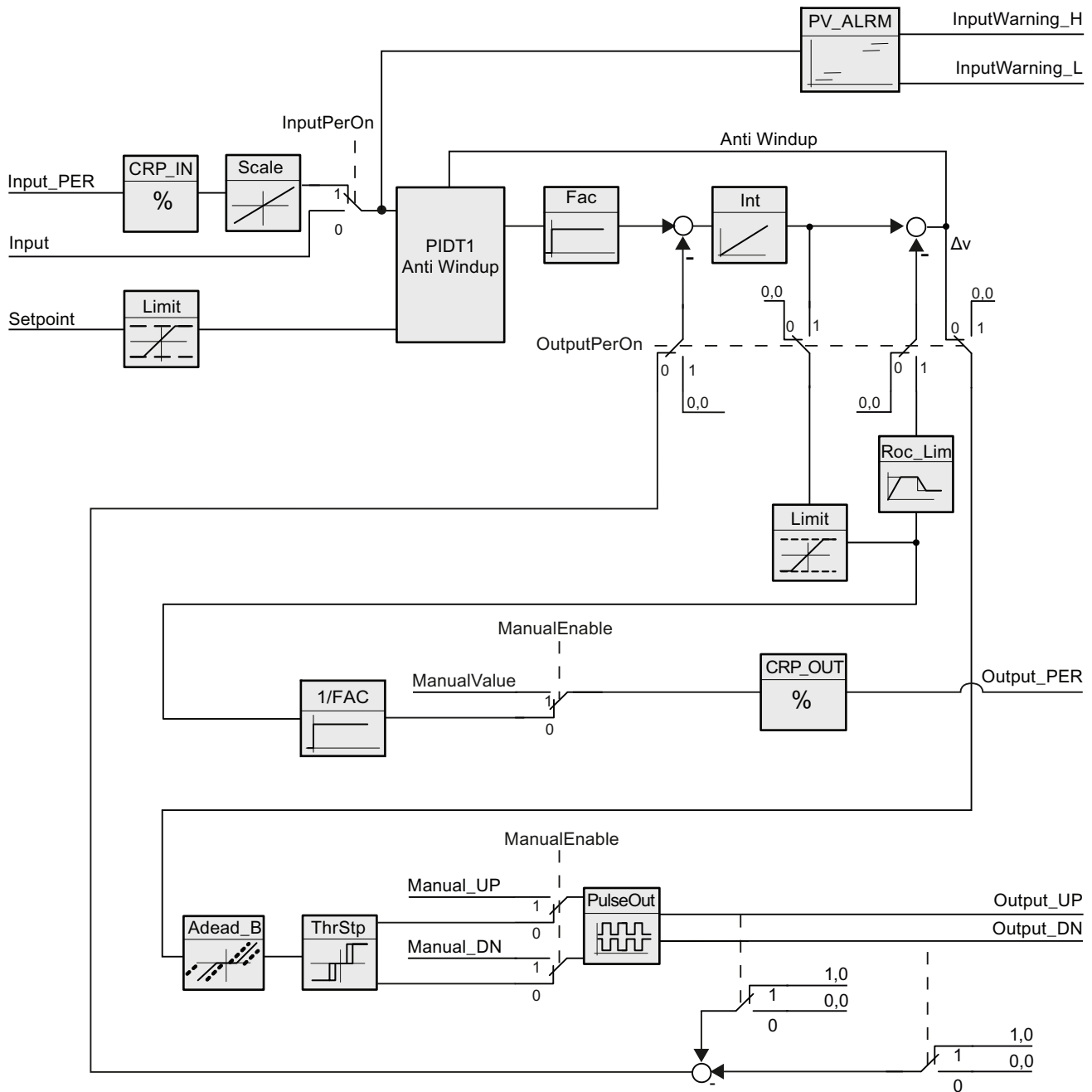


Figure 9-4 Fonctionnement du régulateur PID\_3Step sans signalisation de position



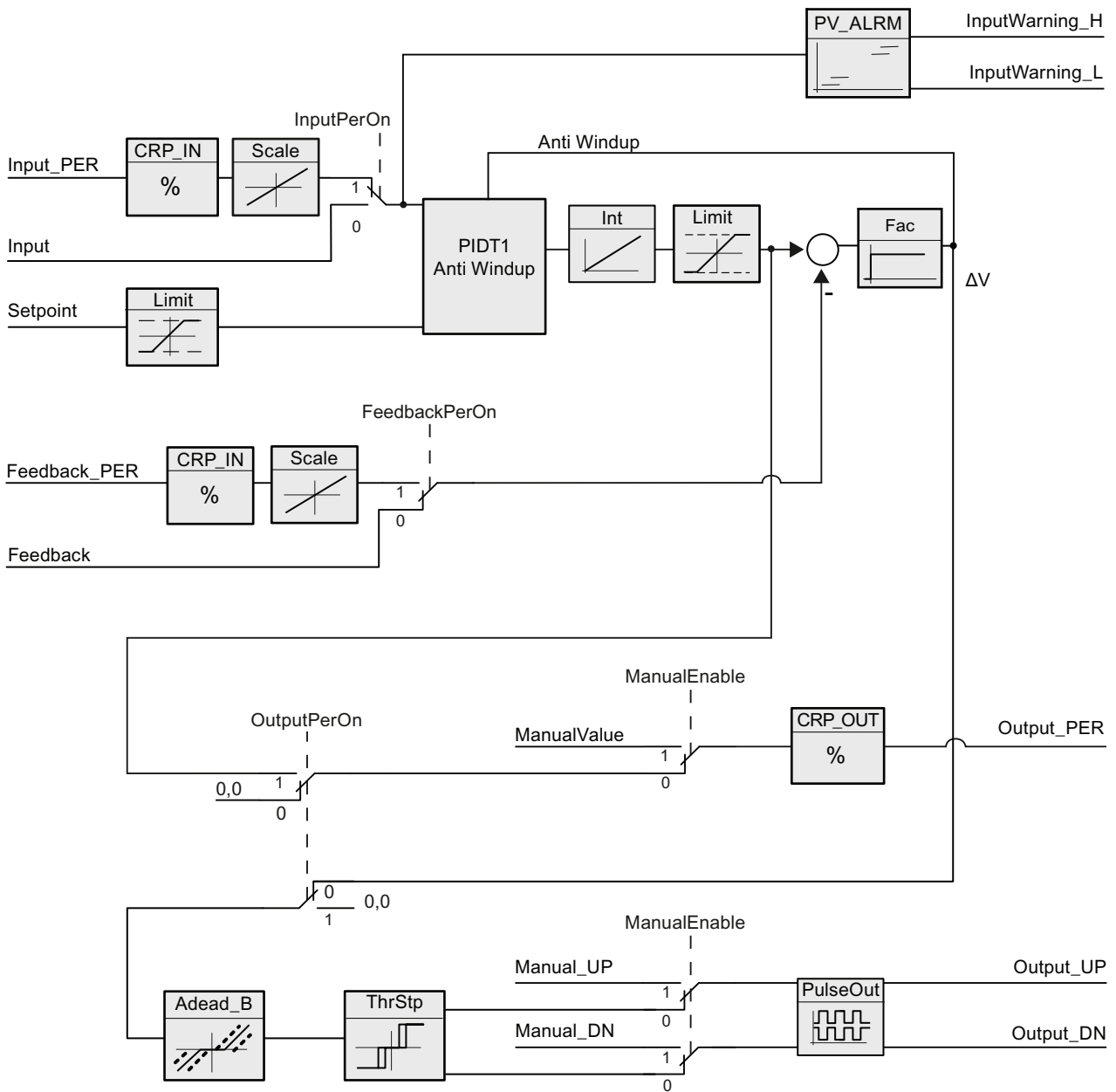


Figure 9-5 Fonctionnement du régulateur PID\_3Step avec signalisation de position activée

### 9.2.5 Paramètre ErrorBits de l'instruction PID\_3STEP

Si plusieurs erreurs sont en attente, les valeurs des codes d'erreur sont données au moyen d'une addition binaire. Ainsi, l'affichage du code d'erreur 0003 signifie que les erreurs 0001 et 0002 sont également en attente.

Tableau 9- 18 Paramètre ErrorBits de l'instruction PID\_3STEP

ErrorBit (DW#16#...)	Description
0000	Pas d'erreur
0001	Le paramètre "Input" se situe en dehors des limites de la mesure : Input > Config.InputUpperLimit ou Input < Config.InputLowerLimit Si ActivateRecoverMode a la valeur TRUE et que ErrorBehaviour est égal à 1, l'actionneur se déplace vers la valeur de réglage de remplacement. Si ActivateRecoverMode a la valeur TRUE et que ErrorBehaviour est égal à 0, l'actionneur s'arrête à sa position en cours. Si ActivateRecoverMode a la valeur FALSE, l'actionneur s'arrête à sa position en cours. PID_3STEP V1.1 : Vous pouvez déplacer l'actionneur en mode manuel. PID_3STEP V1.0 : Le mode manuel n'est pas possible dans cet état. Vous ne pourrez redémarrer l'actionneur qu'après élimination de l'erreur.
0002	Valeur invalide dans le paramètre "Input_PER". Vérifiez s'il y a une erreur en attente au niveau de l'entrée analogique. Si le mode automatique était actif avant l'apparition de l'erreur, que ActivateRecoverMode a la valeur TRUE et que l'erreur n'est plus en attente, PID_3STEP repasse en mode automatique.
0004	Erreur pendant l'optimisation fine. L'oscillation de la mesure n'a pas pu être conservée.
0008	Erreur au démarrage de l'optimisation préalable. La mesure est trop proche de la consigne. Démarrez l'optimisation fine.
0010	La consigne ne peut pas être modifiée pendant l'optimisation fine.
0020	Impossible d'exécuter une optimisation préalable en mode automatique ou pendant l'optimisation fine.
0040	Erreur pendant l'optimisation fine. La consigne est trop proche des limites de la consigne.
0080	Erreur pendant l'optimisation préalable. Configuration incorrecte des limites de la valeur de réglage. Vérifiez que les limites de la valeur de réglage sont configurées correctement et correspondent au sens dans lequel se fait la commande.
0100	Une erreur pendant l'optimisation fine a entraîné des paramètres invalides.
0200	Valeur invalide dans le paramètre "Input" : Le format numérique de la valeur est invalide. Si le mode automatique était actif avant l'apparition de l'erreur, que ActivateRecoverMode a la valeur TRUE et que l'erreur n'est plus en attente, PID_3STEP repasse en mode automatique.
0400	Le calcul de la valeur de réglage a échoué. Vérifiez les paramètres PID.

ErrorBit (DW#16#...)	Description
0800	<p>Erreur de période d'échantillonnage : PID_3STEP n'est pas appelé dans la période d'échantillonnage de l'OB d'alarme cyclique.</p> <p>Si le mode automatique était actif avant l'apparition de l'erreur, que ActivateRecoverMode a la valeur TRUE et que l'erreur n'est plus en attente, PID_3STEP repasse en mode automatique.</p>
1000	<p>Valeur invalide dans le paramètre "Setpoint" : Le format numérique de la valeur est invalide.</p> <p>Si le mode automatique était actif avant l'apparition de l'erreur, que ActivateRecoverMode a la valeur TRUE et que l'erreur n'est plus en attente, PID_3STEP repasse en mode automatique.</p>
2000	<p>Valeur invalide dans le paramètre Feedback_PER.</p> <p>Vérifiez s'il y a une erreur en attente au niveau de l'entrée analogique.</p> <p>L'actionneur ne peut pas être déplacé vers la valeur de réglage de remplacement et ne quitte pas la position en cours. Le mode manuel n'est pas possible dans cet état. Vous devez désactiver la signalisation de position (Config. FeedbackOn = FALSE) pour que l'actionneur quitte cet état.</p> <p>Si le mode automatique était actif avant l'apparition de l'erreur, que ActivateRecoverMode a la valeur TRUE et que l'erreur n'est plus en attente, PID_3STEP repasse en mode automatique.</p>
4000	<p>Valeur invalide dans le paramètre Feedback. Le format numérique de la valeur est invalide.</p> <p>L'actionneur ne peut pas être déplacé vers la valeur de réglage de remplacement et ne quitte pas la position en cours. Le mode manuel n'est pas possible dans cet état. Vous devez désactiver la signalisation de position (Config. FeedbackOn = FALSE) pour que l'actionneur quitte cet état.</p> <p>Si le mode automatique était actif avant l'apparition de l'erreur, que ActivateRecoverMode a la valeur TRUE et que l'erreur n'est plus en attente, PID_3STEP repasse en mode automatique.</p>
8000	<p>Erreur dans la signalisation de position numérique. Actuator_H = TRUE et Actuator_L = TRUE.</p> <p>L'actionneur ne peut pas être déplacé vers la valeur de réglage de remplacement et ne quitte pas la position en cours. Le mode manuel n'est pas possible dans cet état.</p> <p>Vous devez désactiver "Butée signale actionneur" (Config.ActuatorEndStopOn = FALSE) pour que l'actionneur quitte cet état.</p> <p>Si le mode automatique était actif avant l'apparition de l'erreur, que ActivateRecoverMode a la valeur TRUE et que l'erreur n'est plus en attente, PID_3STEP repasse en mode automatique.</p>

### 9.2.6 Configuration du régulateur PID

Les paramètres de l'objet technologique déterminent le fonctionnement du régulateur PID. Utilisez l'icône pour ouvrir l'éditeur de configuration.

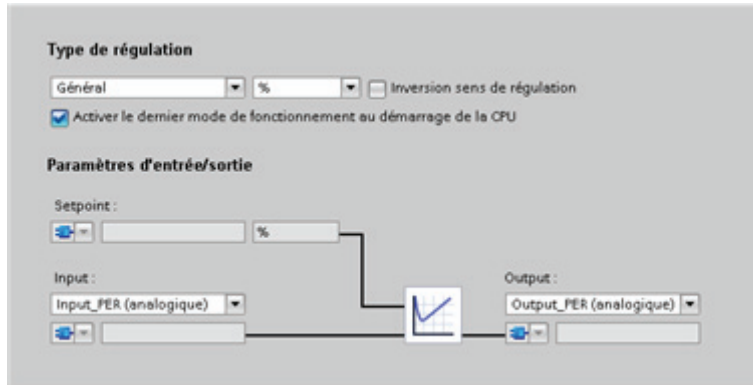


Figure 9-6 Editeur de configuration pour PID\_Compact (paramètres de base)

Tableau 9- 19Exemple de paramètres de configuration pour l'instruction PID\_Compact

Paramètres		Description
Paramètres de base	Type de régulation	Sélectionne les unités physiques.
	Inversion du sens de régulation	Permet la sélection d'une boucle PID à action inverse. <ul style="list-style-type: none"> <li>• Si la case n'est pas cochée, la boucle PID est à action directe et la valeur de réglage de la boucle PID augmente si la mesure est inférieure à la consigne.</li> <li>• Si cette case est cochée, la valeur de réglage de la boucle PID augmente si la mesure est supérieure à la consigne.</li> </ul>
	Activer le dernier mode après redémarrage de la CPU	Redémarre la boucle PID après sa réinitialisation ou si une limite de mesure a été dépassée et est revenue dans la plage valide.
	Mesure	Sélectionne soit le paramètre Input, soit le paramètre Input_PER (analogique) pour la mesure. Input_PER peut provenir directement d'un module d'entrées analogiques.
	Valeur de réglage	Sélectionne soit le paramètre Output, soit le paramètre Output_PER (analogique) pour la valeur de réglage. Output_PER peut aller directement à un module de sorties analogiques.
Mesure	Met à l'échelle la plage ainsi que les limites pour la mesure. Si la mesure devient inférieure à la limite inférieure ou supérieure à la limite supérieure, la boucle PID passe en mode inactif et met la valeur de réglage à 0. Pour utiliser Input_PER, vous <b>devez</b> mettre à l'échelle la mesure analogique.	

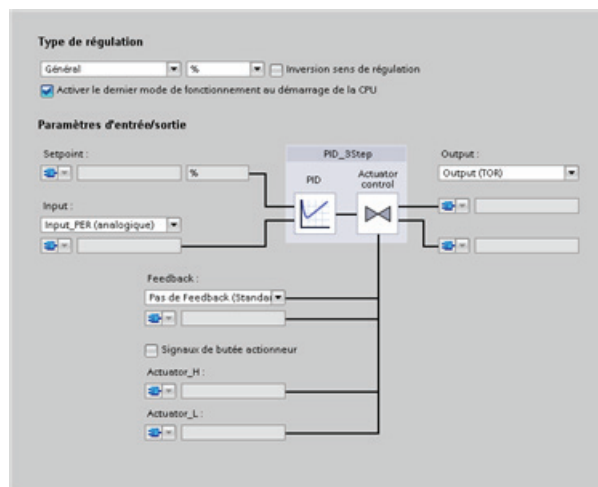


Figure 9-7 Editeur de configuration pour PID\_3Step (paramètres de base)

Tableau 9- 20 Exemple de paramètres de configuration pour l'instruction PID\_3Step

Paramètres		Description
Paramètres de base	Type de régulation	Sélectionne les unités physiques.
	Inversion du sens de régulation	Permet la sélection d'une boucle PID à action inverse. <ul style="list-style-type: none"> <li>• Si la case n'est pas cochée, la boucle PID est à action directe et la valeur de réglage de la boucle PID augmente si la mesure est inférieure à la consigne.</li> <li>• Si cette case est cochée, la valeur de réglage de la boucle PID augmente si la mesure est supérieure à la consigne.</li> </ul>
	Activer le dernier mode après redémarrage de la CPU	Redémarre la boucle PID après sa réinitialisation ou si une limite de mesure a été dépassée et est revenue dans la plage valide.
	Mesure	Sélectionne soit le paramètre Input, soit le paramètre Input_PER (analogique) pour la mesure. Input_PER peut provenir directement d'un module d'entrées analogiques.
	Valeur de réglage	Sélectionne soit l'utilisation de sorties TOR (Output_UP et Output_DN), soit l'utilisation d'une sortie analogique (Output_PER) pour la valeur de réglage.
	Rétroaction	Sélectionne le type d'état renvoyé par la boucle PID : <ul style="list-style-type: none"> <li>• Pas de rétroaction (valeur par défaut)</li> <li>• Rétroaction</li> <li>• Feedback_PER</li> </ul>
Mesure	Met à l'échelle la plage ainsi que les limites pour la mesure. Si la mesure devient inférieure à la limite inférieure ou supérieure à la limite supérieure, la boucle PID passe en mode inactif et met la valeur de réglage à 0. Pour utiliser Input_PER, vous <b>devez</b> mettre à l'échelle la mesure analogique.	
Actionneur	Temps de transition du moteur	Définit le temps nécessaire pour passer de l'état ouvert à l'état fermé de la vanne (vous trouverez cette valeur sur la fiche technique ou la plaque signalétique de la vanne).
	Temps d'activation minimum	Définit le temps de mouvement minimum de la vanne (vous trouverez cette valeur sur la fiche technique ou la plaque signalétique de la vanne).

Paramètres		Description
	Temps de désactivation minimum	Définit le temps de pause minimum de la vanne (vous trouverez cette valeur sur la fiche technique ou la plaque signalétique de la vanne).
	Comportement d'erreur	Définit le comportement de la vanne lorsqu'une erreur est détectée ou que la boucle PID est réinitialisée. Si vous optez pour l'utilisation d'une position de substitution, entrez la "Position de sécurité". Pour une rétroaction ou une sortie analogique, sélectionnez une valeur entre la limite supérieure et la limite inférieure pour la valeur de réglage. Pour des sorties TOR, vous pouvez choisir 0% (désactivé) ou 100% (activé).
	Mise à l'échelle de la rétroaction de position <sup>1</sup>	<ul style="list-style-type: none"> <li>• "Butée supérieure" et "Butée inférieure" définissent respectivement la position positive maximale (entièrement ouvert) et la position négative maximale (entièrement fermé). "Butée supérieure" doit être supérieure à "Butée inférieure".</li> <li>• "Limite supérieure de la mesure" et "Limite inférieure de la mesure" définissent les positions supérieure et inférieure de la vanne pendant le réglage et le mode automatique.</li> <li>• "FeedbackPER" ("Low" et "High") définit la rétroaction analogique de la position de la vanne. "FeedbackPER High" doit être supérieur à "FeedbackPER Low".</li> </ul>

<sup>1</sup> "Mise à l'échelle de la rétroaction de position" ne peut être modifié que si vous avez activé "Rétroaction" dans les paramètres de base.

### 9.2.7 Mise en service du régulateur PID

Utilisez l'éditeur de mise en service pour configurer le régulateur PID afin qu'une optimisation préalable soit effectuée à la mise en route et en fonctionnement. Pour ouvrir l'éditeur de mise en service, cliquez sur l'icône correspondante dans l'instruction ou dans le navigateur du projet.



Tableau 9- 21 Exemple d'écran de configuration (PID\_3Step)

	<ul style="list-style-type: none"><li>• Mesure : Pour afficher la consigne, la mesure et la valeur de réglage dans un affichage de courbes en temps réel, entrez la période d'échantillonnage et cliquez sur le bouton "Démarrer".</li><li>• Optimisation : Pour optimiser la boucle PID, sélectionnez soit "Optimisation préalable" soit "Optimisation fine" (manuelle) et cliquez sur le bouton "Démarrer". Le régulateur PID exécute plusieurs phases pour calculer les temps de réponse et d'actualisation du système. Les paramètres optimisés appropriés sont calculés à partir de ces valeurs.</li></ul> <p>Une fois le processus d'optimisation achevé, vous pouvez stocker les nouveaux paramètres en cliquant sur le bouton "Charger les paramètres PID dans le projet" dans la partie "Paramètres PID" de l'éditeur de mise en service.</p> <p>Si une erreur se produit pendant l'optimisation, la valeur de réglage de la boucle PID est mise à 0. Le régulateur PID passe alors en mode "inactif". L'état signale l'erreur.</p>
--	----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

## 9.3 Commande de mouvement

La CPU fournit une fonctionnalité de commande de mouvement (Motion Control) permettant de faire fonctionner des moteurs à pas et des servomoteurs à interface d'impulsion. La fonctionnalité de commande de mouvement se charge de la commande et de la surveillance des entraînements.

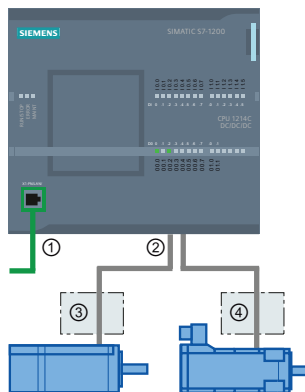
- L'objet technologique "Axe" configure les données mécaniques de l'entraînement, l'interface de l'entraînement, les paramètres dynamiques et d'autres propriétés de l'entraînement.
- Vous configurez les sorties d'impulsions et de sens de la CPU pour commander l'entraînement.
- Votre programme utilisateur utilise les instructions de commande de mouvement pour commander l'axe et déclencher des tâches de déplacement.
- Utilisez l'interface PROFINET pour établir la liaison en ligne entre la CPU et la console de programmation. En plus des fonctions en ligne de la CPU, des fonctions supplémentaires de mise en service et de diagnostic sont disponibles pour la commande de mouvement.

---

### Remarque

Les modifications que vous apportez à la configuration de commande de mouvement et que vous chargez à l'état MARCHE ne prennent effet qu'après passage de la CPU de l'état ARRET à l'état MARCHE.

---



- ① PROFINET
- ② Sorties d'impulsions et de sens
- ③ Partie puissance du moteur pas à pas
- ④ Partie puissance du servomoteur

Les variantes DC/DC/DC de la CPU S7-1200 ont des sorties intégrées pour la commande directe d'entraînements. Les variantes à relais de la CPU ont besoin du Signal Board à sorties DC pour commander des entraînements.

Un Signal Board (SB) augmente les E/S intégrées en ajoutant quelques E/S supplémentaires. Un SB à 2 sorties TOR peut être utilisé comme sorties d'impulsions et de sens pour commander un moteur. Un SB à 4 sorties TOR peut être utilisé comme sorties d'impulsions et de sens pour commander deux moteurs. Les sorties relais intégrées ne peuvent pas être utilisées comme sorties d'impulsions pour commander des moteurs.

**Remarque**

**Les sorties de trains d'impulsions ne peuvent pas être utilisées par d'autres instructions dans le programme utilisateur**

Lorsque vous configurez les sorties de la CPU ou du Signal Board en tant que générateurs d'impulsions (pour les instructions PWM ou de commande de mouvement), les adresses des sorties correspondantes (Q0.0 à Q0.3, Q4.0 à Q4.3) sont supprimées de la mémoire Q et ne peuvent pas être utilisées à d'autres fins dans le programme utilisateur. Si votre programme utilisateur écrit une valeur dans une sortie utilisée comme générateur d'impulsions, la CPU n'écrit pas cette valeur dans la sortie physique.

Tableau 9- 22 Nombre maximum d'entraînements pouvant être commandés

Type de CPU		Pas de SB installé	Avec un SB (2 sorties DC)	Avec un SB (4 sorties DC)
CPU 1211C	DC/DC/DC	2	2	2
	AC/DC/RLY	0	1	2
	DC/DC/RLY	0	1	2
CPU 1212C	DC/DC/DC	2	2	2
	AC/DC/RLY	0	1	2
	DC/DC/RLY	0	1	2
CPU 1214C	DC/DC/DC	2	2	2
	AC/DC/RLY	0	1	2
	DC/DC/RLY	0	1	2
CPU 1215C	DC/DC/DC	4	4	4
	AC/DC/RLY	0	1	2



Type de CPU	Pas de SB installé	Avec un SB (2 sorties DC)	Avec un SB (4 sorties DC)
DC/DC/RLY	0	1	2

Tableau 9- 23 Fréquences limites des sorties d'impulsions

Sortie d'impulsions	Fréquence
Intégrée	2 PTO : $2 \text{ Hz} \leq f \leq 100 \text{ kHz}$
	2 PTO : $2 \text{ Hz} \leq f \leq 20 \text{ kHz}$
SB standard	$2 \text{ Hz} \leq f \leq 20 \text{ kHz}$
SB rapides (200 kHz)	Instructions MC V2 : $2 \text{ Hz} \leq f \leq 200 \text{ kHz}$
	Instructions MC V1 : $2 \text{ Hz} \leq f \leq 100 \text{ kHz}$ <sup>1</sup>

<sup>1</sup> Les instructions MC V1 acceptent une fréquence maximale de 100 kHz.

<b>IMPORTANT</b>
La fréquence d'impulsion maximale des générateurs d'impulsions est de 100 kHz pour les sorties TOR de la CPU, de 20 kHz pour les sorties TOR du SB standard et de 200 kHz pour les sorties TOR des SB rapides (ou de 100 kHz pour les instructions MC V1).

### Configuration d'un générateur d'impulsions

1. Ajoutez un objet technologique :

### 9.3 Commande de mouvement

- Dans l'arborescence du projet, affichez le détail du nœud "Objets technologiques" et sélectionnez "Ajouter nouvel objet".
- Sélectionnez l'icône "Axe" (renommez-la si nécessaire) et cliquez sur "OK" pour ouvrir l'éditeur de configuration pour l'objet axe.
- Affichez les propriétés "Sélection de PTO pour la commande d'axe" sous les "Paramètres de base" et sélectionnez l'impulsion désirée. Notez les deux sorties Q affectées à l'impulsion et au sens.

---

#### Remarque

Si PTO n'a pas été configuré auparavant dans les propriétés CPU, il est configuré pour utiliser l'une des sorties intégrées.

Si vous utilisez une sortie de Signal Board, sélectionnez le bouton "Configuration d'appareil" pour afficher les propriétés CPU. Dans les "Options d'impulsions" sous "Affectation de paramètres", configurez la source de la sortie sur une sortie de Signal Board. "Pulse\_1" et "Pulse\_3" sont les seules sorties d'impulsions disponibles sur le Signal Board.

---

- Configurez les autres paramètres de base et avancés.
2. Programmez votre application : Insérez l'instruction MC\_Power dans un bloc de code.
    - Pour l'entrée Axis, sélectionnez l'objet technologique axe que vous avez créé et configuré.
    - Lorsque l'entrée Enable à la valeur VRAI, les autres instructions de mouvement sont autorisées à fonctionner.
    - Lorsque l'entrée Enable a la valeur FAUX, les autres instructions de mouvement sont annulées.

---

#### Remarque

N'utilisez qu'une instruction MC\_Power par axe.

---

3. Insérez les autres instructions de mouvement pour produire le mouvement requis.

---

#### Remarque

Configuration d'un générateur d'impulsions sur des sorties de Signal Board : Sélectionnez les propriétés "Générateurs d'impulsions (PTO/PWM)" pour une CPU (dans la configuration d'appareil) et activez un générateur d'impulsions. Deux générateurs d'impulsions sont disponibles pour chaque CPU S7-1200 V1.0, V2.0, V2.1 et V2.2. Les CPU S7-1200 V3.0 disposent de quatre générateurs d'impulsions. Sous "Options d'impulsions" dans cette même zone de configuration, sélectionnez "Générateur d'impulsions sous forme de : PTO".

---

---

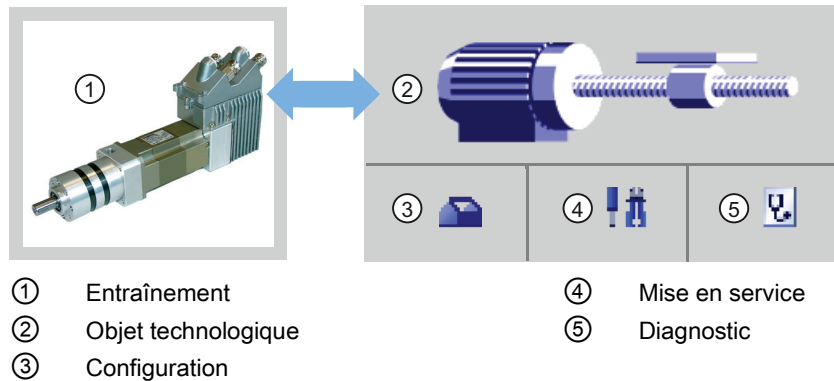
### Remarque

La CPU calcule les tâches de mouvement en "tranches" ou segments de 10 ms. Pendant l'exécution d'une tranche, la tranche suivante est mise en file d'attente pour être exécutée. Si vous interrompez la tâche de mouvement en cours sur un axe (par l'exécution d'une autre tâche de mouvement pour cet axe), l'exécution de cette nouvelle tâche de mouvement pourra ne commencer qu'après 20 ms au maximum (reste de la tranche en cours + tranche en file d'attente).

---

## 9.3.1 Configuration de l'axe

STEP 7 fournit les outils de configuration, les outils de mise en service et les outils de diagnostic pour l'objet technologique "Axe".



---

### Remarque

PTO exige la fonctionnalité interne d'un compteur rapide (HSC). Cela signifie que le compteur rapide correspondant ne peut pas être utilisé ailleurs.

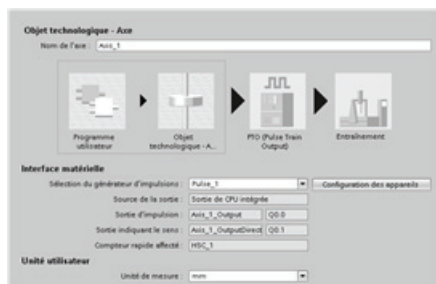
L'affectation entre PTO et HSC est fixe. Lorsque PTO1 est activé, il est associé à HSC1. Lorsque PTO2 est activé, il est associé à HSC2. Cela n'est vrai que pour les CPU S7-1200 V1.0, V2.0, V2.1 et V2.2. Les CPU S7-1200 V3.0 ne présentent pas cette restriction.

Vous ne pouvez pas surveiller la valeur en cours (par exemple, dans ID 1000) lorsque des impulsions surviennent.

---

Tableau 9- 24 Outils STEP 7 pour la commande de mouvement

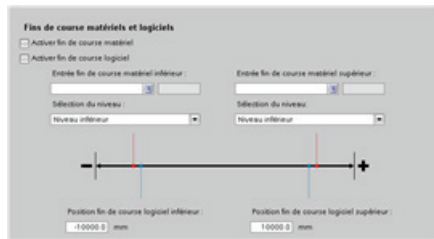
Outil	Description
Configuration	<p>Configure les propriétés suivantes de l'objet technologique "Axe" :</p> <ul style="list-style-type: none"> <li>• Sélection de la sortie PTO à utiliser et configuration de l'interface d'entraînement</li> <li>• Propriétés de la mécanique et du rapport de transmission de l'entraînement (ou machine ou système)</li> <li>• Propriétés pour les limites de positionnement, la dynamique et le référencement</li> </ul> <p>Sauvegardez la configuration dans le bloc de données de l'objet technologique.</p>
Mise en service	<p>Teste la fonction de votre axe sans avoir à créer un programme utilisateur. Lorsque l'outil est lancé, le panneau de commande s'affiche. Les commandes suivantes sont disponibles sur le panneau de commande :</p> <ul style="list-style-type: none"> <li>• Libérer et bloquer l'axe</li> <li>• Déplacer l'axe en mode Manuel à vue</li> <li>• Positionner l'axe en termes absolus et relatifs</li> <li>• Référencer l'axe</li> <li>• Acquitter les erreurs</li> </ul> <p>La vitesse et l'accélération/la décélération peuvent être spécifiées pour les commandes de mouvement. Le panneau de commande affiche également l'état en cours de l'axe.</p>
Diagnostic	<p>Surveille l'état en cours et les informations d'erreur concernant l'axe et l'entraînement.</p>



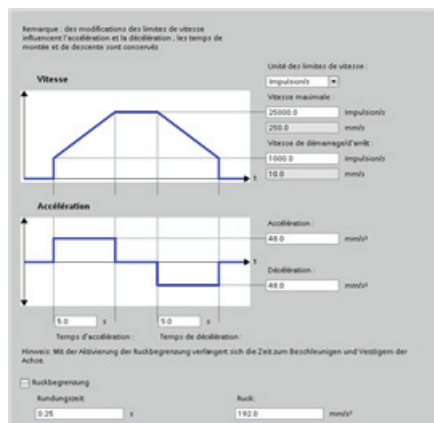
Une fois que vous avez créé l'objet technologique pour l'axe, vous configurez l'axe en définissant les paramètres de base, tels que la sortie PTO et la configuration de l'interface d'entraînement. Vous configurez également d'autres propriétés de l'axe, telles que les limites de positionnement, la dynamique et le référencement.

**IMPORTANT**

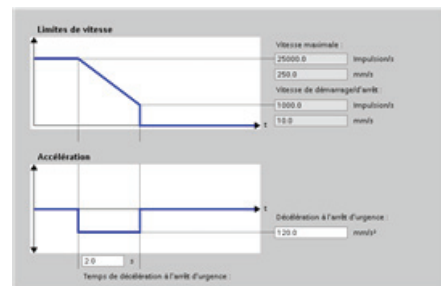
Vous devrez peut-être adapter les valeurs des paramètres d'entrée des instructions de commande de mouvement à la nouvelle unité de dimension dans le programme utilisateur.



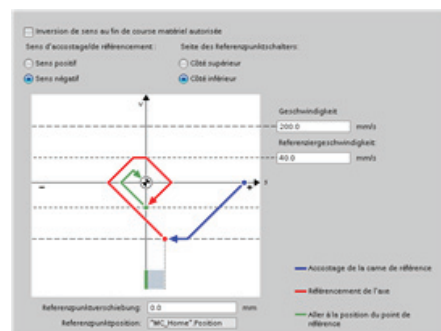
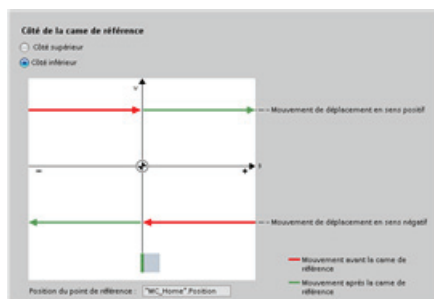
Configurez les propriétés pour les signaux de l'entraînement, la mécanique de l'entraînement et la surveillance de position (fins de course matériels et logiciels).




Vous configurez la dynamique du mouvement et le comportement de la commande d'arrêt d'urgence.



Vous configurez en outre le comportement de référencement (passif et actif).



Utilisez le panneau de commande "Mise en service" pour tester la fonctionnalité indépendamment de votre programme utilisateur.

 Cliquez sur l'icône "Démarrer" pour mettre l'axe en service.

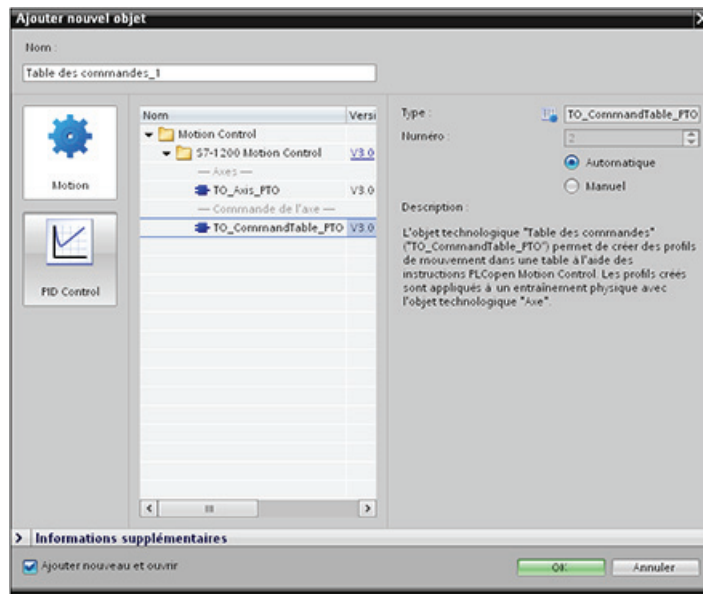
Le panneau de commande montre l'état en cours de l'axe. Vous pouvez non seulement libérer et bloquer l'axe, mais également tester le positionnement de l'axe (en termes absolus et relatifs) et vous pouvez indiquer la vitesse, l'accélération et la décélération. Vous pouvez également tester les tâches de référencement et le mode Manuel à vue. Le panneau de commande vous permet en outre d'acquiescer les erreurs.

### 9.3.2 Configuration de la table de commande TO\_CommandTable\_PTO

Vous pouvez configurer une instruction CommandTable à l'aide des objets technologiques.

#### Ajout d'un objet technologique

1. Dans l'arborescence du projet, affichez le détail du nœud "Objets technologiques" et sélectionnez "Ajouter nouvel objet".
2. Sélectionnez l'icône "CommandTable" (renommez-la si nécessaire) et cliquez sur "OK" pour ouvrir l'éditeur de configuration pour l'objet CommandTable.



#### Planification des étapes pour votre application

Vous pouvez créer la séquence de mouvement désirée dans la fenêtre de configuration "Table de commande" et vérifier le résultat dans la vue graphique du graphique de tendance.

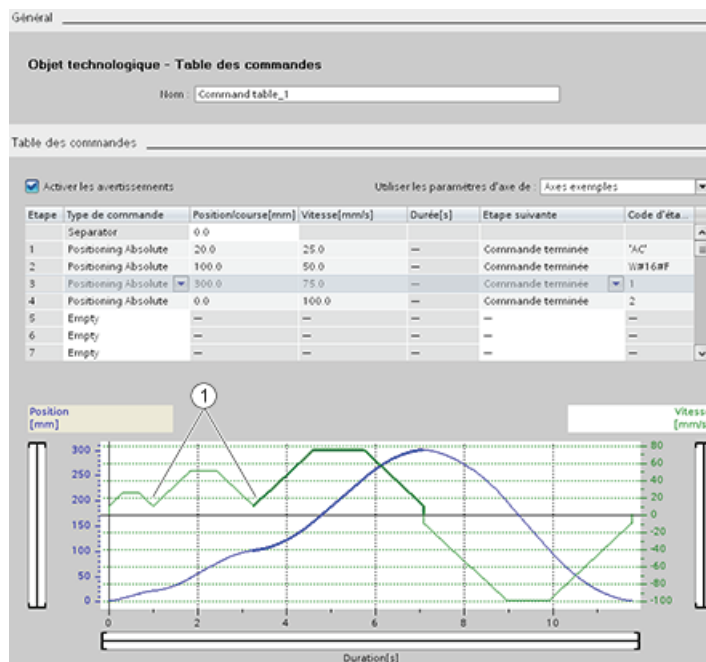
Vous pouvez sélectionner les types de commandes qui doivent être utilisés pour le traitement de la table de commande. Il est possible d'entrer jusqu'à 32 étapes. Les commandes sont traitées l'une après l'autre, produisant aisément un profil de mouvement complexe.

Tableau 9- 25 Types de commandes MC\_CommandTable

Type de commande	Description
Empty	Marque de réservation pour toute commande à ajouter. L'entrée Empty n'est pas prise en compte lors du traitement de la table de commande.
Halt	Pause de l'axe. Remarque : cette commande n'est exécutée qu'après une commande de consigne de vitesse.
Positioning Relative	Positionnement de l'axe en fonction de la distance. La commande déplace l'axe de la distance indiquée à la vitesse donnée.

Type de commande	Description
Positioning Absolute	Positionnement de l'axe en fonction de l'emplacement. La commande déplace l'axe à l'emplacement indiqué, en utilisant la vitesse donnée.
Velocity setpoint	Déplacement de l'axe à la vitesse donnée.
Wait	Attente de l'expiration de la durée indiquée. "Wait" n'interrompt pas un accostage actif.
Separator	Ajout d'une ligne séparatrice au-dessus de la ligne sélectionnée. La ligne séparatrice permet de définir plus d'un profil dans une seule table de commande.

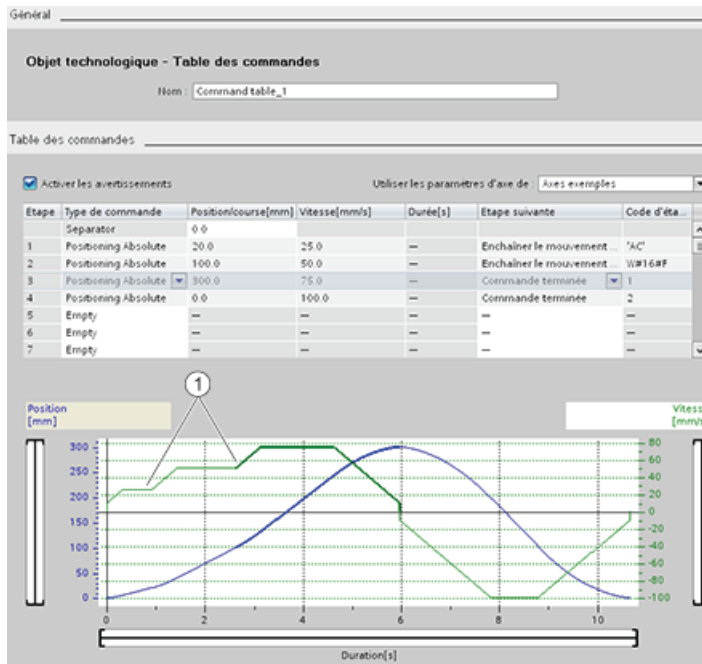
Dans la figure ci-dessous, "Commande achevée" est utilisé comme transition à l'étape suivante. Ce type de transition permet à votre appareil de décélérer jusqu'à la vitesse de démarrage/d'arrêt, puis de réaccélérer au début de l'étape suivante.



① L'axe décélère jusqu'à la vitesse de démarrage/d'arrêt entre les étapes.

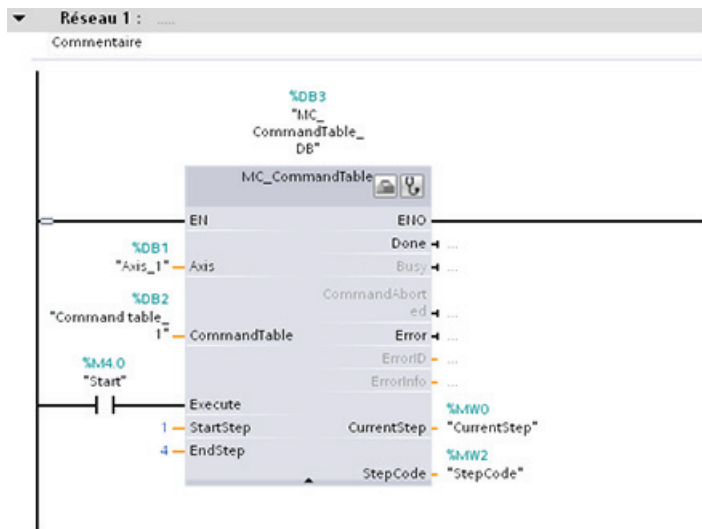
Dans la figure ci-dessous, "Lissage mouvement" est utilisé comme transition à l'étape suivante. Ce type de transition permet à l'appareil de conserver sa vitesse lors de l'entrée dans l'étape suivante, lissant ainsi la transition d'une étape à la suivante pour l'appareil. Le lissage peut réduire le temps total nécessaire à l'exécution d'un profil complet. Sans lissage, l'exécution de l'exemple prend sept secondes. Avec le lissage, le temps d'exécution est réduit d'une seconde pour atteindre un total de six secondes.

9.3 Commande de mouvement



① L'axe continue à se déplacer et accélère ou décélère selon la vitesse de l'étape suivante, permettant ainsi un gain de temps et une réduction de l'usure mécanique.

Le fonctionnement de votre table CommandTable est géré par une instruction MC\_CommandTable, comme illustré ci-dessous :





### 9.3.3 Instructions de commande de mouvement (Motion Control)

#### Remarque

La CPU calcule les tâches de mouvement en "tranches" ou segments de 10 ms. Pendant l'exécution d'une tranche, la tranche suivante est mise en file d'attente pour être exécutée. Si vous interrompez la tâche de mouvement en cours sur un axe (par l'exécution d'une autre tâche de mouvement pour cet axe), l'exécution de cette nouvelle tâche de mouvement pourra ne commencer qu'après 20 ms au maximum (reste de la tranche en cours + tranche en file d'attente).

#### 9.3.3.1 Instruction MC\_Power

<b>IMPORTANT</b>
Si l'axe est désactivé à la suite d'une erreur, il sera automatiquement réactivé une fois l'erreur éliminée et acquittée. Cela nécessite que le paramètre d'entrée Enable ait conservé la valeur VRAI pendant ce processus.

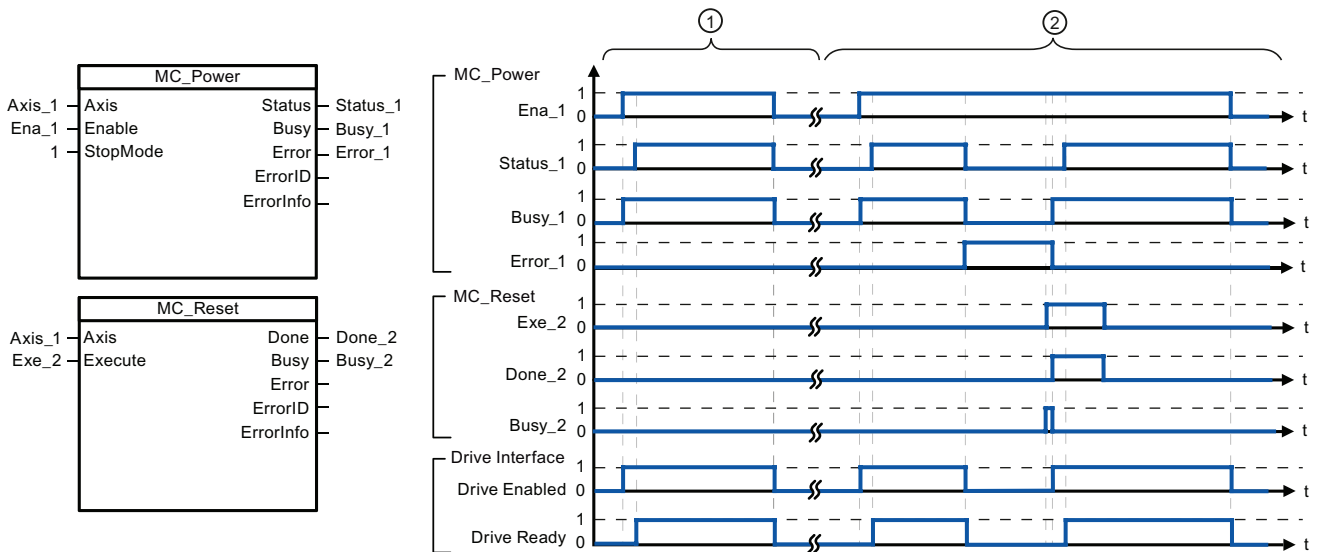
Tableau 9- 26 Instruction MC\_Power

CONT/LOG	SCL	Description
	<pre>"MC_Power_DB" (   Axis:= _multi_fb_in_,   Enable:= _bool_in_,   StopMode:= _int_in_,   Status=&gt; _bool_out_,   Busy=&gt; _bool_out_,   Error=&gt; _bool_out_,   ErrorID=&gt; _word_out_,   ErrorInfo=&gt; _word_out_);</pre>	<p>L'instruction de commande de mouvement MC_Power valide ou inhibe un axe. Les conditions suivantes doivent être remplies pour que vous puissiez libérer ou bloquer l'axe :</p> <ul style="list-style-type: none"> <li>• L'objet technologique a été configuré correctement.</li> <li>• Il n'y a pas d'erreur de libération/blocage en attente.</li> </ul> <p>L'exécution de MC_Power ne peut pas être annulée par une tâche de commande de mouvement. Le blocage de l'axe (paramètre d'entrée Enable = FAUX) annule toutes les tâches de commande de mouvement pour l'objet technologique associé.</p>

- 1 STEP 7 crée automatiquement le DB lorsque vous insérez l'instruction.
- 2 Dans l'exemple SCL, "MC\_Power\_DB" est le nom du DB d'instance.

Tableau 9- 27 Paramètres pour l'instruction MC\_Power

Paramètre et type		Type de données	Description
Axis	IN	TO_Axis_1	Objet technologique Axe
Enable	IN	Bool	<ul style="list-style-type: none"> <li>FALSE (valeur par défaut) : Toutes les tâches actives sont annulées conformément au mode "StopMode" paramétré et l'axe s'arrête.</li> <li>TRUE : La commande de mouvement tente de valider l'axe.</li> </ul>
StopMode	IN	Int	<ul style="list-style-type: none"> <li>0 : Arrêt d'urgence : Si une demande d'inhibition de l'axe est en cours, l'axe exécute la décélération d'urgence configurée. L'axe est bloqué dès qu'il est immobilisé.</li> <li>1 : Arrêt immédiat : Si une demande d'inhibition de l'axe est en cours, l'axe est arrêté sans décélération. La sortie d'impulsions est arrêtée immédiatement.</li> <li>2 : Arrêt d'urgence avec commande d'à-coup : Si une demande d'inhibition de l'axe est en cours, l'axe exécute la décélération d'urgence configurée. Si la commande d'à-coup est activée, l'à-coup configuré est pris en compte. L'axe est bloqué dès qu'il est immobilisé.</li> </ul>
Status	OUT	Bool	<p>Etat de libération de l'axe :</p> <ul style="list-style-type: none"> <li>FALSE : L'axe est bloqué : <ul style="list-style-type: none"> <li>L'axe n'exécute pas de tâches de commande de mouvement et n'accepte pas de nouvelles tâches (à l'exception d'une tâche MC_Reset).</li> <li>L'axe n'est pas référencé.</li> <li>Lors du blocage, l'état ne passe pas à FAUX tant que l'axe n'est pas immobilisé.</li> </ul> </li> <li>TRUE : L'axe est libéré : <ul style="list-style-type: none"> <li>L'axe est prêt à exécuter des tâches de commande de mouvement.</li> <li>Lors de la libération de l'axe, l'état ne passe à VRAI que lorsque le signal "Entraînement prêt" est présent. Si l'interface d'entraînement "Entraînement prêt" n'a pas été configurée dans la configuration de l'axe, l'état passe à VRAI immédiatement.</li> </ul> </li> </ul>
Busy	OUT	Bool	<p>FALSE : MC_Power n'est pas actif. TRUE : MC_Power est actif.</p>
Error	OUT	Bool	<p>FALSE : Pas d'erreur TRUE : Une erreur s'est produite dans l'instruction de commande de mouvement "MC_Power" ou dans l'objet technologique associé. La cause de l'erreur figure dans les paramètres "ErrorID" et "ErrorInfo".</p>
ErrorID	OUT	Word	Code d'erreur pour le paramètre "Error"
ErrorInfo	OUT	Word	Information d'erreur pour le paramètre "ErrorID"



- ① Un axe est validé puis à nouveau bloqué. Une fois que l'entraînement a renvoyé "Entraînement prêt" à la CPU, la réussite de la libération peut être lue via "Status\_1".
- ② Après une libération d'axe, une erreur ayant entraîné Le blocage de l'axe s'est produite. L'erreur est supprimée et acquittée avec "MC\_Reset". L'axe est alors à nouveau libéré.

Procédez comme suit pour libérer un axe à interface d'entraînement configurée :

1. Vérifiez que les conditions décrites plus haut sont bien remplies.
2. Initialisez le paramètre d'entrée "StopMode" à la valeur désirée. Définissez le paramètre d'entrée "Enable" à VRAI.

La sortie de libération pour "Entraînement libéré" passe à VRAI pour laisser passer le courant vers l'entraînement. La CPU attend le signal "Entraînement prêt" de l'entraînement.

Lorsque le signal "Entraînement prêt" est disponible sur l'entrée Prêt configurée de la CPU, l'axe est libéré. Le paramètre de sortie "Status" et la variable d'objet technologique <nom axe>.StatusBits.Enable prennent la valeur VRAI.

Procédez comme suit pour libérer un axe sans interface d'entraînement configurée :

1. Vérifiez que les conditions décrites plus haut sont bien remplies.
2. Initialisez le paramètre d'entrée "StopMode" à la valeur désirée. Définissez le paramètre d'entrée "Enable" à VRAI. L'axe est libéré. Le paramètre de sortie "Status" et la variable d'objet technologique <nom d'axe>.StatusBits.Enable prennent la valeur VRAI.

Procédez comme suit pour bloquer un axe :


1. Immobilisez l'axe.

Vous pouvez savoir quand l'axe est immobile à l'aide de la variable d'objet technologique <nom d'axe>.StatusBits.StandStill.

2. Définissez le paramètre d'entrée "Enable" à FAUX une fois l'axe immobilisé.
3. Les paramètres de sortie "Busy" et "Status" et la variable d'objet technologique <nom d'axe>.StatusBits.Enable prennent la valeur FAUX lorsque le blocage de l'axe est achevé.

### 9.3.3.2 Instruction MC\_Reset

Tableau 9- 28 Instruction MC\_Reset

CONT/LOG	SCL	Description
	<pre>"MC_Reset_DB" (   Axis:=_multi_fb_in_,   Execute:=_bool_in_,   Restart:=_bool_in_,   Done=&gt;_bool_out_,   Busy=&gt;_bool_out_,   Error=&gt;_bool_out_,   ErrorID=&gt;_word_out_,   ErrorInfo=&gt;_word_out_);</pre>	<p>Utilisez l'instruction MC_Reset pour acquitter les erreurs "Erreur de fonctionnement avec arrêt de l'axe" et "Erreur de configuration". Vous trouverez les erreurs nécessitant un acquittement dans la liste des ErrorIDs et ErrorInfos, sous "Solution".</p> <p>Avant d'utiliser l'instruction MC_Reset, vous devez avoir éliminé la cause de l'erreur de configuration en attente nécessitant un acquittement (par exemple, en changeant une valeur d'accélération invalide dans l'objet technologique Axe en une valeur valide).</p> <p>A partir de V3.0, le paramètre Restart permet de charger la configuration d'axe en mémoire de travail à l'état de fonctionnement MARCHE.</p>

- 1 STEP 7 crée automatiquement le DB lorsque vous insérez l'instruction.
- 2 Dans l'exemple SCL, "MC\_Reset\_DB" est le nom du DB d'instance.

La tâche MC\_Reset ne peut être annulée par aucune autre tâche de commande de mouvement. La nouvelle tâche MC\_Reset n'annule aucune autre tâche de commande de mouvement active.

Tableau 9- 29 Paramètres pour l'instruction MC\_Reset

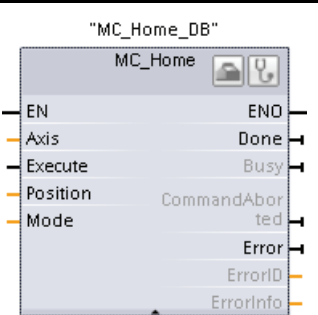
Paramètre et type	Type de données	Description
Axis	IN	TO_Axis_1 Objet technologique Axe
Execute	IN	Bool Démarrage de la tâche en présence d'un front montant
Restart	IN	Bool TRUE = Chargement de la configuration d'axe de la mémoire de chargement dans la mémoire de travail. La commande ne peut être exécutée que lorsque l'axe est bloqué. FALSE = Acquittement des erreurs en attente
Done	OUT	Bool VRAI = L'erreur a été acquittée.
Busy	OUT	Bool VRAI = La tâche est en cours d'exécution.
Error	OUT	Bool VRAI = Une erreur s'est produite pendant l'exécution de la tâche. La cause de l'erreur figure dans les paramètres "ErrorID" et "ErrorInfo".
ErrorID	OUT	Word Code d'erreur pour le paramètre "Error"
ErrorInfo	OUT	Word Information d'erreur pour le paramètre "ErrorID"

Procédez comme suit pour acquitter une erreur avec MC\_Reset :

1. Vérifiez que les conditions décrites plus haut sont bien remplies.
2. Lancez l'acquittement de l'erreur en présence d'un front montant dans le paramètre d'entrée Execute.
3. On sait que l'erreur a été acquittée lorsque Done est égal à VRAI et que la variable d'objet technologique <nom d'axe>.StatusBits.Error est égal à FAUX.

### 9.3.3.3 Instruction MC\_Home

Tableau 9- 30 Instruction MC\_Home

CONT/LOG	SCL	Description
	<pre>"MC_Home_DB" (   Axis:=_multi_fb_in_,   Execute:=_bool_in_,   Position:=_real_in_,   Mode:=_int_in_,   Done=&gt;_bool_out_,   Busy=&gt;_bool_out_,   CommandAborted=&gt;_bool_out_,   Error=&gt;_bool_out_,   ErrorID=&gt;_word_out_,   ErrorInfo=&gt;_word_out_);</pre>	<p>Utilisez l'instruction MC_Home pour établir une correspondance entre les coordonnées de l'axe et la position physique réelle de l'entraînement. Le référencement est nécessaire pour un positionnement absolu de l'axe.</p> <p>L'utilisation de l'instruction MC_Home n'est possible que si l'axe est libéré</p>

- 1 STEP 7 crée automatiquement le DB lorsque vous insérez l'instruction.
- 2 Dans l'exemple SCL, "MC\_Home\_DB" est le nom du DB d'instance.

Les modes de référencement suivants sont disponibles :

- Référencement direct absolu (Mode = 0) : La valeur du paramètre "Position" est prise comme position en cours de l'axe.
- Référencement direct relatif (Mode = 1) : La position en cours de l'axe est décalée de la valeur du paramètre "Position".
- Référencement passif (Mode = 2) : Lors du référencement passif, l'instruction MC\_Home n'exécute aucune prise de référence. L'accostage de la came de référence doit être réalisé par l'utilisateur au moyen d'instructions de commande de mouvement. L'axe est référencé lorsque la came de référence est détectée.
- Référencement actif (Mode = 3) : La prise de référence est exécutée automatiquement.

Tableau 9- 31 Paramètres pour l'instruction MC\_Home

Paramètre et type	Type de données	Description
Axis	IN	TO_Axis_PTO Objet technologique Axe
Execute	IN	Bool Démarrage de la tâche en présence d'un front montant
Position	IN	Real <ul style="list-style-type: none"> <li>• Mode = 0, 2 et 3 (position absolue de l'axe à l'achèvement du référencement)</li> <li>• Mode = 1 (valeur de correction pour la position en cours de l'axe)</li> </ul> Valeurs limites : $-1,0e^{12} \leq \text{Position} \leq 1,0e^{12}$

Paramètre et type		Type de données	Description
Mode	IN	Int	<p>Mode de référencement</p> <ul style="list-style-type: none"> <li>0: Référencement direct absolu La nouvelle position de l'axe correspond à la valeur de position du paramètre "Position".</li> <li>1: Référencement direct relatif La nouvelle position de l'axe correspond à la position en cours de l'axe + la valeur de position du paramètre "Position".</li> <li>2: Référencement passif Référencement selon la configuration de l'axe. Après le référencement, la valeur du paramètre "Position" est prise comme nouvelle position de l'axe.</li> <li>3: Référencement actif Prise de référence selon la configuration de l'axe. Après le référencement, la valeur du paramètre "Position" est prise comme nouvelle position de l'axe.</li> </ul>
Done	OUT	Bool	VRAI = Tâche achevée
Busy	OUT	Bool	VRAI = La tâche est en cours d'exécution.
CommandAborted	OUT	Bool	VRAI = Pendant l'exécution, la tâche a été annulée par une autre tâche.
Error	OUT	Bool	VRAI = Une erreur s'est produite pendant l'exécution de la tâche. La cause de l'erreur figure dans les paramètres "ErrorID" et "ErrorInfo".
ErrorID	OUT	Word	Code d'erreur pour le paramètre "Error"
ErrorInfo	OUT	Word	Information d'erreur pour le paramètre "ErrorID"

**Remarque**

**Perte du référencement de l'axe dans les cas suivants**

- Blocage de l'axe par l'instruction MC\_Power
- Commutation entre mode automatique et mode manuel
- Lors du démarrage d'un référencement actif (le référencement de l'axe est à nouveau disponible à l'achèvement de l'opération de référencement)
- Après mise hors tension puis sous tension de la CPU
- Après un redémarrage de la CPU (transition MARCHE à ARRET ou ARRET à MARCHE)

Procédez comme suit pour référencer l'axe :

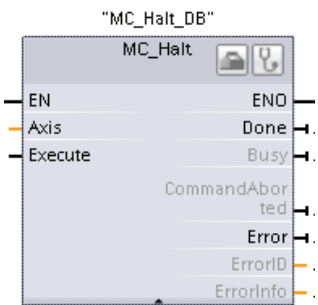
1. Vérifiez que les conditions décrites plus haut sont bien remplies.
2. Initialisez les paramètres d'entrée nécessaires et lancez l'opération de référencement à l'aide d'un front montant dans le paramètre d'entrée "Execute".
3. On sait que le référencement est achevé lorsque le paramètre de sortie "Done" et la variable d'objet technologique <nom d'axe>.StatusBits.HomingDone prennent la valeur VRAI.

Tableau 9- 32 Comportement d'annulation

Mode	Description	
0 ou 1	La tâche MC_Home ne peut être annulée par aucune autre tâche de commande de mouvement. La nouvelle tâche MC_Home n'annule aucune tâche de commande de mouvement active. Les tâches de déplacement par rapport à une position reprennent après le référencement en fonction de la nouvelle position de référence (valeur dans le paramètre d'entrée Position).	
2	La tâche MC_Home peut être annulée par les tâches de commande de mouvement suivantes : Tâche MC_Home Mode = 2, 3 : La nouvelle tâche MC_Home annule la tâche de commande de mouvement active suivante. Tâche MC_Home Mode = 2 : Les tâches de déplacement par rapport à une position reprennent après le référencement en fonction de la nouvelle position de référence (valeur dans le paramètre d'entrée Position).	
3	La tâche MC_Home peut être annulée par les tâches de commande de mouvement suivantes : <ul style="list-style-type: none"> <li>• MC_Home Mode = 3</li> <li>• MC_Halt</li> <li>• MC_MoveAbsolute</li> <li>• MC_MoveRelative</li> <li>• MC_MoveVelocity</li> <li>• MC_MoveJog</li> </ul>	La nouvelle tâche MC_Home annule les tâches de commande de mouvement actives suivantes : <ul style="list-style-type: none"> <li>• MC_Home Mode = 2, 3</li> <li>• MC_Halt</li> <li>• MC_MoveAbsolute</li> <li>• MC_MoveRelative</li> <li>• MC_MoveVelocity</li> <li>• MC_MoveJog</li> </ul>

9.3.3.4 Instruction MC\_Halt

Tableau 9- 33 Instruction MC\_Halt

CONT/LOG	SCL	Description
	<pre>"MC_Halt_DB" (   Axis:= _multi_fb_in_,   Execute:= _bool_in_,   Done=&gt; _bool_out_,   Busy=&gt; _bool_out_,   CommandAborted=&gt; _bool_out_,   Error=&gt; _bool_out_,   ErrorID=&gt; _word_out_,   ErrorInfo=&gt; _word_out_);</pre>	<p>Utilisez l'instruction MC_Halt pour arrêter tout mouvement et immobiliser l'axe. La position d'immobilisation n'est pas définie. L'utilisation de l'instruction MC_Halt n'est possible que si l'axe est libéré.</p>

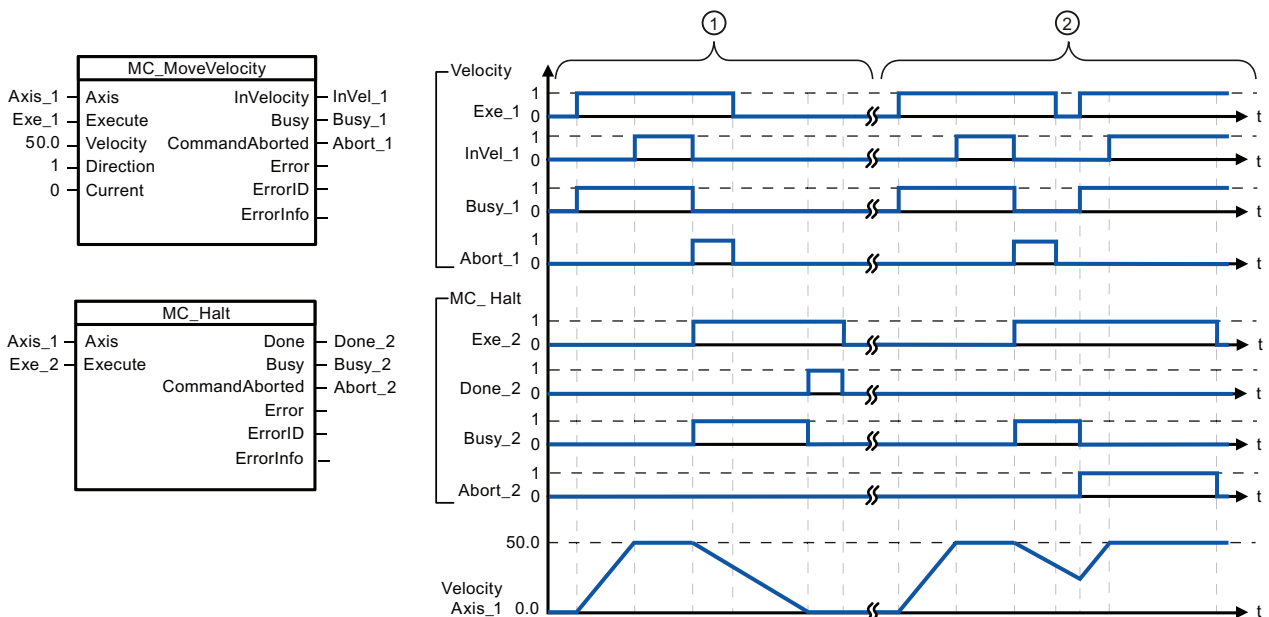
1 STEP 7 crée automatiquement le DB lorsque vous insérez l'instruction.

2 Dans l'exemple SCL, "MC\_Halt\_DB" est le nom du DB d'instance.

Tableau 9- 34 Paramètres pour l'instruction MC\_Halt

Paramètre et type	Type de données	Description	
Axis	IN	TO_Axis_1	Objet technologique Axe
Execute	IN	Bool	Démarrage de la tâche en présence d'un front montant
Done	OUT	Bool	VRAI = La vitesse zéro est atteinte.
Busy	OUT	Bool	VRAI = La tâche est en cours d'exécution.
CommandAborted	OUT	Bool	VRAI = Pendant l'exécution, la tâche a été annulée par une autre tâche.
Error	OUT	Bool	VRAI = Une erreur s'est produite pendant l'exécution de la tâche. La cause de l'erreur figure dans les paramètres "ErrorID" et "ErrorInfo".
ErrorID	OUT	Word	Code d'erreur pour le paramètre "Error"
ErrorInfo	OUT	Word	Information d'erreur pour le paramètre "ErrorID"





Les valeurs suivantes ont été configurées dans la fenêtre de configuration "Dynamique > Générale" : Accélération = 10,0 et Décélération = 5,0

- ① L'axe est freiné jusqu'à l'immobilisation par une tâche MC\_Halt. L'immobilisation de l'axe est signalée via "Done\_2".
- ② Alors qu'une tâche MC\_Halt freine l'axe, cette tâche est annulée par une autre tâche de mouvement. L'annulation est signalée via "Abort\_2".

### Comportement d'annulation

La tâche MC\_Halt peut être annulée par les tâches de commande de mouvement suivantes :

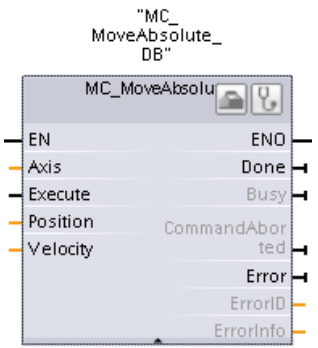
- MC\_Home Mode = 3
- MC\_Halt
- MC\_MoveAbsolute
- MC\_MoveRelative
- MC\_MoveVelocity
- MC\_MoveJog

La nouvelle tâche MC\_Halt annule les tâches de commande de mouvement actives suivantes :

- MC\_Home Mode = 3
- MC\_Halt
- MC\_MoveAbsolute
- MC\_MoveRelative
- MC\_MoveVelocity
- MC\_MoveJog

9.3.3.5 Instruction MC\_MoveAbsolute

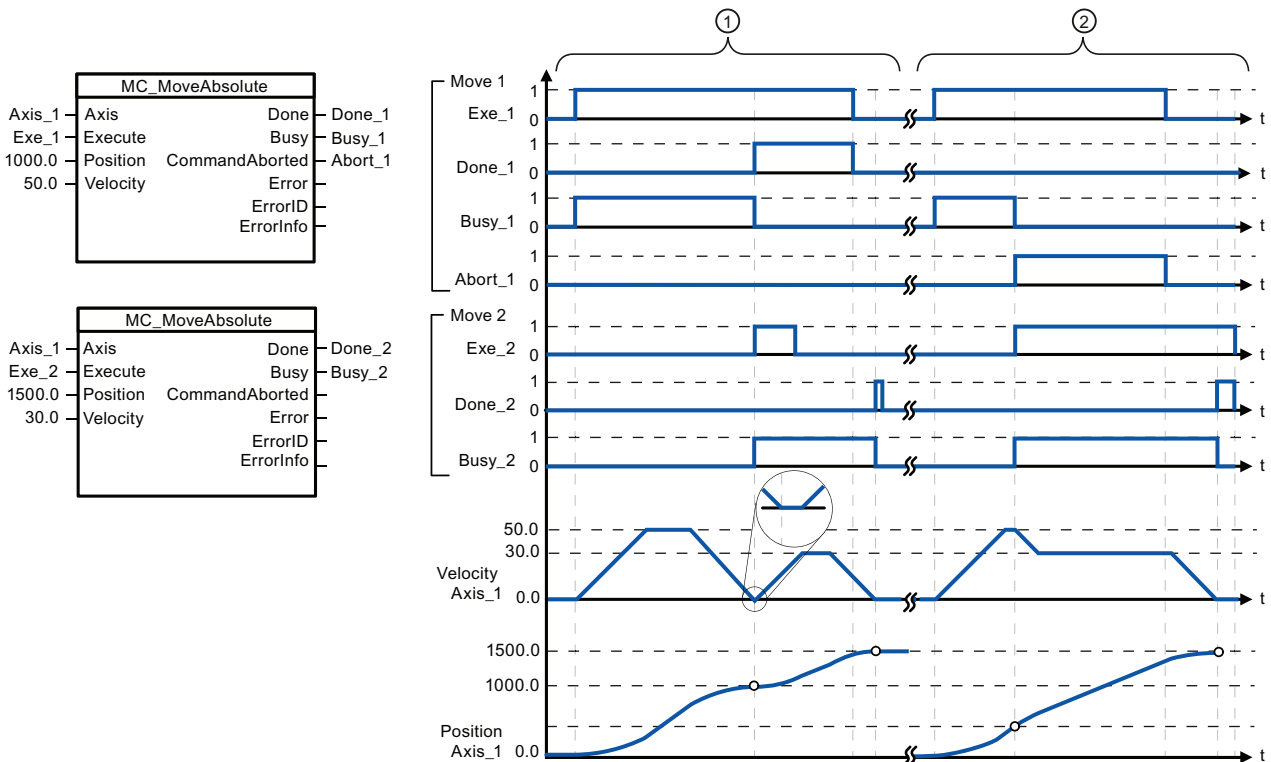
Tableau 9- 35 Instruction MC\_MoveAbsolute

CONT/LOG	SCL	Description
	<pre>"MC_MoveAbsolute_DB" (   Axis:=_multi_fb_in_,   Execute:=_bool_in_,   Position:=_real_in_,   Velocity:=_real_in_,   Done=&gt;_bool_out_,   Busy=&gt;_bool_out_,   CommandAborted=&gt;_bool_out_,   Error=&gt;_bool_out_,   ErrorID=&gt;_word_out_,   ErrorInfo=&gt;_word_out_);</pre>	<p>Utilisez l'instruction MC_MoveAbsolute pour démarrer un déplacement de positionnement de l'axe à une position absolue.</p> <p>L'utilisation de l'instruction MC_MoveAbsolute n'est possible que si l'axe est libéré et référencé.</p>

- 1 STEP 7 crée automatiquement le DB lorsque vous insérez l'instruction.
- 2 Dans l'exemple SCL, "MC\_MoveAbsolute\_DB" est le nom du DB d'instance.

Tableau 9- 36 Paramètres pour l'instruction MC\_MoveAbsolute

Paramètre et type	Type de données	Description
Axis	IN	TO_Axis_1 Objet technologique Axe
Execute	IN	Bool Démarrage de la tâche en présence d'un front montant (valeur par défaut : faux)
Position	IN	Real Position cible absolue (valeur par défaut : 0,0) 0.0) Valeurs limites : $-1,0e^{12} \leq \text{Position} \leq 1,0e^{12}$
Velocity	IN	Real Vitesse de l'axe (valeur par défaut : 10.0) Cette vitesse n'est pas toujours atteinte en raison de l'accélération et de la décélération configurées ainsi que de la position cible à accoster. Valeurs limites : vitesse de démarrage/d'arrêt $\leq \text{Velocity} \leq$ vitesse maximale
Done	OUT	Bool VRAI = La position cible absolue est atteinte.
Busy	OUT	Bool VRAI = La tâche est en cours d'exécution.
CommandAborted	OUT	Bool VRAI = Pendant l'exécution, la tâche a été annulée par une autre tâche.
Error	OUT	Bool VRAI = Une erreur s'est produite pendant l'exécution de la tâche. La cause de l'erreur figure dans les paramètres "ErrorID" et "ErrorInfo".
ErrorID	OUT	Word Code d'erreur pour le paramètre "Error" (valeur par défaut : 0000) 0000)
ErrorInfo	OUT	Word Information d'erreur pour le paramètre "ErrorID" (valeur par défaut : 0000) 0000)



Les valeurs suivantes ont été configurées dans la fenêtre de configuration "Dynamique > Générale" : Accélération = 10,0 et Décélération = 10,0

- ① Un axe est déplacé à la position absolue 1000,0 avec une tâche MC\_MoveAbsolute. Lorsque l'axe atteint la position cible, cela est signalé via "Done\_1". Lorsque "Done\_1" = VRAI, une autre tâche MC\_MoveAbsolute, avec une position cible 1500,0, est lancée. Du fait des temps de réponse (par exemple, temps de cycle du programme utilisateur, etc.), l'axe s'immobilise brièvement (voir le détail agrandi). Lorsque l'axe atteint la nouvelle position cible, cela est signalé via "Done\_2".
- ② Une tâche MC\_MoveAbsolute active est annulée par une autre tâche MC\_MoveAbsolute. L'annulation est signalée via "Abort\_1". L'axe est ensuite déplacé à la nouvelle vitesse vers la nouvelle position cible 1500,0. Lorsque la nouvelle position cible est atteinte, cela est signalé via "Done\_2".

### Comportement d'annulation

La tâche MC\_MoveAbsolute peut être annulée par les tâches de commande de mouvement suivantes :

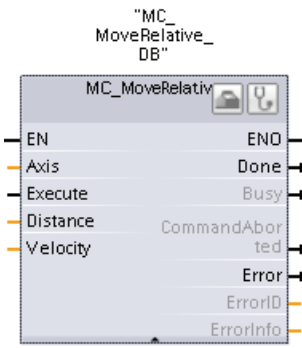
- MC\_Home Mode = 3
- MC\_Halt
- MC\_MoveAbsolute
- MC\_MoveRelative
- MC\_MoveVelocity
- MC\_MoveJog

La nouvelle tâche MC\_MoveAbsolute annule les tâches de commande de mouvement actives suivantes :

- MC\_Home Mode = 3
- MC\_Halt
- MC\_MoveAbsolute
- MC\_MoveRelative
- MC\_MoveVelocity
- MC\_MoveJog

9.3.3.6 Instruction MC\_MoveRelative

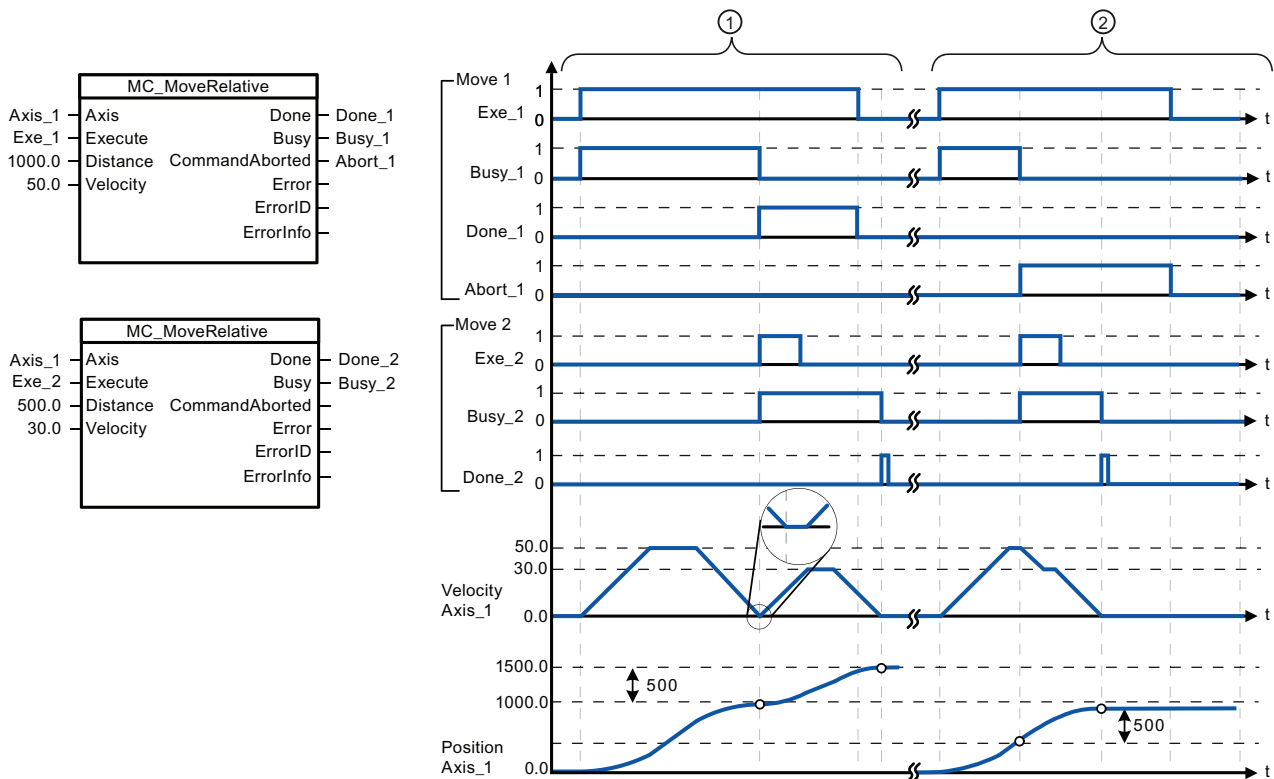
Tableau 9- 37 Instruction MC\_MoveRelative

CONT/LOG	SCL	Description
	<pre>"MC_MoveRelative_DB" (   Axis:= _multi_fb_in_,   Execute:= _bool_in_,   Distance:= _real_in_,   Velocity:= _real_in_,   Done=&gt; _bool_out_,   Busy=&gt; _bool_out_,   CommandAborted=&gt; _bool_out_,   Error=&gt; _bool_out_,   ErrorID=&gt; _word_out_,   ErrorInfo=&gt; _word_out_);</pre>	<p>Utilisez l'instruction MC_MoveRelative pour démarrer un déplacement de positionnement relatif de l'axe par rapport à une position initiale..</p> <p>L'utilisation de l'instruction MC_MoveRelative n'est possible que si l'axe est libéré.</p>

- 1 STEP 7 crée automatiquement le DB lorsque vous insérez l'instruction.
- 2 Dans l'exemple SCL, "MC\_MoveRelative\_DB" est le nom du DB d'instance.

Tableau 9- 38 Paramètres pour l'instruction MC\_MoveRelative

Paramètre et type		Type de données	Description
Axis	IN	TO_Axis_1	Objet technologique Axe
Execute	IN	Bool	Démarrage de la tâche en présence d'un front montant (valeur par défaut : faux)
Distance	IN	Real	Distance à parcourir pour l'opération de positionnement (valeur par défaut : 0,0) 0.0) Valeurs limites : $-1,0e^{12} \leq \text{Distance} \leq 1,0e^{12}$
Velocity	IN	Real	Vitesse de l'axe (valeur par défaut : 10.0) Cette vitesse n'est pas toujours atteinte en raison de l'accélération et de la décélération configurées ainsi que de la distance à parcourir. Valeurs limites : vitesse de démarrage/d'arrêt $\leq \text{Velocity} \leq$ vitesse maximale
Done	OUT	Bool	VRAI = La position cible est atteinte.
Busy	OUT	Bool	VRAI = La tâche est en cours d'exécution.
CommandAborted	OUT	Bool	VRAI = Pendant l'exécution, la tâche a été annulée par une autre tâche.
Error	OUT	Bool	VRAI = Une erreur s'est produite pendant l'exécution de la tâche. La cause de l'erreur figure dans les paramètres "ErrorID" et "ErrorInfo".
ErrorID	OUT	Word	Code d'erreur pour le paramètre "Error" (valeur par défaut : 0000) 0000)
ErrorInfo	OUT	Word	Information d'erreur pour le paramètre "ErrorID" (valeur par défaut : 0000) 0000)



Les valeurs suivantes ont été configurées dans la fenêtre de configuration "Dynamique > Générale" : Accélération = 10,0 et Décélération = 10,0

- ① L'axe est déplacé par une tâche MC\_MoveRelative de la distance ("Distance") 1000,0. Lorsque l'axe atteint la position cible, cela est signalé via "Done\_1". Lorsque "Done\_1" = VRAI, une autre tâche MC\_MoveRelative, avec une distance de déplacement de 500,0, est lancée. Du fait des temps de réponse (par exemple, temps de cycle du programme utilisateur), l'axe s'immobilise brièvement (voir le détail agrandi). Lorsque l'axe atteint la nouvelle position cible, cela est signalé via "Done\_2".
- ② Une tâche MC\_MoveRelative active est annulée par une autre tâche MC\_MoveRelative. L'annulation est signalée via "Abort\_1". L'axe est ensuite déplacé de la nouvelle distance ("Distance") 500,0 à la nouvelle vitesse. Lorsque la nouvelle position cible est atteinte, cela est signalé via "Done\_2".

### Comportement d'annulation

La tâche MC\_MoveRelative peut être annulée par les tâches de commande de mouvement suivantes :

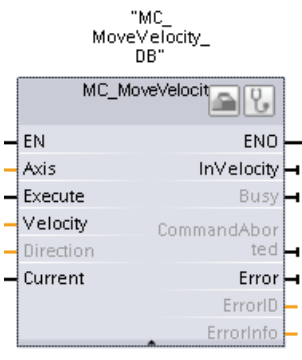
- MC\_Home Mode = 3
- MC\_Halt
- MC\_MoveAbsolute
- MC\_MoveRelative
- MC\_MoveVelocity
- MC\_MoveJog

La nouvelle tâche MC\_MoveRelative annule les tâches de commande de mouvement actives suivantes :

- MC\_Home Mode = 3
- MC\_Halt
- MC\_MoveAbsolute
- MC\_MoveRelative
- MC\_MoveVelocity
- MC\_MoveJog

9.3.3.7 Instruction MC\_MoveVelocity

Tableau 9- 39 Instruction MC\_MoveVelocity

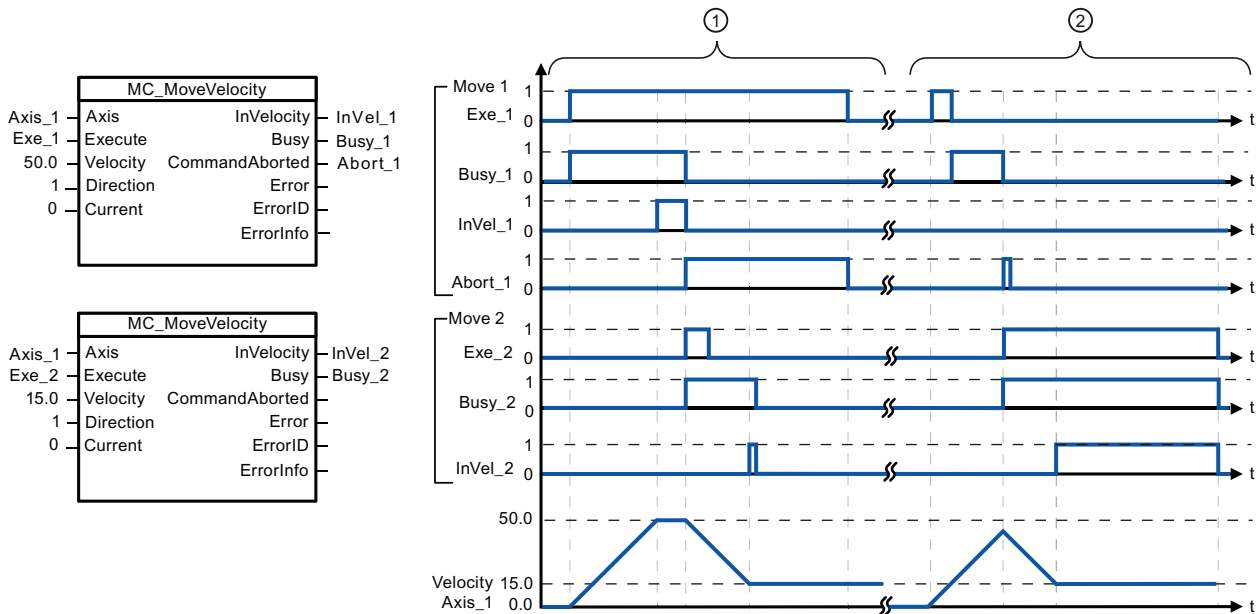
CONT/LOG	SCL	Description
	<pre>"MC_MoveVelocity_DB" (   Axis:=_multi_fb_in_,   Execute:=_bool_in_,   Velocity:=_real_in_,   Direction:=_int_in_,   Current:=_bool_in_,   InVelocity=&gt;_bool_out_,   Busy=&gt;_bool_out_,   CommandAborted=&gt;_bool_out_,   Error=&gt;_bool_out_,   ErrorID=&gt;_word_out_,   ErrorInfo=&gt;_word_out_);</pre>	<p>Utilisez l'instruction MC_MoveVelocity pour déplacer l'axe à la vitesse prédéfinie.</p> <p>L'utilisation de l'instruction MC_MoveVelocity n'est possible que si l'axe est libéré.</p>

- 1 STEP 7 crée automatiquement le DB lorsque vous insérez l'instruction.
- 2 Dans l'exemple SCL, "MC\_MoveVelocity\_DB" est le nom du DB d'instance.

Tableau 9- 40 Paramètres pour l'instruction MC\_MoveVelocity

Paramètre et type		Type de données	Description
Axis	IN	TO_Axis_1	Objet technologique Axe
Execute	IN	Bool	Démarrage de la tâche en présence d'un front montant (valeur par défaut : faux)
Velocity	IN	Real	Indication de vitesse pour le déplacement de l'axe (valeur par défaut : 10,0) 10.0) Valeurs limites : vitesse de démarrage/d'arrêt ≤  Velocity  ≤ vitesse maximale (Velocity = 0,0 est autorisé)
Direction	IN	Int	Indication du sens : <ul style="list-style-type: none"> <li>• 0: 0 : Le sens de rotation correspond au signe de la valeur dans le paramètre "Velocity" (valeur par défaut).</li> <li>• 1: 1 : Sens de rotation positif (le signe de la valeur dans le paramètre "Velocity" n'est pas pris en compte)</li> <li>• 2: Sens de rotation négatif (le signe de la valeur dans le paramètre "Velocity" n'est pas pris en compte)</li> </ul>

Paramètre et type		Type de données	Description
Current	IN	Bool	<p>Conserver la vitesse actuelle :</p> <ul style="list-style-type: none"> <li>• FAUX : "Conserver la vitesse actuelle" est désactivé. Les valeurs des paramètres "Velocity" et "Direction" sont utilisées. (valeur par défaut)</li> <li>• VRAI : "Conserver la vitesse actuelle" est activé. Les valeurs des paramètres "Velocity" et "Direction" ne sont pas prises en compte.</li> </ul> <p>Lorsque l'axe reprend le déplacement à la vitesse en cours, le paramètre "InVelocity" renvoie la valeur VRAI (TRUE).</p>
InVelocity	OUT	Bool	<p>VRAI :</p> <ul style="list-style-type: none"> <li>• Si "Current" = FAUX (FALSE) : La vitesse indiquée dans le paramètre "Velocity" a été atteinte.</li> <li>• Si "Current" = VRAI (TRUE) : L'axe se déplace à la vitesse en cours au démarrage.</li> </ul>
Busy	OUT	Bool	VRAI = La tâche est en cours d'exécution.
CommandAborted	OUT	Bool	VRAI = Pendant l'exécution, la tâche a été annulée par une autre tâche.
Error	OUT	Bool	VRAI = Une erreur s'est produite pendant l'exécution de la tâche. La cause de l'erreur figure dans les paramètres "ErrorID" et "ErrorInfo".
ErrorID	OUT	Word	Code d'erreur pour le paramètre "Error" (valeur par défaut : 0000) 0000)
ErrorInfo	OUT	Word	Information d'erreur pour le paramètre "ErrorID" (valeur par défaut : 0000) 0000)



Les valeurs suivantes ont été configurées dans la fenêtre de configuration "Dynamique > Générale" : Accélération = 10,0 et Décélération = 10,0

- ① Une tâche MC\_MoveVelocity active utilise "InVel\_1" pour signaler que sa vitesse cible a été atteinte. Elle est ensuite annulée par une autre tâche MC\_MoveVelocity. L'annulation est signalée via "Abort\_1". Lorsque la nouvelle vitesse cible de 15,0 est atteinte, cela est signalé via "InVel\_2". L'axe poursuit alors son déplacement à la nouvelle vitesse constante.
- ② Une tâche active MC\_MoveVelocity est annulée par une autre tâche MC\_MoveVelocity avant d'atteindre sa vitesse cible. L'annulation est signalée via "Abort\_1". Lorsque la nouvelle vitesse cible de 15,0 est atteinte, cela est signalé via "InVel\_2". L'axe poursuit alors son déplacement à la nouvelle vitesse constante.

### Comportement d'annulation

La tâche MC\_MoveVelocity peut être annulée par les tâches de commande de mouvement suivantes :

- MC\_Home Mode = 3
- MC\_Halt
- MC\_MoveAbsolute
- MC\_MoveRelative
- MC\_MoveVelocity
- MC\_MoveJog

La nouvelle tâche MC\_MoveVelocity annule les tâches de commande de mouvement actives suivantes :

- MC\_Home Mode = 3
- MC\_Halt
- MC\_MoveAbsolute
- MC\_MoveRelative
- MC\_MoveVelocity
- MC\_MoveJog



**Remarque**

**Comportement en cas de vitesse définie à 0 (Velocity = 0,0)**

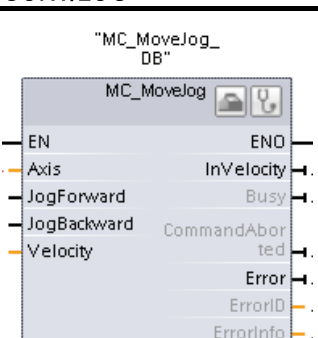
Une tâche MC\_MoveVelocity avec "Velocity" = 0,0 (par exemple, une tâche MC\_Halt) annule les tâches de déplacement actives et arrête l'axe avec la décélération configurée. Lorsque l'axe s'immobilise, le paramètre de sortie "InVelocity" indique VRAI pendant au moins un cycle du programme.

"Busy" indique la valeur VRAI pendant l'opération de décélération et passe à FAUX conjointement avec "InVelocity". Si le paramètre "Execute" est VRAI, "InVelocity" et "Busy" sont verrouillés.

Lorsque la tâche MC\_MoveVelocity est démarrée, le bit d'état "SpeedCommand" est mis à 1 dans l'objet technologique. Le bit d'état "ConstantVelocity" est mis à 1 lors de l'immobilisation de l'axe. Les deux bits sont adaptés à la nouvelle situation lorsqu'une nouvelle tâche de déplacement est démarrée.

**9.3.3.8 Instruction MC\_MoveJog**

Tableau 9- 41 Instruction MC\_MoveJog

CONT/LOG	SCL	Description
	<pre>"MC_MoveJog_DB" (   Axis:=_multi_fb_in_,   JogForward:=_bool_in_,   JogBackward:=_bool_in_,   Velocity:=_real_in_,   InVelocity=&gt;_bool_out_,   Busy=&gt;_bool_out_,   CommandAborted=&gt;_bool_out_,   Error=&gt;_bool_out_,   ErrorID=&gt;_word_out_,   ErrorInfo=&gt;_word_out_);</pre>	<p>Utilisez l'instruction MC_MoveJog pour déplacer l'axe à la vitesse prédéfinie en mode Manuel à vue. Cette instruction sert typiquement au test et à la mise en service.</p> <p>L'utilisation de l'instruction MC_MoveJog n'est possible que si l'axe est libéré.</p>

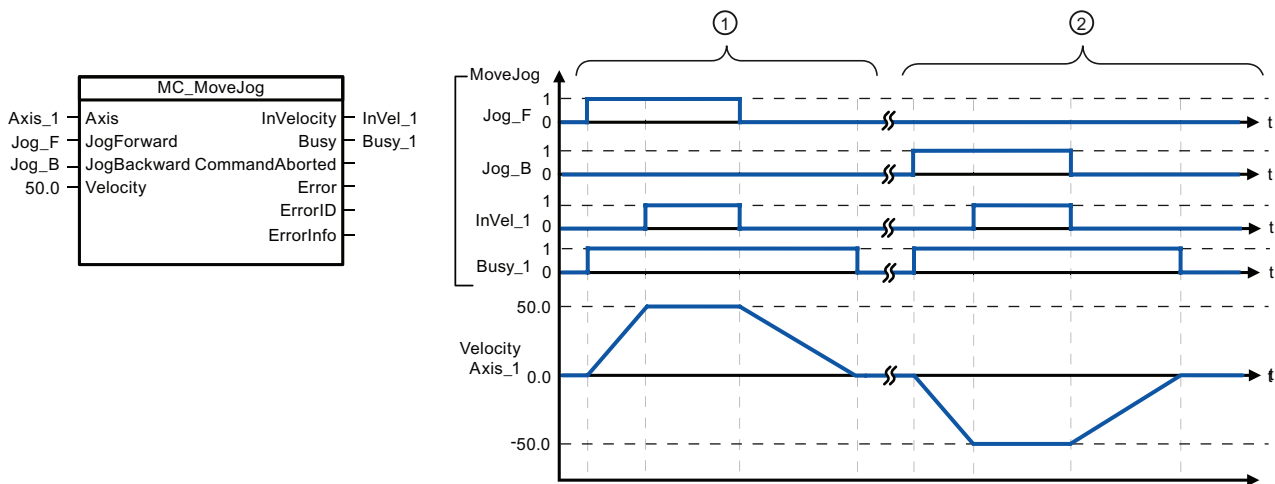
- 1 STEP 7 crée automatiquement le DB lorsque vous insérez l'instruction.
- 2 Dans l'exemple SCL, "MC\_MoveJog\_DB" est le nom du DB d'instance.

Tableau 9- 42 Paramètres pour l'instruction MC\_MoveJog

Paramètre et type		Type de données	Description
Axis	IN	TO_Axis_1	Objet technologique Axe
JogForward <sup>1</sup>	IN	Bool	Tant que le paramètre est VRAI, l'axe se déplace dans le sens positif à la vitesse indiquée dans le paramètre "Velocity". Le signe de la valeur dans le paramètre "Velocity" n'est pas pris en compte. (valeur par défaut : faux)

Paramètre et type		Type de données	Description
JogBackward <sup>1</sup>	IN	Bool	Tant que le paramètre est VRAI, l'axe se déplace dans le sens négatif à la vitesse indiquée dans le paramètre "Velocity". Le signe de la valeur dans le paramètre "Velocity" n'est pas pris en compte. (valeur par défaut : faux)
Velocity	IN	Real	Vitesse prédéfinie pour le mode Manuel à vue (valeur par défaut : 10,0) 10.0) Valeurs limites : vitesse de démarrage/d'arrêt ≤  Velocity  ≤ vitesse maximale
InVelocity	OUT	Bool	VRAI = La vitesse indiquée dans le paramètre "Velocity" a été atteinte.
Busy	OUT	Bool	VRAI = La tâche est en cours d'exécution.
CommandAborted	OUT	Bool	VRAI = Pendant l'exécution, la tâche a été annulée par une autre tâche.
Error	OUT	Bool	VRAI = Une erreur s'est produite pendant l'exécution de la tâche. La cause de l'erreur figure dans les paramètres "ErrorID" et "ErrorInfo".
ErrorID	OUT	Word	Code d'erreur pour le paramètre "Error" (valeur par défaut : 0000) 0000)
ErrorInfo	OUT	Word	Information d'erreur pour le paramètre "ErrorID" (valeur par défaut : 0000) 0000)

<sup>1</sup> Si les deux paramètres JogForward et JogBackward sont simultanément à VRAI, l'axe s'arrête avec la décélération configurée. Une erreur est signalée dans les paramètres "Error", "ErrorID" et "ErrorInfo".



Les valeurs suivantes ont été configurées dans la fenêtre de configuration "Dynamique > Générale" : Accélération = 10,0 et Décélération = 5,0

- ① "Jog\_F" déplace l'axe dans le sens positif en mode Manuel à vue. Lorsque la vitesse cible de 50,0 est atteinte, cela est signalé via "InVelo\_1". L'axe freine jusqu'à s'immobiliser à nouveau après la réinitialisation de Jog\_F.
- ② "Jog\_B" déplace l'axe dans le sens négatif en mode Manuel à vue. Lorsque la vitesse cible de 50,0 est atteinte, cela est signalé via "InVelo\_1". L'axe freine jusqu'à s'immobiliser à nouveau après la réinitialisation de Jog\_B.

### Comportement d'annulation

La tâche MC\_MoveJog peut être annulée par les tâches de commande de mouvement suivantes :

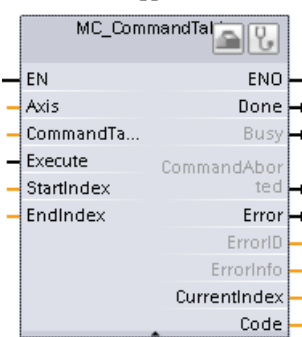
- MC\_Home Mode = 3
- MC\_Halt
- MC\_MoveAbsolute
- MC\_MoveRelative
- MC\_MoveVelocity
- MC\_MoveJog

La nouvelle tâche MC\_MoveJog annule les tâches de commande de mouvement actives suivantes :

- MC\_Home Mode = 3
- MC\_Halt
- MC\_MoveAbsolute
- MC\_MoveRelative
- MC\_MoveVelocity
- MC\_MoveJog

### 9.3.3.9 Instruction MC\_CommandTable

Tableau 9- 43 Instruction MC\_CommandTable

CONT/LOG	SCL	Description
	<pre>"MC_CommandTable_DB" (   Axis:= _multi_fb_in_,   CommandTable:= _multi_fb_in_,   Execute:= _bool_in_,   StartIndex:= _uint_in_,   EndIndex:= _uint_in_,   Done=&gt; _bool_out_,   Busy=&gt; _bool_out_,   CommandAborted=&gt; _bool_out_,   Error=&gt; _bool_out_,   ErrorID=&gt; _word_out_,   ErrorInfo=&gt; _word_out_,   CurrentIndex=&gt; _uint_out_,   Code=&gt; _word_out_ );</pre>	<p>Exécute une série de mouvements individuels pour un axe de commande de mouvement, ces mouvements individuels pouvant être combinés en une séquence de mouvement.</p> <p>Les mouvements individuels sont configurés dans une table de commande d'objet technologique pour la sortie de trains d'impulsions (TO_CommandTable_PTO).</p>

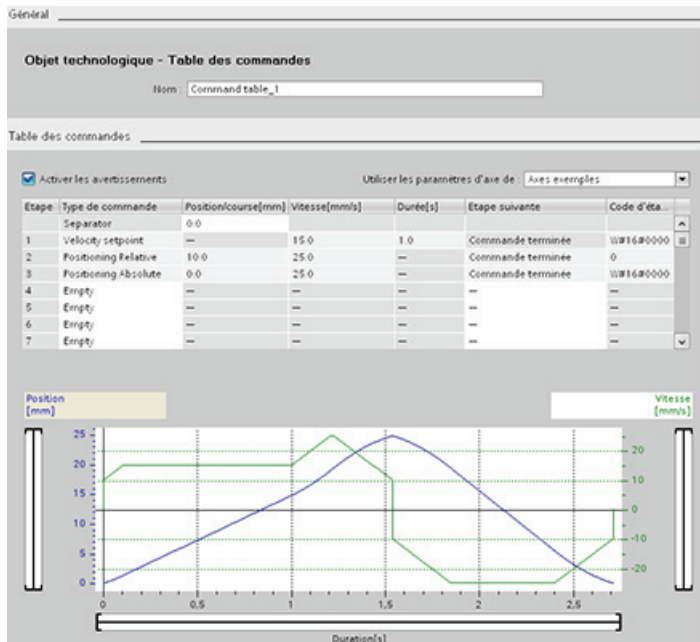
- 1 STEP 7 crée automatiquement le DB lorsque vous insérez l'instruction.
- 2 Dans l'exemple SCL, "MC\_CommandTable\_DB" est le nom du DB d'instance.

Tableau 9- 44 Paramètres pour l'instruction MC\_CommandTable

Paramètre et type	Type de données	Valeur initiale	Description	
Axis	IN	TO_Axis_1	-	Objet technologique Axe
Table	IN	TO_CommandTable_1	-	Objet technologique Table de commande
Execute	IN	Booll	FAUX	Démarrage de la tâche en présence d'un front montant
StartIndex	IN	Int	1	Démarrage du traitement de la table de commande à cet indice Limites : $1 \leq \text{StartIndex} \leq \text{EndIndex}$

Paramètre et type		Type de données	Valeur initiale	Description
EndIndex	IN	Int	32	Fin du traitement de la table de commande à cet indice Limites : StartIndex ≤ EndIndex ≤ 32
Done	OUT	Bool	FAUX	Le traitement de MC_CommandTable a été achevé avec succès.
Busy	OUT	Bool	FAUX	Opération en cours
CommandAborted	OUT	Bool	FAUX	Le traitement de la tâche a été interrompu par une autre tâche.
Error	OUT	Bool	FAUX	Une erreur s'est produite pendant le traitement. La cause est indiquée par les paramètres ErrorID et ErrorInfo..
ErrorID	OUT	Word	16#0000	Code d'erreur
ErrorInfo	OUT	Word	16#0000	Informations d'erreur
Step	OUT	Int	0	Indice en cours de traitement
Code	OUT	Word	16#0000	Identificateur défini par l'utilisateur de l'indice en cours de traitement

Vous pouvez créer la séquence de mouvement désirée dans la fenêtre de configuration "Table de commande" et vérifier le résultat dans la vue graphique du graphique de tendance.



Vous pouvez sélectionner les types de commandes qui doivent être utilisés pour le traitement de la table de commande. Il est possible d'entrer jusqu'à 32 tâches. Les commandes sont traitées l'une après l'autre.

Tableau 9- 45 Types de commandes MC\_CommandTable

Type de commande	Description
Empty	Marque de réservation pour toute commande à ajouter. L'entrée Empty n'est pas prise en compte lors du traitement de la table de commande.
Halt	Pause de l'axe. Remarque : cette commande n'est exécutée qu'après une commande de consigne de vitesse.
Positioning Relative	Positionnement de l'axe en fonction de la distance. La commande déplace l'axe de la distance indiquée à la vitesse donnée.
Positioning Absolute	Positionnement de l'axe en fonction de l'emplacement. La commande déplace l'axe à l'emplacement indiqué, en utilisant la vitesse donnée.
Velocity setpoint	Déplacement de l'axe à la vitesse donnée.
Wait	Attente de l'expiration de la durée indiquée. "Wait" n'interrompt pas un accostage actif.
Separator	Ajout d'une ligne séparatrice au-dessus de la ligne sélectionnée. La ligne séparatrice permet de définir plus d'un profil dans une seule table de commande.

Conditions requises pour l'exécution de MC\_CommandTable :

- L'objet technologique TO\_Axis\_PTO V2.0 doit être correctement configuré.
- L'objet technologique TO\_CommandTable\_PTO doit être correctement configuré.
- L'axe doit être débloqué.

#### Comportement d'annulation

La tâche MC\_CommandTable peut être annulée par les tâches de commande de mouvement suivantes :

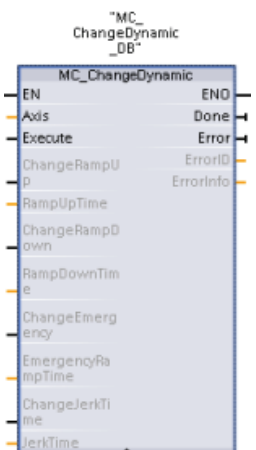
- MC\_Home Mode = 3
- MC\_Halt
- MC\_MoveAbsolute
- MC\_MoveRelative
- MC\_MoveVelocity
- MC\_MoveJog
- MC\_CommandTable

La nouvelle tâche MC\_CommandTable annule les tâches de commande de mouvement actives suivantes :

- MC\_Home Mode = 3
- MC\_Halt
- MC\_MoveAbsolute
- MC\_MoveRelative
- MC\_MoveVelocity
- MC\_MoveJog
- MC\_CommandTable
- La tâche de commande de mouvement en cours avec le lancement de la première commande "Positioning Relative", "Positioning Absolute", "Velocity setpoint" ou "Halt"

9.3.3.10 MC\_ChangeDynamic

Tableau 9- 46 Instruction MC\_ChangeDynamic

CONT/LOG	SCL	Description
	<pre>"MC_ChangeDynamic_DB" (     Execute:=_bool_in_,     ChangeRampUp:=_bool_in_,     RampUpTime:=_real_in_,     ChangeRampDown:=_bool_in_,     RampDownTime:=_real_in_,     ChangeEmergency:=_bool_in_,     EmergencyRampTime:=_real_in_,     ChangeJerkTime:=_bool_in_,     JerkTime:=_real_in_,     Done=&gt;_bool_out_,     Error=&gt;_bool_out_,     ErrorID=&gt;_word_out_,     ErrorInfo=&gt;_word_out_);</pre>	<p>Change les paramètres dynamiques d'un axe de commande de mouvement :</p> <ul style="list-style-type: none"> <li>• Changement de la valeur du temps d'accélération</li> <li>• Changement de la valeur du temps de décélération</li> <li>• Changement de la valeur du temps de décélération d'urgence</li> <li>• Changement de la valeur du temps de lissage (à-coup)</li> </ul>

- 1 STEP 7 crée automatiquement le DB lorsque vous insérez l'instruction.
- 2 Dans l'exemple SCL, "MC\_ChangeDynamic\_DB" est le nom du DB d'instance.

Tableau 9- 47 Paramètres pour l'instruction MC\_ChangeDynamic

Paramètre et type		Type de données	Description
Axis	IN	TO_Axis_1	Objet technologique Axe
Execute	IN	Bool	Démarrage de la commande en présence d'un front montant. Valeur par défaut : FALSE
ChangeRampUp	IN	Bool	TRUE = Changement du temps d'accélération conformément au paramètre d'entrée "RampUpTime". Valeur par défaut : FALSE
RampUpTime	IN	Real	Temps (en secondes) pour accélérer de l'arrêt à la vitesse maximale configurée sans limite d'à-coup. Valeur par défaut : 5.00 Le changement affecte la variable <nom d'axe>. Config.DynamicDefaults.Acceleration. L'entrée en vigueur du changement est montrée dans la description de cette variable.
ChangeRampDown	IN	Bool	TRUE = Changement du temps de décélération conformément au paramètre d'entrée "RampDownTime". Valeur par défaut : FALSE
RampDownTime	IN	Real	Temps (en secondes) pour décélérer l'axe de la vitesse maximale configurée jusqu'à l'immobilisation sans limite d'à-coup. Valeur par défaut : 5.00 Le changement affecte la variable <nom d'axe>. Config.DynamicDefaults.Deceleration. L'entrée en vigueur du changement est montrée dans la description de cette variable.

Paramètre et type		Type de données	Description
ChangeEmergency	IN	Bool	TRUE = Changement du temps de décélération d'urgence conformément au paramètre d'entrée "EmergencyRampTime". Valeur par défaut : FALSE
EmergencyRampTime	IN	Real	Temps (en secondes) pour décélérer l'axe de la vitesse maximale configurée jusqu'à l'immobilisation sans limite d'à-coup en mode d'arrêt d'urgence. Valeur par défaut : 2.00 Le changement affecte la variable <nom d'axe>. Config.DynamicDefaults.EmergencyDeceleration. L'entrée en vigueur du changement est montrée dans la description de cette variable.
ChangeJerkTime	IN	Bool	TRUE = Changement du temps de lissage selon le paramètre d'entrée "JerkTime". Valeur par défaut : FALSE
JerkTime	IN	Real	Temps de lissage (en secondes) utilisé pour les rampes d'accélération et de décélération de l'axe. Valeur par défaut : 0.25 Le changement affecte la variable <nom d'axe>. Config.DynamicDefaults.Jerk. L'entrée en vigueur du changement est montrée dans la description de cette variable.
Done	OUT	Bool	TRUE = Les valeurs modifiées ont été écrites dans le bloc de données technologique. La description des variables montre quand le changement entre en vigueur. Valeur par défaut : FALSE
Error	OUT	Bool	TRUE = Une erreur s'est produite pendant l'exécution de la commande. La cause de l'erreur figure dans les paramètres "ErrorID" et "ErrorInfo". Valeur par défaut : FALSE
ErrorID	OUT	Word	Code d'erreur. Valeur par défaut : 16#0000
ErrorInfo	IN	Word	Informations d'erreur. Valeur par défaut : 16#0000

Conditions requises pour l'exécution de MC\_ChangeDynamic :

- L'objet technologique TO\_Axis\_PTO V2.0 doit être correctement configuré.
- L'axe doit être débloqué.

### Comportement d'annulation

Une instruction MC\_ChangeDynamic ne peut être annulée par aucune autre instruction de commande de mouvement.

Une nouvelle instruction MC\_ChangeDynamic n'annule aucune tâche de commande de mouvement active.

**Remarque**

Les paramètres d'entrée "RampUpTime", "RampDownTime", "EmergencyRampTime" et "RoundingOffTime" peuvent contenir des valeurs faisant passer les paramètres d'axe résultants "accélération", "décélération", "décélération d'urgence" et "à-coup" en dehors des limites autorisées.

Veillez à maintenir les paramètres de MC\_ChangeDynamic à l'intérieur des limites des paramètres de configuration dynamiques pour l'objet technologique Axe.

**9.3.4 Fonctionnement de la commande de mouvement pour le S7-1200**

**9.3.4.1 Sorties de la CPU pour la commande de mouvement**

La CPU fournit quatre générateurs d'impulsions. Chaque générateur d'impulsions fournit une sortie d'impulsions et une sortie de sens pour la commande d'un entraînement de moteur pas à pas ou d'un entraînement de servomoteur à interface d'impulsion. La sortie d'impulsions fournit à l'entraînement les impulsions nécessaires au mouvement du moteur. La sortie de sens commande le sens de déplacement de l'entraînement.

Les sorties d'impulsions et de sens sont affectées l'une à l'autre de manière permanente. Les sorties CPU intégrées et les sorties d'un Signal Board peuvent être utilisées comme sorties d'impulsions et de sens. Vous choisissez entre sorties CPU intégrées et sorties de Signal Board pendant la configuration d'appareil, sous "Générateurs d'impulsions (PTO/PWM)" dans l'onglet "Propriétés". Seul PTO (Pulse Train Output) s'applique à la commande de mouvement.

La sortie PTO génère un signal carré de fréquence variable. La génération d'impulsions est gérée par les informations de configuration et d'exécution fournies via la configuration matérielle et/ou des SFC/SFB.

Selon la sélection de l'utilisateur, lorsque la CPU est à l'état MARCHE, ce sont soit les valeurs stockées dans la mémoire image, soit les sorties de génération d'impulsions qui pilotent les sorties TOR. A l'état ARRET, le générateur PTO ne pilote pas les sorties.

Tableau 9- 48 Affectations d'adresses des sorties d'impulsions et de sens

Utilisation des sorties pour la commande de mouvement		
	Impulsion	Sens
PTO 0		
E/S intégrées	Q0.0	Q0.1
E/S du SB	Q4.0	Q4.1
PTO 1		
E/S intégrées	Q0.2	Q0.3
E/S du SB	Q4.2	Q4.3



Utilisation des sorties pour la commande de mouvement		
PTO 2		
E/S intégrées	Q0.4 <sup>1</sup>	Q0.5 <sup>1</sup>
E/S du SB	Q4.0	Q4.1
PTO 3		
E/S intégrées	Q0.6 <sup>2</sup>	Q0.7 <sup>2</sup>
E/S du SB	Q4.2	Q4.3

- <sup>1</sup> La CPU 1211C ne comporte pas de sorties Q0.4, Q0.5, Q0.6 et Q0.7. Ces sorties ne peuvent donc pas être utilisées dans la CPU 1211C.
- <sup>2</sup> La CPU 1212C ne comporte pas de sorties Q0.6 et Q0.7. Ces sorties ne peuvent donc pas être utilisées dans la CPU 1212C.
- <sup>3</sup> Ce tableau s'applique aux fonctions PTO des CPU 1211C, CPU 1212C, CPU 1214C et CPU 1215C.

## Interface d'entraînement

Pour la commande de mouvement, vous pouvez configurer facultativement une interface d'entraînement pour "Entraînement libéré" et "Entraînement prêt". Lors de l'utilisation de l'interface d'entraînement, la sortie TOR pour la validation de l'entraînement et l'entrée TOR pour "Entraînement prêt" peuvent être sélectionnées librement.

---

### Remarque

Le firmware prendra le contrôle via les sorties d'impulsions et de sens correspondantes si la sortie PTO (Pulse Train Output) a été sélectionnée et affectée à un axe.

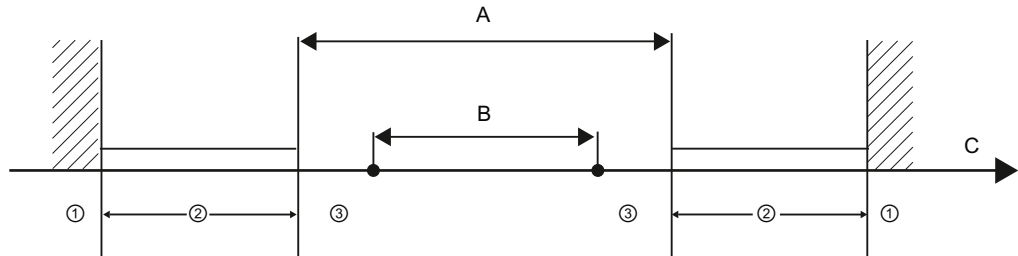
Avec cette mainmise de la fonction de commande, le lien entre la mémoire image et la sortie physique est également rompu. Alors que l'utilisateur peut écrire dans la mémoire image des sorties d'impulsions et de sens via le programme utilisateur ou la table de visualisation, ces valeurs ne sont jamais transférées à la sortie physique. En conséquence, il n'est pas non plus possible de surveiller la sortie physique via le programme utilisateur ou la table de visualisation. Les informations lues ne font que refléter la valeur de la mémoire image et ne correspondent en rien à l'état actuel de la sortie physique.

Pour toutes les autres sorties CPU qui ne sont pas utilisées en permanence par le firmware CPU, l'état de la sortie physique peut être commandé ou surveillé comme d'habitude via la mémoire image.

---

### 9.3.4.2 Fins de course matériels et logiciels pour la commande de mouvement

Utilisez les fins de course matériels et logiciels pour limiter la plage de déplacement autorisée et la plage de travail de votre axe.



- |   |                                                 |   |                                           |
|---|-------------------------------------------------|---|-------------------------------------------|
| ① | Butée mécanique                                 | A | Plage de déplacement autorisée pour l'axe |
| ② | Fins de course matériels inférieur et supérieur | B | Plage de travail de l'axe                 |
| ③ | Fins de course logiciels inférieur et supérieur | C | Distance                                  |

Il faut activer les fins de course matériels et logiciels avant de les utiliser dans la configuration ou dans le programme utilisateur. Les fins de course logiciels sont actifs uniquement après le référencement de l'axe.

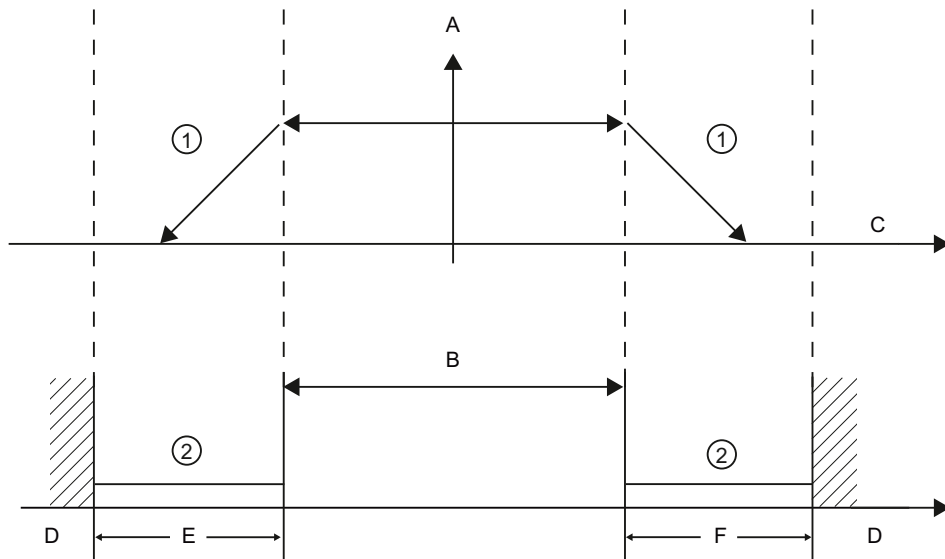
### Fins de course matériels

Les fins de course matériels définissent la plage de déplacement maximale de l'axe. Les fins de course matériels sont des commutateurs physiques qui doivent être raccordés à des entrées aptes aux alarmes de la CPU. Utilisez uniquement des fins de course matériels qui restent commutés en permanence après avoir été accostés. Cet état de commutation ne peut être annulé qu'après un retour dans la plage de déplacement autorisée.

Tableau 9- 49 Entrées disponibles pour les générateurs d'impulsions

Description	RPS	LIM-	LIM+
PTO 0			
E/S intégrées		10.0 - 11.5	
E/S du SB		14.0 - 14.3	
PTO 1			
E/S intégrées		10.0 - 11.5	
E/S du SB		14.0 - 14.3	
PTO 2			
E/S intégrées		10.0 - 11.5	
E/S du SB		14.0 - 14.3	
PTO 3			
E/S intégrées		10.0 - 11.5	
E/S du SB		14.0 - 14.3	

Lors de l'accostage des fins de course matériels, l'axe freine jusqu'à s'immobiliser avec la décélération d'urgence configurée. La décélération d'urgence configurée doit être suffisante pour arrêter l'axe de manière fiable avant la butée mécanique. Le schéma suivant montre le comportement de l'axe après qu'il a accosté les fins de course matériels.



- ① L'axe freine jusqu'à s'immobiliser avec la décélération d'urgence configurée.
- ② Plage dans laquelle les fins de course matériels signalent l'état "accosté".
- A [vitesse]
- B Plage de déplacement autorisée
- C Distance
- D Butée mécanique
- E Fin de course matériel inférieur
- F Fin de course matériel supérieur

**ATTENTION**

Si le temps de filtre pour une voie d'entrée TOR est modifié par rapport à un paramétrage précédent, une nouvelle valeur d'entrée de niveau "0" peut devoir être présente pendant une durée cumulée allant jusqu'à 20,0 ms pour que le filtre réagisse pleinement aux nouvelles entrées. Pendant ce temps, les événements d'impulsion "0" courts de moins de 20,0 ms peuvent ne pas être détectés ni comptés.

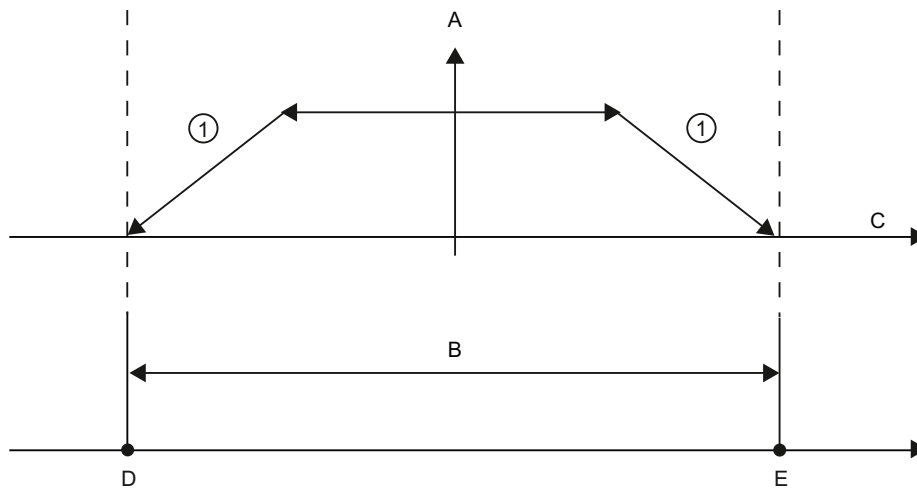
Cette modification des temps de filtre peut provoquer un fonctionnement inattendu des machines ou du processus, pouvant entraîner la mort, des blessures graves et des dommages matériels importants.

Pour garantir la prise en compte immédiate d'un nouveau temps de filtre, mettez la CPU hors tension puis sous tension.

### Fins de course logiciels

Les fins de course logiciels limitent la plage de travail de l'axe. Ils doivent se trouver en deçà des fins de course matériels par rapport à la plage de déplacement. Comme les positions des fins de course logiciels sont flexibles, la plage de travail de l'axe peut être restreinte sur une base individuelle en fonction du profil de déplacement en cours. Contrairement aux fins de course matériels, les fins de course logiciels sont réalisés exclusivement au moyen du logiciel et ne nécessitent pas leurs propres commutateurs.

Si des fins de course logiciels sont activés, un déplacement actif est arrêté à la position du fin de course logiciel. L'axe freine avec la décélération configurée. Le schéma suivant montre le comportement de l'axe lorsqu'il atteint les fins de course logiciels.



- ① L'axe freine jusqu'à s'immobiliser avec la décélération configurée.
- A [vitesse]
- B Plage de travail
- C Distance
- D Fin de course logiciel inférieur
- E Fin de course logiciel supérieur

Utilisez des fins de course matériels supplémentaires si une butée mécanique est placée après les fins de course logiciels et qu'il y a risque de dommages mécaniques.

### Informations supplémentaires

Votre programme utilisateur peut annuler les limites de positionnement matérielles ou logicielles en activant ou en désactivant à la fois la fonctionnalité de limite matérielle et logicielle. La sélection se fait dans le DB de l'axe.

- Pour activer ou désactiver la fonctionnalité de limite matérielle, accédez à la variable "Active" (Bool) dans le chemin de DB "<nom d'axe>/Config/PositonLimits\_HW". L'état de la variable "Active" active ou désactive l'utilisation de limites de positionnement matérielles.
- Pour activer ou désactiver la fonctionnalité de limite de positionnement logicielle, accédez à la variable "Active" (Bool) dans le chemin de DB "<nom d'axe>/Config/Position Limits\_SW". L'état de la variable "Active" active ou désactive les limites de positionnement logicielles.

Vous pouvez également modifier les limites de positionnement logicielles avec votre programme utilisateur (par exemple, pour offrir une plus grande souplesse pour la configuration d'une machine ou réduire la durée de remplacement d'une machine). Votre programme utilisateur peut écrire de nouvelles valeurs dans les variables "MinPosition" et "MaxPosition" (unités physiques en format Real) dans le DB <nom d'axe>/Config/PositionLimits\_SW".

#### 9.3.4.3 Référencement

Le référencement consiste à établir une correspondance entre les coordonnées de l'axe et la position physique réelle de l'entraînement (si l'entraînement est actuellement en position x, l'axe sera ajusté pour être en position x). Pour les axes commandés en position, les entrées et les affichages concernant la position se réfèrent exactement à ces coordonnées de l'axe.

---

#### Remarque

La correspondance entre les coordonnées de l'axe et la situation réelle est extrêmement importante. Cette étape est nécessaire pour garantir que la position cible absolue de l'axe est également atteinte exactement avec l'entraînement.

---

L'instruction MC\_Home déclenche le référencement de l'axe.

Il existe quatre fonctions de référencement différentes. Les deux premières fonctions permettent à l'utilisateur de définir la position en cours de l'axe et les deux autres à positionner l'axe par rapport à un capteur de référencement.

- Mode 0 - Référencement direct absolu : Lorsqu'il est activé, ce mode indique à l'axe exactement où il est. Il donne à la variable de position interne la valeur de l'entrée Position de l'instruction de référencement. Ce mode est utilisé pour le calibrage et la configuration des machines.

La position de l'axe est définie sans tenir compte de la came de référence. Les déplacements en cours ne sont pas annulés. La valeur du paramètre d'entrée Position de l'instruction MC\_Home est immédiatement prise comme point de référence de l'axe. Pour affecter le point de référence à une position mécanique précise, l'axe doit être immobile à cette position pendant l'opération de référencement.

- Mode 1 - Référencement direct relatif : Lorsqu'il est activé, ce mode utilise la variable de position interne à laquelle il ajoute la valeur de l'entrée Position de l'instruction de référencement. Ce mode sert typiquement à prendre en compte le décalage d'une machine.

La position de l'axe est définie sans tenir compte de la came de référence. Les déplacements en cours ne sont pas annulés. La position de l'axe après le référencement est définie comme suit : nouvelle position de l'axe = position en cours de l'axe + valeur du paramètre Position de l'instruction MC\_Home

- Mode 2 - Référencement passif : Lorsque l'axe se déplace et franchit la came de référence, la position en cours est définie comme position de référence. Cette fonction permet de prendre en compte l'usure normale des machines et le jeu entre dents et d'éviter ainsi la compensation manuelle de l'usure. L'entrée Position de l'instruction de référencement s'ajoute, comme précédemment, à la position indiquée par la came de référence, ce qui permet un décalage aisée de la position de référence.

Lors du référencement passif, l'instruction MC\_Home n'exécute aucune prise de référence. L'accostage de la came de référence doit être réalisé par l'utilisateur au moyen d'instructions de commande de mouvement. L'axe est référencé conformément à la configuration lorsque la came de référence est détectée. Les déplacements en cours ne sont pas annulés au démarrage du référencement passif.

- Mode 3 - Référencement actif : Ce mode constitue la méthode de référencement de l'axe la plus précise. Le sens et la vitesse de déplacement initiaux sont configurés dans les paramètres avancés de référencement de la configuration d'objet technologique. Cela dépend de la configuration de la machine. Il est également possible de déterminer si le front montant ou le front descendant du signal de came de référence constitue la position de référence. Pratiquement tous les capteurs ont une plage active et si la position Régime permanent activé était utilisée comme signal de référence, il y aurait la possibilité d'une erreur dans la position de référence puisque la plage active du signal activé couvrirait une plage de distance. L'utilisation du front montant ou du front descendant de ce signal permet d'obtenir des résultats beaucoup plus précis pour la position de référence. Comme avec tous les autres modes, la valeur de l'entrée Position de l'instruction de référencement est ajoutée à la position référencée matérielle.

En mode de référencement actif, l'instruction MC\_Home exécute la prise de référence requise. L'axe est référencé conformément à la configuration lorsque la came de référence est détectée. Les déplacements en cours sont annulés.

Les modes 0 et 1 ne nécessitent aucun déplacement de l'axe. Ils servent typiquement lors de la configuration et du calibrage. Les modes 2 et 3 nécessitent un déplacement de l'axe et franchissent un capteur qui est configuré en tant que came de référence dans l'objet technologique Axe. Le point de référence peut être placé dans la zone de travail de l'axe ou hors de la zone de travail normale mais à l'intérieur de la plage de déplacement.

### Configuration des paramètres de référencement

Configurez les paramètres pour le référencement actif et passif dans la fenêtre de configuration "Référencement". La méthode de référencement est définie à l'aide du paramètre d'entrée "Mode" de l'instruction de commande de mouvement. Ici, Mode = 2 signifie référencement passif et Mode = 3 signifie référencement actif.

<b>IMPORTANT</b>
<p>Prenez l'une des mesures suivantes pour garantir que la machine n'atteint pas une butée mécanique en cas d'inversion de sens :</p> <ul style="list-style-type: none"> <li>• Maintenez une vitesse d'accostage faible.</li> <li>• Augmentez l'accélération/décélération configurée.</li> <li>• Augmentez la distance entre le fin de course matériel et la butée mécanique.</li> </ul>

Tableau 9- 50 Paramètres de configuration pour le référencement de l'axe

<b>Paramètre</b>	<b>Description</b>
Entrée de la came de référence (référencement actif et passif)	<p>Sélectionnez, dans la liste déroulante, l'entrée TOR pour la came de référence. L'entrée doit pouvoir émettre des alarmes. Les entrées intégrées de la CPU et les entrées d'un Signal Board inséré peuvent être sélectionnées comme entrées pour la came de référence.</p> <p>Le temps de filtre par défaut pour les entrées TOR est de 6,4 ms. Lorsqu'on utilise les entrées TOR comme came de référence, il peut se produire des décélérations indésirables et donc des imprécisions. Selon la vitesse d'approche et l'étendue de la came de référence, le point de référence peut ne pas être détecté. Il est possible de régler le temps de filtre sous "Filtre d'entrée" dans la configuration d'appareil des entrées TOR.</p> <p>Le temps de filtre indiqué doit être inférieur à la durée du signal d'entrée au niveau de la came de référence.</p>
Inversion automatique du sens lorsque les fins de course matériels sont atteints (référencement actif uniquement)	<p>Cochez cette case si vous souhaitez utiliser les fins de course matériels comme came d'inversion pour la prise de référence. Les fins de course matériels doivent être configurés et activés pour l'inversion de sens.</p> <p>Si le fin de course matériel est atteint pendant que le référencement est actif, l'axe freine avec la décélération configurée (et non avec la décélération d'urgence) et effectue une inversion de sens. La came de référence est ensuite recherchée en sens inverse.</p> <p>Si l'inversion de sens n'est pas active et que l'axe atteint le fin de course matériel pendant que le référencement est actif, la prise de référence est interrompue avec une erreur et l'axe est freiné avec la décélération d'urgence.</p>

Paramètre	Description
Sens d'accostage (référencement actif et passif)	Avec la sélection de sens, vous définissez le sens d'accostage utilisé pendant le référencement actif pour rechercher la came de référence, ainsi que le sens de référencement. Le sens de référencement indique le sens de déplacement utilisé par l'axe pour accoster le côté configuré de la came de référence afin d'effectuer le référencement.
Came de référence (référencement actif et passif)	<ul style="list-style-type: none"> <li>• Référencement actif : Indiquez si l'axe doit être référencé du côté gauche ou droit de la came de référence. Selon la position de départ de l'axe et la configuration des paramètres de référencement, la séquence de prise de référence peut différer du schéma dans la fenêtre de configuration.</li> <li>• Référencement passif : Avec le référencement passif, les mouvements de déplacement aux fins de référencement doivent être réalisés par l'utilisateur à l'aide de commandes de mouvement. Le côté de la came de référence sur lequel se fait le référencement dépend des facteurs suivants : <ul style="list-style-type: none"> <li>- Configuration du sens d'accostage</li> <li>- Configuration de la came de référence</li> <li>- Sens de déplacement actif pendant le référencement passif</li> </ul> </li> </ul>
Vitesse d'attaque (référencement actif uniquement)	Indiquez la vitesse à laquelle la came de référence est recherchée pendant la prise de référence. Valeurs limites (indépendantes de l'unité utilisateur sélectionnée) : vitesse de démarrage/d'arrêt ≤ vitesse d'attaque ≤ vitesse maximale
Vitesse d'accostage (référencement actif uniquement)	Indiquez la vitesse à laquelle l'axe doit accoster la came de référence pour réaliser le référencement. Valeurs limites (indépendantes de l'unité utilisateur sélectionnée) : vitesse de démarrage/d'arrêt ≤ vitesse d'accostage ≤ vitesse maximale
Décalage de la position de référence (référencement actif uniquement)	Si le point de référence désiré diffère de la position de la came de référence, vous pouvez indiquer le décalage de la position de référence dans ce champ. Si cette valeur est différente de 0, l'axe exécute les actions suivantes après le référencement au niveau de la came de référence : <ol style="list-style-type: none"> <li>1. Déplacez l'axe de la valeur du décalage de la position de référence à la vitesse d'accostage.</li> <li>2. Lorsque la position du décalage de la position de référence est atteinte, la position de l'axe prend la valeur de la position de référence absolue. La position de référence absolue est indiquée via le paramètre "Position" de l'instruction de commande de mouvement "MC_Home".</li> </ol> Valeurs limites (indépendantes de l'unité utilisateur sélectionnée) : -1,0e12 ≤ décalage de la position de référence ≤ 1,0e12



Tableau 9- 51 Facteurs affectant le référencement

Facteurs déterminants :			Résultat :
Configuration du sens d'accostage	Configuration de la came de référence	Sens de déplacement en cours	Référencement sur la came de référence
Positif	Côté gauche (négatif)	Sens positif	Gauche
		Sens négatif	Droite
Positif	Côté droit (positif)	Sens positif	Droite
		Sens négatif	Gauche
Négatif	Côté gauche (négatif)	Sens positif	Droite
		Sens négatif	Gauche
Négatif	Côté droit (positif)	Sens positif	Gauche
		Sens négatif	Droite

### Séquence pour le référencement actif

Vous démarrez le référencement actif avec l'instruction de commande de mouvement "MC\_Home" (paramètre d'entrée Mode = 3). Dans ce cas, le paramètre d'entrée Position" indique les coordonnées absolues du point de référence. Vous pouvez également lancer le référencement actif dans le panneau de commande à des fins de test.

Le schéma suivant montre un exemple de courbe caractéristique pour une prise de référence active avec les paramètres de configuration suivants :

- sens d'accostage = sens d'accostage positif
- came de référence = côté droit (positif)
- valeur du décalage de la position de référence > 0

Tableau 9- 52 Caractéristiques de vitesse du référencement MC

Fonctionnement		Remarques	
		A	Vitesse d'attaque
		B	Vitesse d'accostage
		C	Coordonnée de la position de référence
		D	Décalage de la position de référence
①	Phase de recherche (segment bleu de la courbe) : Au démarrage du référencement actif, l'axe accélère jusqu'à la vitesse d'attaque configurée et recherche la came de référence à cette vitesse.		
②	Accostage du point de référence (segment rouge de la courbe) : Une fois la came de référence détectée, l'axe freine et exécute une inversion de sens pour être référencé du côté configuré de la came de référence à la vitesse d'accostage configurée.		
③	Déplacement vers la position du point de référence (segment vert de la courbe) : Après son référencement à la came de référence, l'axe se déplace vers les coordonnées du point de référence à la vitesse d'accostage. Lorsqu'il atteint les coordonnées du point de référence, l'axe est arrêté à la valeur de position indiquée dans le paramètre d'entrée Position de l'instruction MC_Home.		

**Remarque**

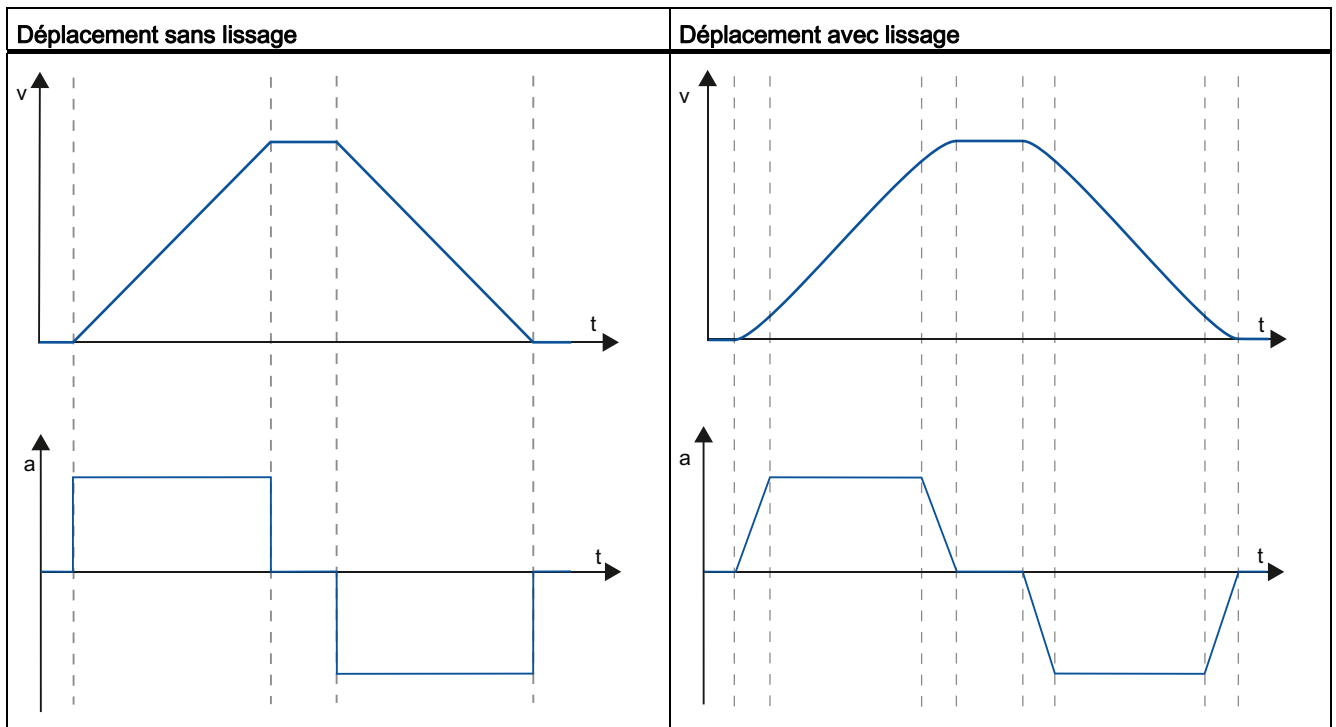
Si la recherche de référencement ne fonctionne pas comme vous le prévoyiez, vérifiez les entrées affectées aux fins de course matériels ou à la came de référence. Les alarmes sur front de ces entrées ont peut-être été désactivées dans la configuration d'appareil.

Examinez les données de configuration de l'objet technologique Axe concerné pour voir, le cas échéant, les entrées qui sont affectées à "HW Low Limit Switch Input", "HW High Limit Switch Input" et "Input reference point switch". Puis, ouvrez la configuration d'appareil pour la CPU et examinez chacune des entrées affectées. Vérifiez que les options "Activer la détection de front montant" et "Activer la détection de front descendant" sont toutes deux sélectionnées. Si elles ne le sont pas, effacez les entrées concernées dans la configuration de l'axe puis sélectionnez-les à nouveau.

### 9.3.4.4 Limitation d'à-coup

La limitation d'à-coup permet de réduire les contraintes auxquelles sont soumises les pièces mécaniques pendant une rampe d'accélération et de décélération. La valeur pour l'accélération et la décélération n'est pas modifiée soudainement lorsque la limitation est active ; elle est adaptée lors d'une phase de transition. La figure ci-dessous montre la courbe de vitesse et la courbe d'accélération avec et sans limitation d'à-coup.

Tableau 9- 53 Limitation d'à-coup



La limitation d'à-coup donne un profil de vitesse "lissé" au mouvement de l'axe. Cela garantit le démarrage et le freinage en douceur d'un convoyeur à bande, par exemple.

### 9.3.5 Mise en service

#### Fonction de diagnostic "Bits d'état et d'erreur"

La fonction de diagnostic "Bits d'état et d'erreur" vous permet de surveiller les principaux messages d'état et d'erreur de l'axe. L'affichage de la fonction de diagnostic est disponible en mode en ligne dans les modes de fonctionnement "Commande manuelle" et "Commande automatique" lorsque l'axe est actif.

Tableau 9- 54 Etat de l'axe

Etat	Description
Libéré	L'axe est libéré et prêt à être commandé via des tâches de commande de mouvement. (Variable de l'objet technologique : <nom d'axe>.StatusBits.Enable)
Référencé	L'axe est référencé et est en mesure d'exécuter des tâches de positionnement absolu de l'instruction "MC_MoveAbsolute". Il n'est pas nécessaire que l'axe soit référencé pour le positionnement relatif. Situations spéciales : <ul style="list-style-type: none"> <li>• Pendant le référencement actif, l'état est FAUX.</li> <li>• Si un axe référencé est soumis à un référencement passif, l'état est mis à VRAI pendant le référencement passif.</li> </ul> (Variable de l'objet technologique : <nom d'axe>.StatusBits.HomingDone)
Erreur	Une erreur s'est produite dans l'objet technologique Axe. En mode automatique, des informations détaillées sur l'erreur sont fournies par les paramètres ErrorID et ErrorInfo des instructions de commande de mouvement. En mode manuel, le champ "Dernière erreur" du panneau de commande affiche des informations détaillées sur la cause de l'erreur. (Variable de l'objet technologique : <nom d'axe>.StatusBits.Error)
Panneau de commande actif	Le mode "Commande manuelle" a été activé dans le panneau de commande. Le panneau de commande est en charge de la commande de l'objet technologique Axe. L'axe ne peut pas être piloté à partir du programme utilisateur. (Variable de l'objet technologique : <nom d'axe>.StatusBits.ControlPanelActive)

Tableau 9- 55 Etat de l'entraînement

Etat	Description
Entraînement prêt	L'entraînement est prêt à fonctionner. (Variable de l'objet technologique : <nom d'axe>.StatusBits.DriveReady)
Erreur	L'entraînement a signalé une erreur après défaillance de son signal Prêt. (Variable de l'objet technologique : <nom d'axe>.ErrorBits.DriveFault)

Tableau 9- 56 Etat du déplacement de l'axe

<b>Etat</b>	<b>Description</b>
Immobilisation	L'axe est immobilisé. (Variable de l'objet technologique : <nom d'axe>.StatusBits.StandStill)
Accélération	L'axe accélère. (Variable de l'objet technologique : <nom d'axe>.StatusBits.Acceleration)
Vitesse constante	L'axe se déplace à vitesse constante. (Variable de l'objet technologique : <nom d'axe>.StatusBits.ConstantVelocity)
Décélération	L'axe décélère (freine). (Variable de l'objet technologique : <nom d'axe>.StatusBits.Deceleration)

Tableau 9- 57 Etat du mode de déplacement

<b>Etat</b>	<b>Description</b>
Positionnement	L'axe exécute une tâche de positionnement de l'instruction "MC_MoveAbsolute" ou "MC_MoveRelative" ou du panneau de commande. (Variable de l'objet technologique : <nom d'axe>.StatusBits.PositioningCommand)
Prescription de vitesse	L'axe exécute à vitesse prescrite une tâche de l'instruction "MC_MoveVelocity" ou "MC_MoveJog" ou du panneau de commande. (Variable de l'objet technologique : <nom d'axe>.StatusBits.SpeedCommand)
Référencement	L'axe exécute une tâche de référencement de l'instruction "MC_Home" ou du panneau de commande. (Variable de l'objet technologique : <nom d'axe>.StatusBits.Homing)

Tableau 9- 58 Bits d'erreur

<b>Erreur</b>	<b>Description</b>
Fin de course logiciel min. atteint	Le fin de course logiciel inférieur a été atteint. (Variable de l'objet technologique : <nom d'axe>.ErrorBits.SwLimitMinReached)
Fin de course logiciel min. dépassé	Le fin de course logiciel inférieur a été dépassé. (Variable de l'objet technologique : <nom d'axe>.ErrorBits.SwLimitMinExceeded)
Fin de course logiciel max. atteint	Le fin de course logiciel supérieur a été atteint. (Variable de l'objet technologique : <nom d'axe>.ErrorBits.SwLimitMaxReached)
Fin de course logiciel max. dépassé	Le fin de course logiciel supérieur a été dépassé. (Variable de l'objet technologique : <nom d'axe>.ErrorBits.SwLimitMaxExceeded)
Fin de course matériel négatif	Le fin de course matériel inférieur a été accosté. (Variable de l'objet technologique : <nom d'axe>.ErrorBits.HwLimitMin)
Fin de course matériel positif	Le fin de course matériel supérieur a été accosté. (Variable de l'objet technologique : <nom d'axe>.ErrorBits.HwLimitMax)
PTO et HSC déjà utilisés	Un deuxième axe, libéré via "MC_Power", utilise les mêmes PTO et HSC. (Variable de l'objet technologique : <nom d'axe>.ErrorBits.HwUsed)

Erreur	Description
Erreur de configuration	L'objet technologique Axe a été mal configuré ou des données de configuration éditables ont été modifiées de manière incorrecte pendant l'exécution du programme utilisateur. (Variable de l'objet technologique : <nom d'axe>.ErrorBits.ConfigFault)
Erreur générale	Une erreur interne s'est produite. (Variable de l'objet technologique : <nom d'axe>.ErrorBits.SystemFault)

### Fonction de diagnostic "Etat du déplacement"

La fonction de diagnostic "Etat du déplacement" vous permet de surveiller l'état de déplacement de l'axe. L'affichage de la fonction de diagnostic est disponible en mode en ligne dans les modes de fonctionnement "Commande manuelle" et "Commande automatique" lorsque l'axe est actif.

Tableau 9- 59 Etat du déplacement

Etat	Description
Position cible	Le champ "Position cible" affiche la position cible actuelle d'une tâche de positionnement active de l'instruction "MC_MoveAbsolute" ou "MC_MoveRelative" ou du panneau de commande. La valeur de la position cible n'est valable que pendant l'exécution d'une tâche de positionnement. (Variable de l'objet technologique : <nom d'axe>.MotionStatus.TargetPosition)
Position actuelle	Le champ "Position actuelle" affiche la position actuelle de l'axe. Si l'axe n'est pas référencé, la valeur indique la position par rapport à la position de validation de l'axe. (Variable de l'objet technologique : <nom d'axe>.MotionStatus.Position)
Vitesse actuelle	Le champ "Vitesse actuelle" affiche la vitesse actuelle de l'axe. (Variable de l'objet technologique : <nom d'axe>.MotionStatus.Velocity)

Tableau 9- 60 Limites dynamiques

Limite dynamique	Description
Vitesse	Le champ "Vitesse" affiche la vitesse maximale configurée pour l'axe. (Variable de l'objet technologique : <nom d'axe>.Config.DynamicLimits.MaxVelocity)
Accélération	Le champ "Accélération" affiche l'accélération maximale configurée pour l'axe. (Variable de l'objet technologique : <nom d'axe>.Config.DynamicDefaults.Acceleration)
Décélération	Le champ "Décélération" affiche la décélération maximale configurée pour l'axe. (Variable de l'objet technologique : <nom d'axe>.Config.DynamicDefaults.Deceleration)

## 9.3.6 Suivi des commandes actives

### 9.3.6.1 Suivi des instructions MC avec paramètre de sortie "Done"

Les instructions de commande de mouvement avec paramètre de sortie "Done" sont démarrées via le paramètre d'entrée "Execute" et se concluent de manière définie (par exemple, la réussite du référencement avec l'instruction "MC\_Home" ). La tâche est achevée et l'axe est à l'arrêt.

- Le paramètre de sortie "Done" a la valeur VRAI si la tâche s'est achevée correctement.
- Les paramètres de sortie "Busy", "CommandAborted" et "Error" signalent respectivement que la tâche est encore en cours de traitement, qu'elle a été annulée ou qu'une erreur s'est produite. L'instruction "MC\_Reset" ne peut pas être annulée et ne présente donc pas de paramètre de sortie "CommandAborted".
  - Pendant l'exécution de la tâche de commande de mouvement, le paramètre de sortie "Busy" indique la valeur VRAI. Si la tâche est achevée, a été annulée ou a été arrêtée en raison d'une erreur, le paramètre de sortie "Busy" prend la valeur FAUX. Cette modification a lieu quel que soit le signal du paramètre d'entrée "Execute".
  - Les paramètres de sortie "Done", "CommandAborted" et "Error" prennent la valeur VRAI pendant au moins un cycle. Ces messages d'état sont rémanents pendant que le paramètre d'entrée "Execute" a la valeur VRAI.

Les tâches des instructions de commande de mouvement suivantes ont une fin définie :

- MC\_Reset
- MC\_Home
- MC\_Halt
- MC\_MoveAbsolute
- MC\_MoveRelative

Le comportement des bits d'état est représenté ci-après pour divers types de situations.

- Le premier exemple montre le comportement de l'axe pour une tâche achevée. Si la tâche de commande de mouvement a été exécutée complètement à son achèvement, le paramètre de sortie "Done" prend la valeur VRAI. L'état logique du paramètre d'entrée "Execute" influe sur la durée de signalisation dans le paramètre de sortie "Done".
- Le deuxième exemple montre le comportement de l'axe pour une tâche annulée. Si la tâche de commande de mouvement est annulée pendant son exécution, le paramètre de sortie "CommandAborted" prend la valeur VRAI. L'état logique du paramètre d'entrée "Execute" influe sur la durée de signalisation dans le paramètre de sortie "CommandAborted".
- Le troisième exemple montre le comportement de l'axe si une erreur se produit. Si une erreur se produit pendant l'exécution de la tâche de commande de mouvement, le paramètre de sortie "Error" prend la valeur VRAI. L'état logique du paramètre d'entrée "Execute" influe sur la durée de signalisation dans le paramètre de sortie "Error".

Tableau 9- 61 Exemple 1 : Exécution complète de la tâche

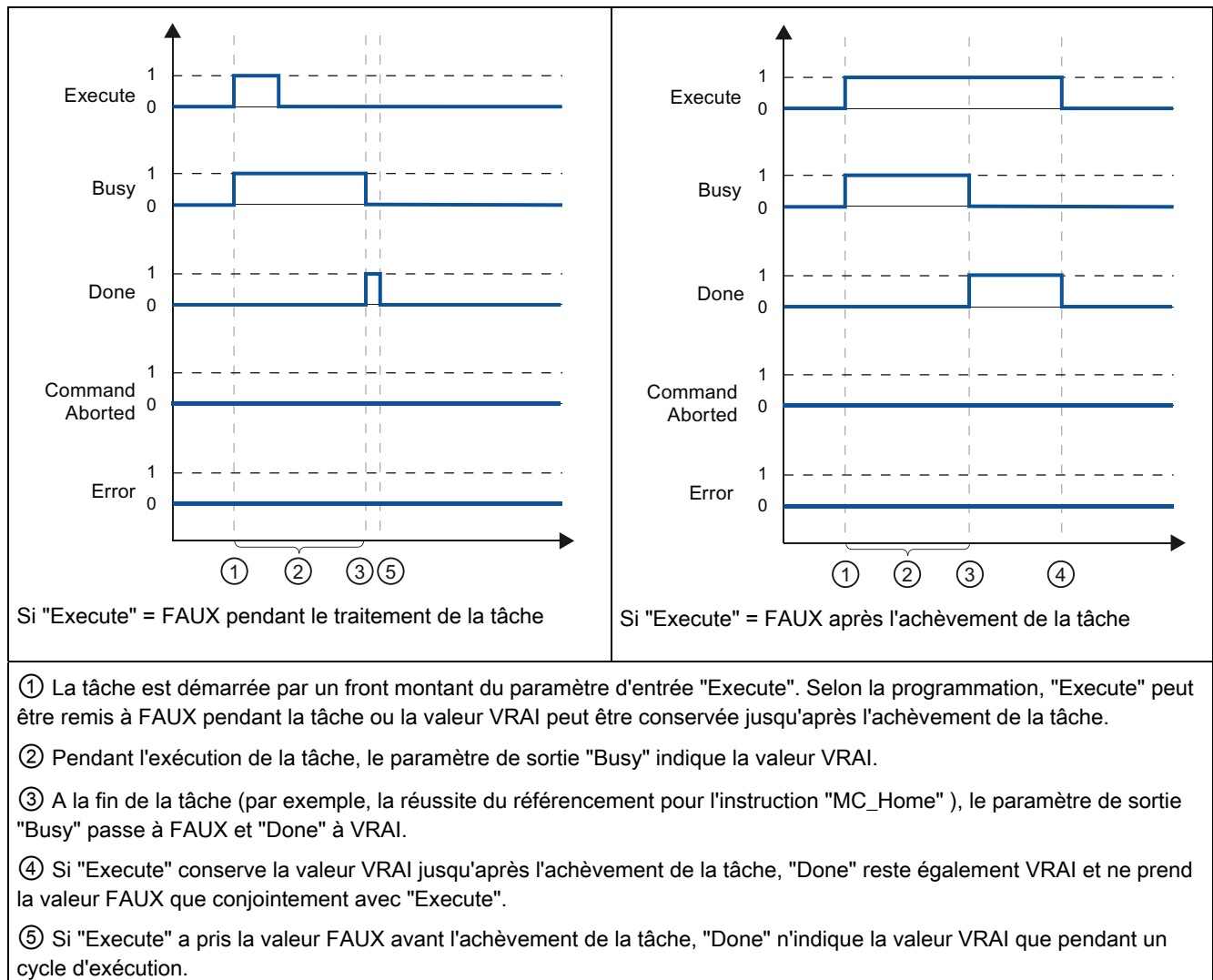




Tableau 9- 62 Exemple 2 : Annulation de la tâche

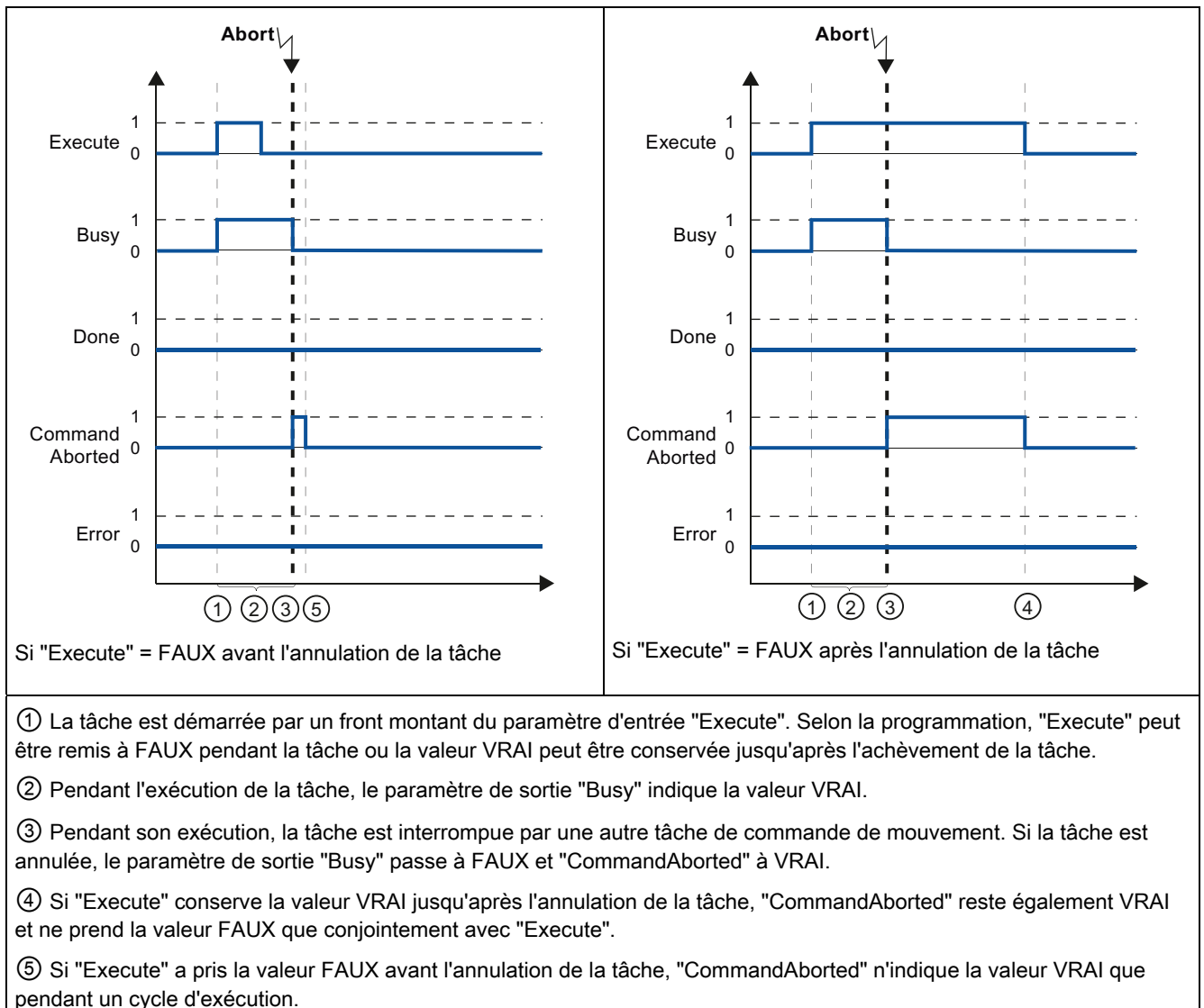
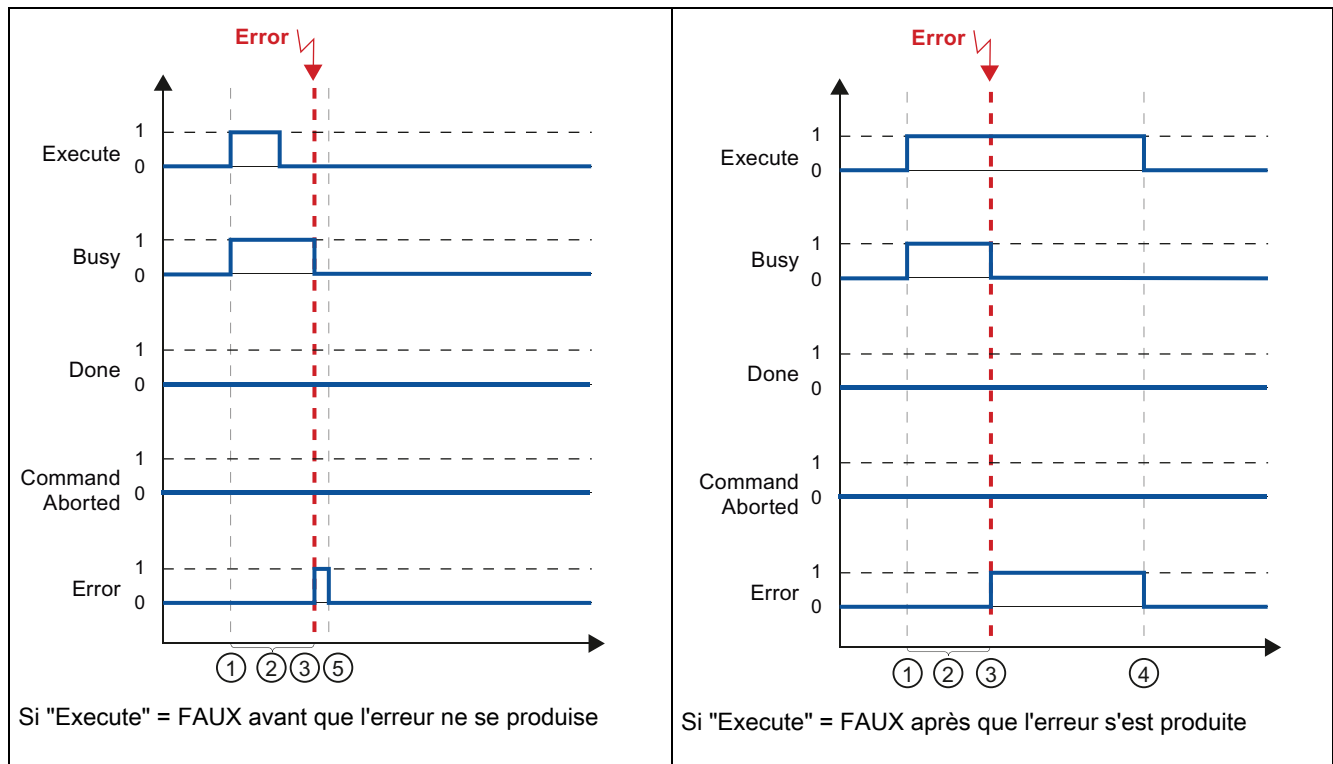


Tableau 9- 63 Exemple 3 : Erreur pendant l'exécution de la tâche



- ① La tâche est démarrée par un front montant du paramètre d'entrée "Execute". Selon la programmation, "Execute" peut être remis à FAUX avant l'apparition de l'erreur ou la valeur VRAI peut être conservée jusqu'après l'apparition de l'erreur.
- ② Pendant l'exécution de la tâche, le paramètre de sortie "Busy" indique la valeur VRAI.
- ③ Une erreur se produit pendant l'exécution de la tâche. Lorsque l'erreur se produit, le paramètre de sortie "Busy" passe à FAUX et "Error" à VRAI.
- ④ Si "Execute" conserve la valeur VRAI jusqu'après l'apparition de l'erreur, "Error" reste également VRAI et ne prend la valeur FAUX que conjointement avec "Execute".
- ⑤ Si "Execute" a pris la valeur FAUX avant l'apparition de l'erreur, "Error" n'indique la valeur VRAI que pendant un cycle d'exécution.

### 9.3.6.2 Suivi de l'instruction MC\_Velocity

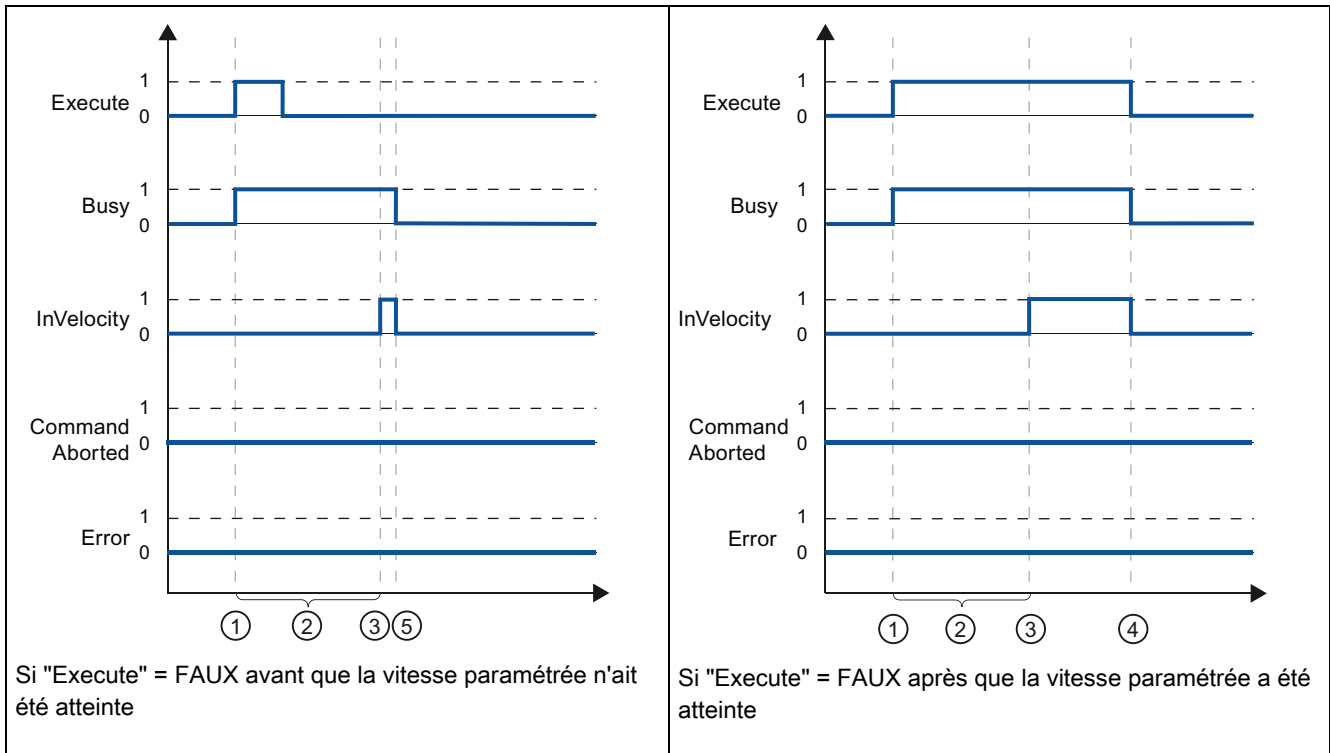
Les tâches de l'instruction de commande de mouvement "MC\_MoveVelocity" s'exécutent de manière constante à la vitesse indiquée.

- Les tâches de l'instruction "MC\_MoveVelocity" n'ont pas de fin définie. L'objectif de la tâche est rempli lorsque la vitesse paramétrée est atteinte pour la première fois et que l'axe se déplace à vitesse constante. Lorsque la vitesse paramétrée est atteinte, le paramètre de sortie "InVelocity" prend la valeur VRAI.
- La tâche est achevée lorsque la vitesse paramétrée a été atteinte et que le paramètre d'entrée "Execute" a pris la valeur FAUX. Toutefois, le mouvement de l'axe n'est pas encore achevé à l'achèvement de la tâche. Son mouvement peut, par exemple, être arrêté à l'aide de la tâche "MC\_Halt".
- Les paramètres de sortie "Busy", "CommandAborted" et "Error" signalent respectivement que la tâche est encore en cours de traitement, qu'elle a été annulée ou qu'une erreur s'est produite.
  - Pendant l'exécution de la tâche de commande de mouvement, le paramètre de sortie "Busy" indique la valeur VRAI. Si la tâche est achevée, a été annulée ou a été arrêtée en raison d'une erreur, le paramètre de sortie "Busy" prend la valeur FAUX. Cette modification a lieu quel que soit le signal du paramètre d'entrée "Execute".
  - Les paramètres de sortie "InVelocity", "CommandAborted" et "Error" prennent la valeur VRAI pendant au moins un cycle lorsque leur condition est remplie. Ces messages d'état sont rémanents pendant que le paramètre d'entrée "Execute" a la valeur VRAI.

Le comportement des bits d'état est représenté ci-après pour divers types de situations.

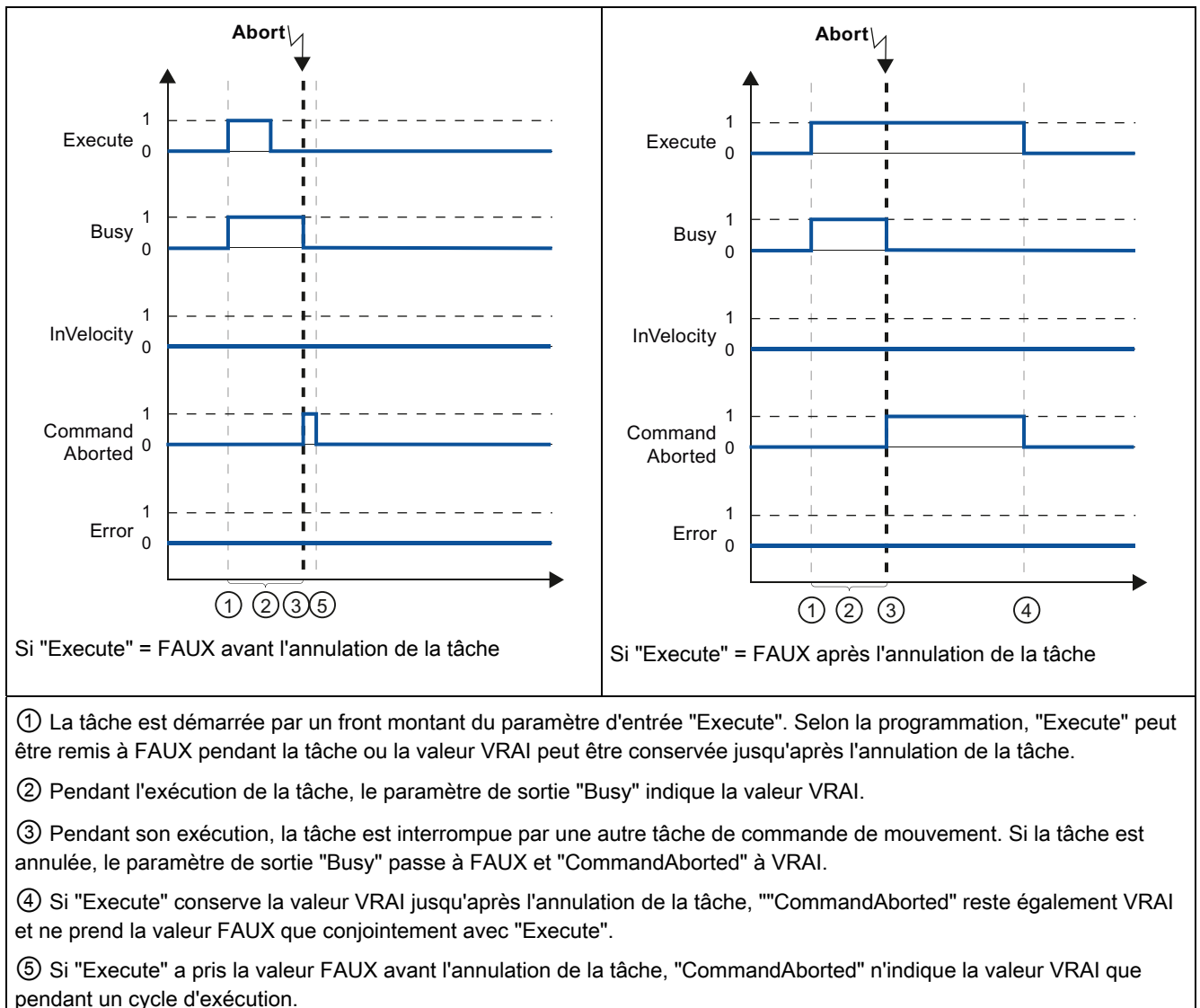
- Le premier exemple montre ce qui se passe lorsque l'axe atteint la vitesse paramétrée. Si la tâche de commande de mouvement a été exécutée au moment où la vitesse paramétrée est atteinte, le paramètre de sortie "InVelocity" prend la valeur VRAI. L'état logique du paramètre d'entrée "Execute" influe sur la durée de signalisation dans le paramètre de sortie "InVelocity".
- Le deuxième exemple montre ce qui se passe si la tâche est annulée avant que la vitesse paramétrée n'ait été atteinte. Si la tâche de commande de mouvement est annulée avant que la vitesse paramétrée n'ait été atteinte, le paramètre de sortie "CommandAborted" prend la valeur VRAI. L'état logique du paramètre d'entrée "Execute" influe sur la durée de signalisation dans le paramètre de sortie "CommandAborted".
- Le troisième exemple montre le comportement de l'axe si une erreur survient avant que la vitesse paramétrée n'ait été atteinte. Si une erreur se produit pendant l'exécution de la tâche de commande de mouvement avant que la vitesse paramétrée n'ait été atteinte, le paramètre de sortie "Error" prend la valeur VRAI. L'état logique du paramètre d'entrée "Execute" influe sur la durée de signalisation dans le paramètre de sortie "Error".

Tableau 9- 64 Exemple 1 : La vitesse paramétrée est atteinte



- ① La tâche est démarrée par un front montant du paramètre d'entrée "Execute". Selon la programmation, "Execute" peut être remis à FAUX avant que la vitesse paramétrée n'ait été atteinte ou bien seulement après que la vitesse paramétrée a été atteinte.
- ② Pendant l'exécution de la tâche, le paramètre de sortie "Busy" indique la valeur VRAI.
- ③ Lorsque la vitesse paramétrée est atteinte, le paramètre de sortie "InVelocity" prend la valeur VRAI.
- ④ Si "Execute" conserve la valeur VRAI même après que la vitesse paramétrée a été atteinte, la tâche reste active. "InVelocity" et "Busy" conservent la valeur VRAI et leur état ne passe à FAUX que conjointement avec "Execute".
- ⑤ Si "Execute" a pris la valeur FAUX avant que la vitesse paramétrée n'ait été atteinte, la tâche est achevée lorsque la vitesse paramétrée est atteinte. "InVelocity" indique la valeur VRAI pendant un cycle d'exécution et passe à FAUX conjointement avec "Busy".

Tableau 9- 65 Exemple 2 : La tâche est annulée avant que la vitesse paramétrée n'ait été atteinte

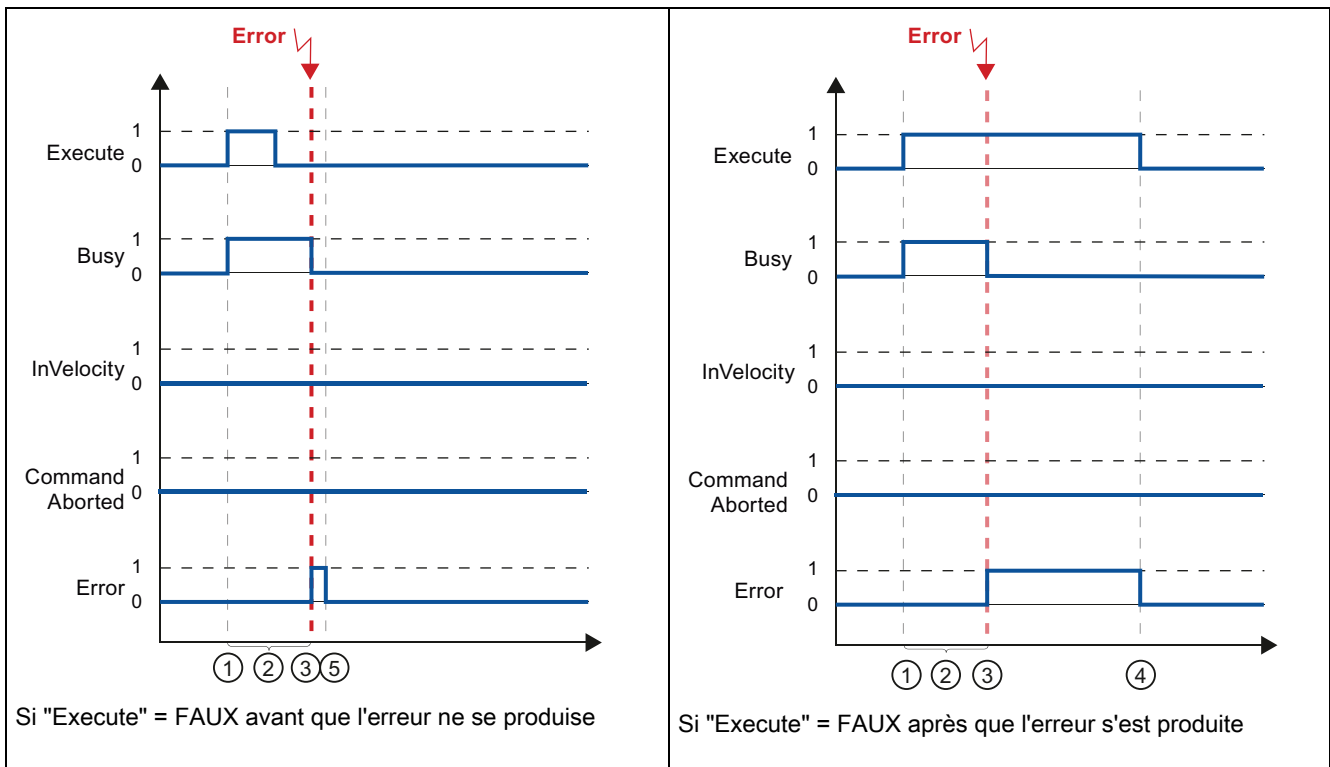


### Remarque

Une annulation n'est pas signalée dans le paramètre de sortie "CommandAborted" dans les conditions suivantes :

- La vitesse paramétrée a été atteinte, le paramètre d'entrée "Execute" a la valeur FAUX et une nouvelle tâche de commande de mouvement est déclenchée.
- Lorsque la vitesse paramétrée a été atteinte et que le paramètre d'entrée "Execute" a pris la valeur FAUX la tâche est achevée. Ainsi, le démarrage d'une nouvelle tâche n'est pas signalée comme une annulation.

Tableau 9- 66 Exemple 3 : Une erreur se produit avant que la vitesse paramétrée n'ait été atteinte



- ① La tâche est démarrée par un front montant du paramètre d'entrée "Execute". Selon la programmation, "Execute" peut être remis à FAUX avant l'apparition de l'erreur ou la valeur VRAI peut être conservée jusqu'après l'apparition de l'erreur.
- ② Pendant l'exécution de la tâche, le paramètre de sortie "Busy" indique la valeur VRAI.
- ③ Une erreur se produit pendant l'exécution de la tâche. Lorsque l'erreur se produit, le paramètre de sortie "Busy" passe à FAUX et "Error" à VRAI.
- ④ Si "Execute" conserve la valeur VRAI jusqu'après l'apparition de l'erreur, "Error" reste également VRAI et ne prend la valeur FAUX que conjointement avec "Execute".
- ⑤ Si "Execute" a pris la valeur FAUX avant l'apparition de l'erreur, "Error" n'indique la valeur VRAI que pendant un cycle d'exécution.

**Remarque**

Une erreur n'est pas signalée dans le paramètre de sortie "Error" dans les conditions suivantes :

- La vitesse paramétrée a été atteinte, le paramètre d'entrée "Execute" a la valeur FAUX et une erreur d'axe se produit (accostage du fin de course logiciel, par exemple).
- Lorsque la vitesse paramétrée a été atteinte et que le paramètre d'entrée "Execute" a pris la valeur FAUX la tâche est achevée. Après l'achèvement de la tâche, l'erreur d'axe est signalée uniquement dans l'instruction "MC\_Power".

### 9.3.6.3 Suivi de l'instruction MC\_MoveJog

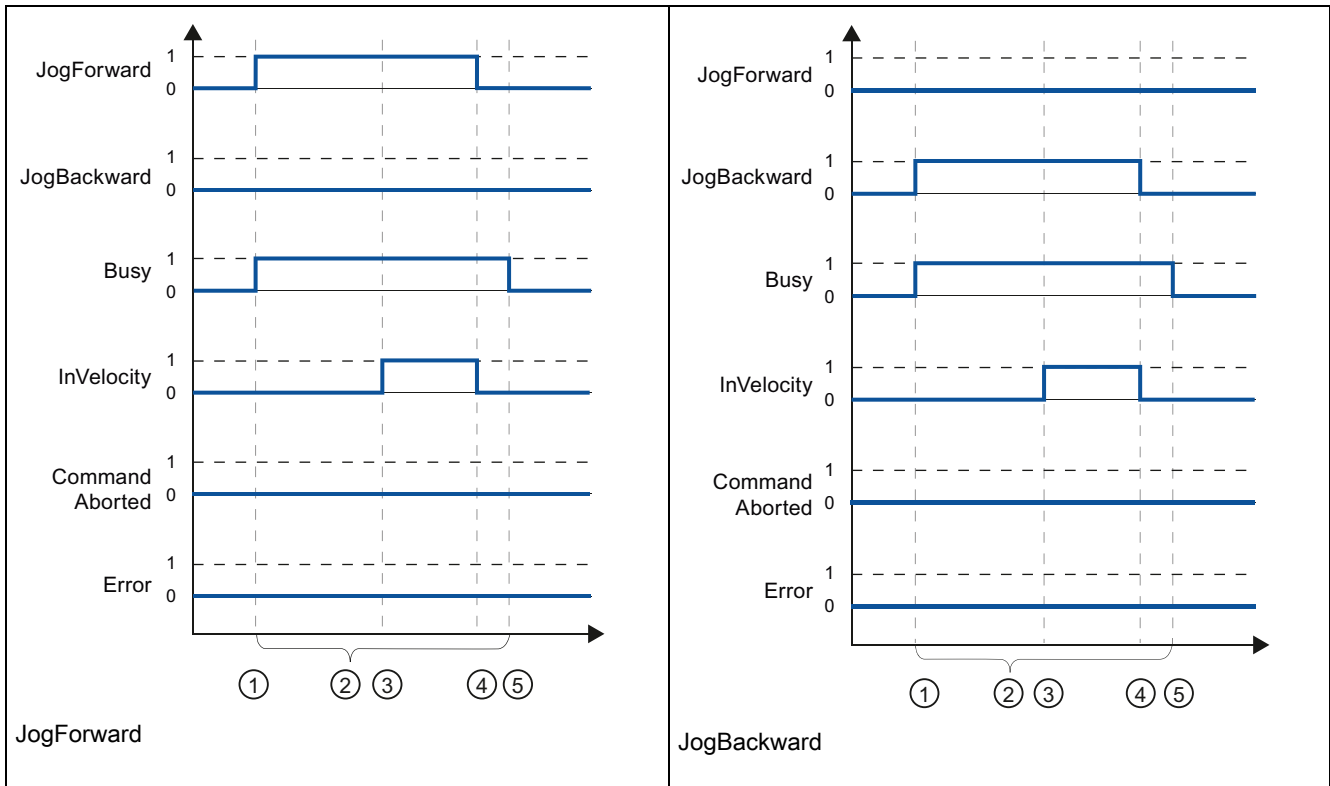
Les tâches de l'instruction de commande de mouvement "MC\_MoveJog" réalisent un mouvement en mode Manuel à vue.

- Les tâches de l'instruction "MC\_MoveJog" n'ont pas de fin définie. L'objectif de la tâche est rempli lorsque la vitesse paramétrée est atteinte pour la première fois et que l'axe se déplace à vitesse constante. Lorsque la vitesse paramétrée est atteinte, le paramètre de sortie "InVelocity" prend la valeur VRAI.
- La tâche est achevée lorsque le paramètre d'entrée "JogForward" ou "JogBackward" a pris la valeur FAUX et que l'axe s'est immobilisé.
- Les paramètres de sortie "Busy", "CommandAborted" et "Error" signalent respectivement que la tâche est encore en cours de traitement, qu'elle a été annulée ou qu'une erreur s'est produite.
  - Pendant l'exécution de la tâche de commande de mouvement, le paramètre de sortie "Busy" indique la valeur VRAI. Si la tâche est achevée, a été annulée ou a été arrêtée en raison d'une erreur, le paramètre de sortie "Busy" prend la valeur FAUX.
  - Le paramètre de sortie "InVelocity" est à l'état VRAI tant que l'axe se déplace à la vitesse paramétrée. Les paramètres de sortie "CommandAborted" et "Error" indiquent l'état pendant au moins un cycle. Ces messages d'état sont rémanents aussi longtemps que l'un ou l'autre paramètre d'entrée "JogForward" ou "JogBackward" a la valeur VRAI.

Le comportement des bits d'état est représenté ci-après pour divers types de situations.

- Le premier exemple montre le comportement de l'axe si la vitesse paramétrée est atteinte et maintenue. Si la tâche de commande de mouvement a été exécutée au moment où la vitesse paramétrée est atteinte, le paramètre de sortie "InVelocity" prend la valeur VRAI.
- Le deuxième exemple montre le comportement de l'axe si la tâche est annulée. Si la tâche de commande de mouvement est annulée pendant son exécution, le paramètre de sortie "CommandAborted" prend la valeur VRAI. Ce comportement est indépendant du fait que la vitesse paramétrée a été ou non atteinte.
- Le troisième exemple montre le comportement de l'axe si une erreur survient. Si une erreur se produit pendant l'exécution de la tâche de commande de mouvement, le paramètre de sortie "Error" prend la valeur VRAI. Ce comportement est indépendant du fait que la vitesse paramétrée a été ou non atteinte.

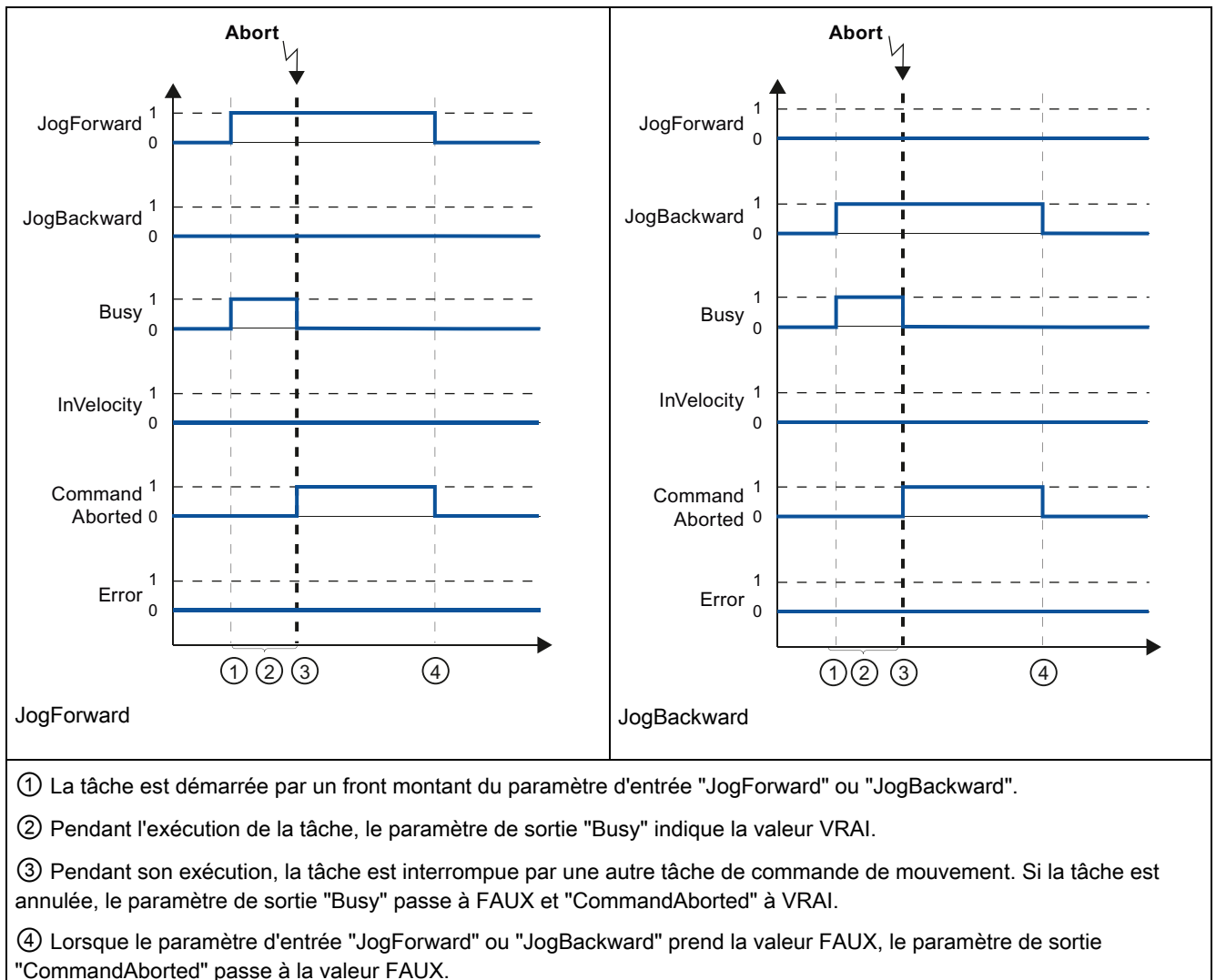
Tableau 9- 67 Exemple 1 : La vitesse paramétrée est atteinte et maintenue



- ① La tâche est démarrée par un front montant du paramètre d'entrée "JogForward" ou "JogBackward".
- ② Pendant l'exécution de la tâche, le paramètre de sortie "Busy" indique la valeur VRAI.
- ③ Lorsque la vitesse paramétrée est atteinte, le paramètre de sortie "InVelocity" prend la valeur VRAI.
- ④ Lorsque le paramètre d'entrée "JogForward" ou "JogBackward" prend la valeur FAUX, le mouvement de l'axe s'arrête. L'axe commence à décélérer. Il en résulte que l'axe ne se déplace plus à vitesse constante et le paramètre de sortie "InVelocity" passe à l'état FAUX.
- ⑤ Si l'axe s'est immobilisé, la tâche de commande de mouvement est achevée et le paramètre de sortie "Busy" prend la valeur FAUX.



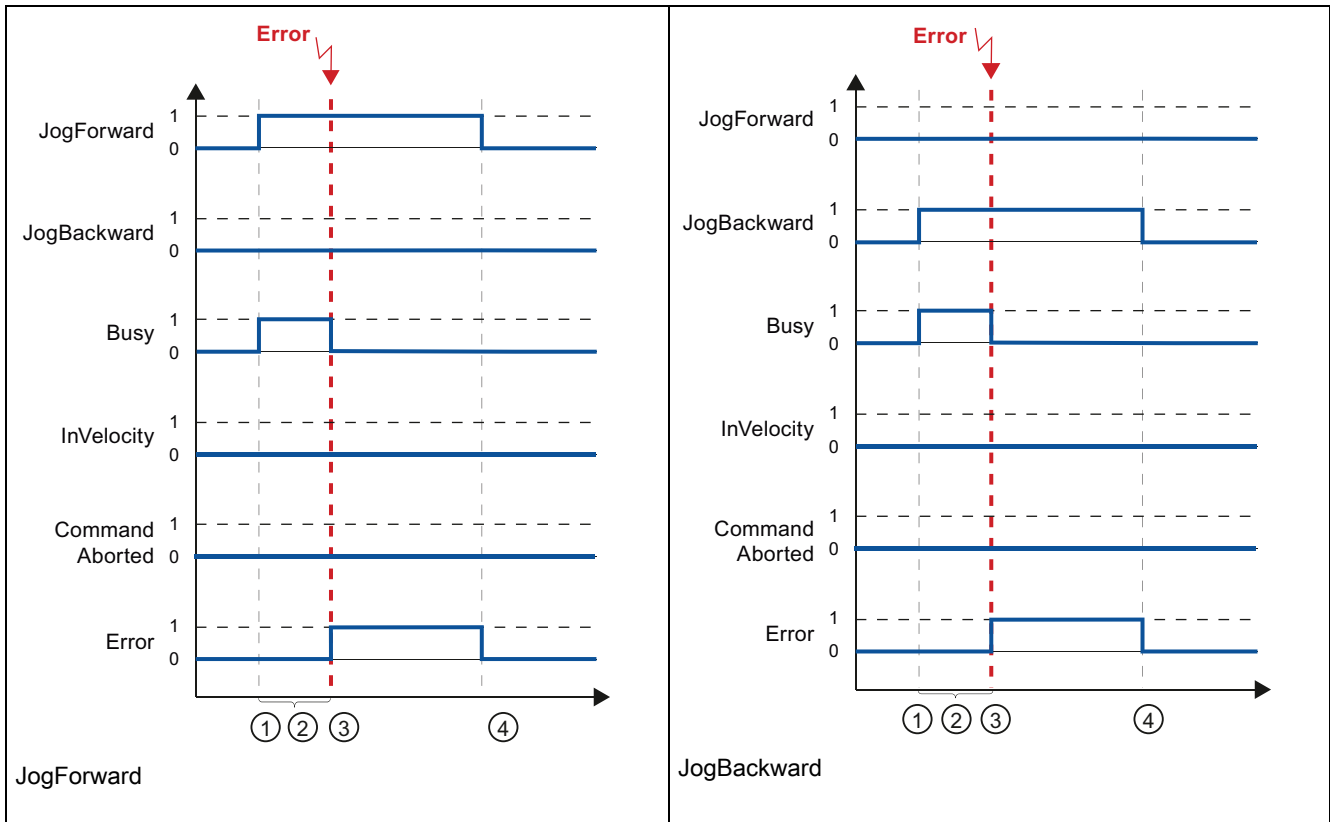
Tableau 9- 68 Exemple 2 : La tâche est annulée pendant l'exécution



**Remarque**

L'annulation de la tâche est signalée dans le paramètre de sortie "CommandAborted" pendant un cycle d'exécution seulement, si toutes les conditions ci-dessous sont remplies :  
 Les paramètres d'entrée "JogForward" et "JogBackward" ont la valeur FAUX (mais l'axe est encore en cours de décélération) et une nouvelle tâche de commande de mouvement est déclenchée.

Tableau 9- 69Exemple 3 : Une erreur s'est produite pendant l'exécution de la tâche



- ① La tâche est démarrée par un front montant du paramètre d'entrée "JogForward" ou "JogBackward".
- ② Pendant l'exécution de la tâche, le paramètre de sortie "Busy" indique la valeur VRAI.
- ③ Une erreur se produit pendant l'exécution de la tâche. Lorsque l'erreur se produit, le paramètre de sortie "Busy" passe à FAUX et "Error" à VRAI.
- ④ Lorsque le paramètre d'entrée "JogForward" ou "JogBackward" prend la valeur FAUX, le paramètre de sortie "Error" passe à la valeur FAUX.

**Remarque**

L'apparition de l'erreur est signalée dans le paramètre de sortie "Error" pendant un cycle d'exécution seulement, si toutes les conditions ci-dessous sont remplies :

Les paramètres d'entrée "JogForward" et "JogBackward" ont la valeur FAUX (mais l'axe est encore en cours de décélération) et une nouvelle erreur se produit (accostage du fin de course logiciel, par exemple).

Le S7-1200 offre plusieurs types de communication entre les CPU et les consoles de programmation, IHM et autres CPU.

## PROFINET

PROFINET est utilisé pour l'échange de données par le biais du programme utilisateur avec d'autres partenaires de communication via Ethernet :

- La CPU offre les prises en charge PROFINET et PROFIBUS suivantes :
  - Dans V3.0, PROFINET prend en charge 16 périphériques IO avec 256 sous-modules au maximum. PROFIBUS autorise 3 maîtres PROFIBUS DP indépendants, prenant en charge 32 périphériques IO avec 512 sous-modules au maximum par périphérique IO.
  - Dans V2.2, PROFINET prend en charge 8 périphériques IO avec 128 sous-modules au maximum (si le nombre d'esclaves ou sous-modules PROFIBUS configurés est inférieur ou égal à 8). PROFIBUS prend en charge 16 périphériques PROFIBUS IO au maximum sur un seul maître, avec 256 sous-modules au maximum par périphérique IO.
- Communication S7
- User Datagram Protocol (UDP)
- ISO sur TCP (RFC 1006)
- Transport Control Protocol (TCP)

## Contrôleur IO PROFINET RT

En tant que contrôleur IO utilisant PROFINET RT, la CPU offre les prises en charge suivantes sur le réseau PN local ou via un coupleur (link) PN/PN (voir PROFIBUS et PROFINET International ([www.profinet.com](http://www.profinet.com)) pour plus d'informations) :

- Dans V3.0, le S7-1200 communique avec 16 appareils PN au maximum.
- Dans V2.2, le S7-1200 communique avec 8 appareils PN au maximum.

## PROFIBUS

PROFIBUS est utilisé pour l'échange de données par le biais du programme utilisateur avec d'autres partenaires de communication via le réseau PROFIBUS :

- Avec le CM 1242-5, la CPU fonctionne en tant qu'esclave PROFIBUS DP.
- Avec le CM 1243-5, la CPU fonctionne en tant que maître PROFIBUS DP de classe 1.
- Dans V3.0, les esclaves PROFIBUS DP, maîtres PROFIBUS DP et interfaces AS-i (les trois modules de communication du côté gauche) et PROFINET sont indépendants.

### 10.1 Nombre de liaisons de communication asynchrones prises en charge

- Dans V2.2, la CPU offre les prises en charge PROFINET et PROFIBUS suivantes :
  - Un total de 16 périphériques et 256 sous-modules, avec un maximum de 16 esclaves PROFIBUS DP et 256 sous-modules (si aucun sous-module ni périphérique PROFINET IO n'est configuré).

---

#### Remarque

Dans V2.2, le total de 16 périphériques PROFINET et PROFIBUS comprend :

- les modules esclaves PROFIBUS DP reliés au maître PROFIBUS DP (CM 1243-5),
- tout module esclave PROFIBUS DP (CM 1242-5) relié à la CPU,
- tout périphérique PROFINET connecté à la CPU via le port PROFINET.

Une configuration avec trois CM PROFIBUS (un maître CM 1243-5 et deux modules esclaves CM 1242-5) réduirait, par exemple, le nombre maximum de modules esclaves accessibles au maître PROFIBUS DP (CM 1243-5) à 14.

---

- Interface AC : Le module CM 1243-2 maître AS-i S7-1200 permet le raccordement d'un réseau interface AC à une CPU S7-1200.
- Communication CPU à CPU S7

### Communication TeleService

Avec TeleService via GPRS, une station d'ingénierie sur laquelle STEP 7 est installé communique via le réseau GSM et Internet avec une station SIMATIC S7-1200 munie d'un CP 1242-7. La liaison passe par un serveur TeleControl qui sert d'intermédiaire et qui est connecté à Internet.

## 10.1 Nombre de liaisons de communication asynchrones prises en charge

La CPU prend en charge le nombre maximum suivant de liaisons de communication asynchrones simultanées pour PROFINET et PROFIBUS :

- 8 liaisons pour la communication ouverte (Open User Communications) active ou passive : TSEND\_C, TRCV\_C, TCON, TDISCON, TSEND et TRCV.
- 3 liaisons CPU à CPU S7 pour données GET/PUT serveur
- 8 liaisons CPU à CPU S7 pour données GET/PUT client

---

#### Remarque

Les CPU S7-1200, S7-300 et S7-400 utilisent les instructions GET et PUT pour la communication CPU à CPU S7. Une CPU S7-200 utilise des instructions ETHx\_XFER pour la communication CPU à CPU S7.

---

- Liaisons IHM : La CPU fournit des liaisons IHM spécialisées pour la prise en charge de 3 appareils IHM au maximum (vous pouvez avoir 2 SIMATIC Comfort Panels au maximum). Le nombre total de stations IHM dépend des types de pupitres IHM dans votre configuration. Vous pouvez, par exemple, avoir trois SIMATIC Basic Panels au plus connectés à votre CPU ou bien deux SIMATIC Comfort Panels au plus avec un Basic Panel additionnel.
- Liaisons PG : La CPU fournit des liaisons pour la prise en charge d'une console de programmation (PG).
- Liaisons serveur Web (HTTP) : La CPU fournit des liaisons pour le serveur Web.

## 10.2 PROFINET

### 10.2.1 Liaison appareil local/partenaire

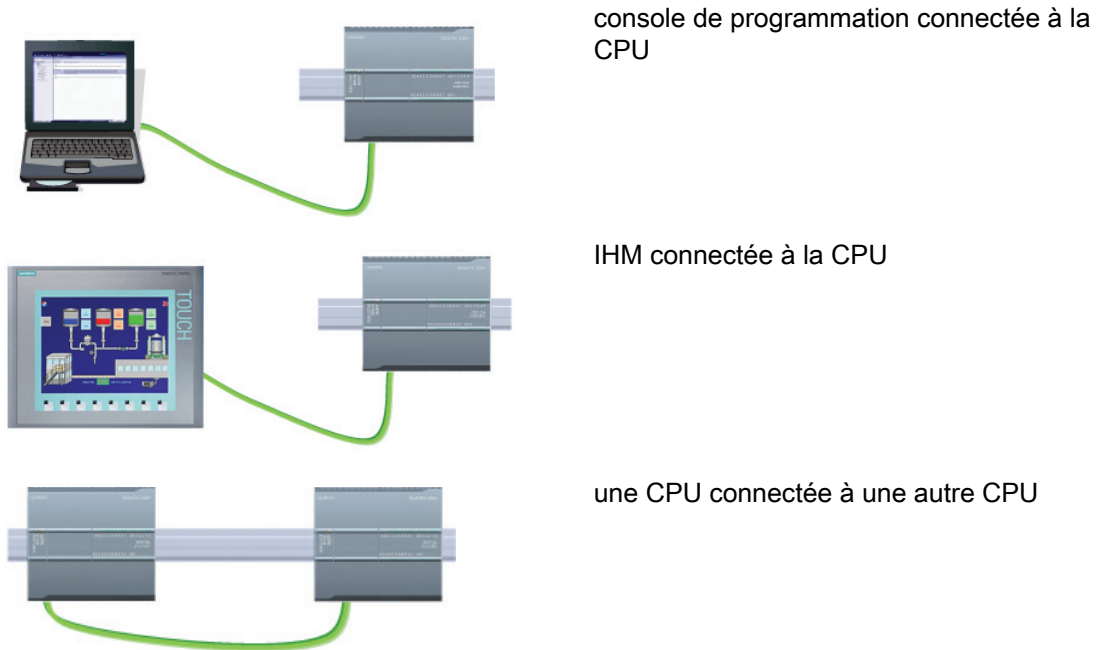
Une liaison appareil local / partenaire (éloigné) définit une affectation logique de deux partenaires de communication pour établir des services de communication. Une liaison définit les éléments suivants :

- Partenaires de communication concernés (un actif, un passif)
- Type de liaison (par exemple, liaison API, IHM ou d'appareil)
- Routage

Les partenaires de communication exécutent les instructions pour configurer et établir la liaison de communication. Vous vous servez de paramètres pour indiquer les partenaires de noeud d'extrémité actif et passif. Une fois la liaison configurée et établie, elle est automatiquement conservée et surveillée par la CPU. Reportez-vous au paragraphe "Configuration de la liaison locale/partenaire" (Page 136) pour plus d'informations sur la configuration des paramètres pour la liaison.

Si la liaison est interrompue (en raison d'une coupure de ligne, par exemple), le partenaire actif tente de rétablir la liaison configurée. Vous n'avez pas besoin de réexécuter l'instruction de communication.

La CPU peut communiquer avec d'autres CPU, avec des consoles de programmation, avec des appareils IHM et avec des appareils non Siemens utilisant des protocoles de communication TCP standard.



### Commutation Ethernet

Le port PROFINET des CPU 1211C, 1212C et 1214C ne contient pas de dispositif de commutation Ethernet. Une connexion directe entre une console de programmation ou une IHM et une CPU ne nécessite pas de commutateur Ethernet. Toutefois, un commutateur Ethernet est obligatoire pour un réseau comportant plus de deux CPU ou appareils IHM.



La CPU 1215C comprend pour sa part un commutateur Ethernet à 2 ports intégré. Vous pouvez avoir un réseau avec une CPU 1215C et deux autres CPU S7-1200. Vous pouvez également utiliser le commutateur Ethernet à 4 ports CSM1277, monté sur châssis, pour connecter plusieurs CPU et appareils IHM.

## 10.2.2 Communication ouverte (Open User Communication)

### 10.2.2.1 ID de liaison pour les instructions PROFINET

Lorsque vous insérez les instructions PROFINET TSEND\_C, TRCV\_C ou TCON dans votre programme utilisateur, STEP 7 crée un DB d'instance pour configurer la voie de communication (ou liaison) entre les appareils. Servez-vous des "Propriétés" de l'instruction pour configurer les paramètres pour la liaison. L'ID de liaison pour cette liaison figure parmi ces paramètres.

- L'ID de liaison doit être unique pour la CPU. Chaque liaison que vous créez doit avoir un DB et un ID de liaison différents.
- La CPU locale et la CPU partenaire peuvent toutes deux utiliser le même numéro d'ID de liaison pour la même liaison, mais les numéros d'ID de liaison ne doivent pas obligatoirement coïncider. Le numéro d'ID de liaison n'est significatif que pour les instructions PROFINET à l'intérieur du programme utilisateur de la CPU concernée.
- Vous pouvez utiliser n'importe quel nombre pour l'ID de liaison de la CPU. Toutefois, configurer les ID de liaison séquentiellement à partir de "1" constitue une méthode facile pour garder une trace du nombre de liaisons utilisées pour une CPU spécifique.

---

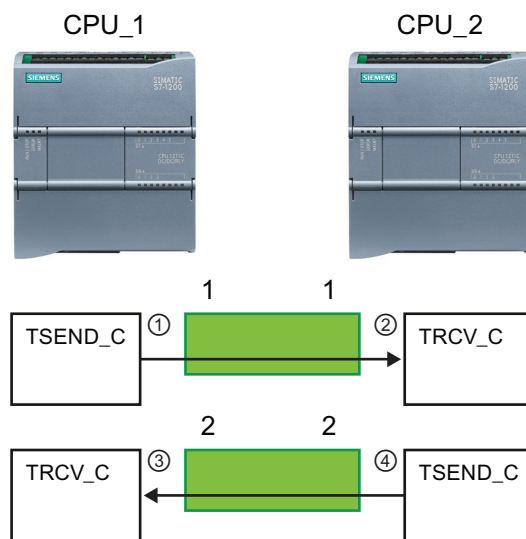
#### Remarque

Chaque instruction TSEND\_C, TRCV\_C ou TCON dans votre programme utilisateur crée une nouvelle liaison. Il est important d'utiliser l'ID de liaison correct pour chaque liaison.

---

L'exemple suivant montre la communication entre deux CPU qui utilisent deux liaisons distinctes pour émettre et recevoir des données.

- L'instruction TSEND\_C dans la CPU\_1 se connecte à l'instruction TRCV\_V dans la CPU\_2 par le biais de la première liaison ("ID de liaison 1" à la fois sur la CPU\_1 et la CPU\_2).
- L'instruction TRCV\_C dans la CPU\_1 se connecte à l'instruction TSEND\_C dans la CPU\_2 par le biais de la deuxième liaison ("ID de liaison 2" à la fois sur la CPU\_1 et la CPU\_2).

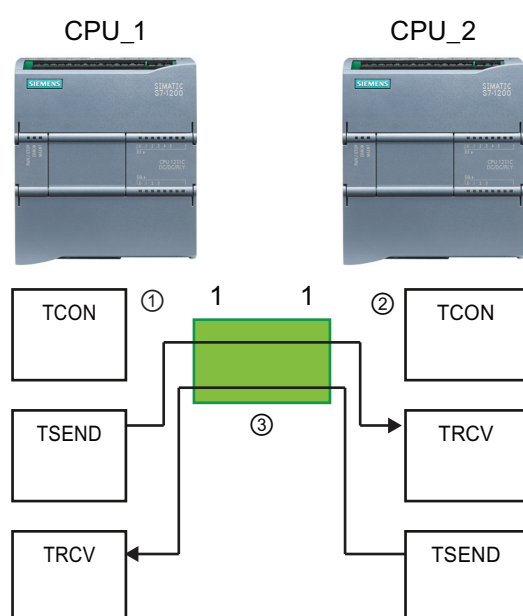


- ① TSEND\_C sur la CPU\_1 crée une liaison et affecte un ID de liaison à cette liaison (ID de liaison 1 pour la CPU\_1).
- ② TRCV\_C sur la CPU\_2 crée la liaison pour la CPU\_2 et affecte l'ID de liaison (ID de liaison 1 pour la CPU\_2).
- ③ TRCV\_C sur la CPU\_1 crée une deuxième liaison pour la CPU\_1 et affecte un ID de liaison différent à cette liaison (ID de liaison 2 pour la CPU\_1).
- ④ TSEND\_C sur la CPU\_2 crée une deuxième liaison et affecte un ID de liaison différent à cette liaison (ID de liaison 2 pour la CPU\_2).



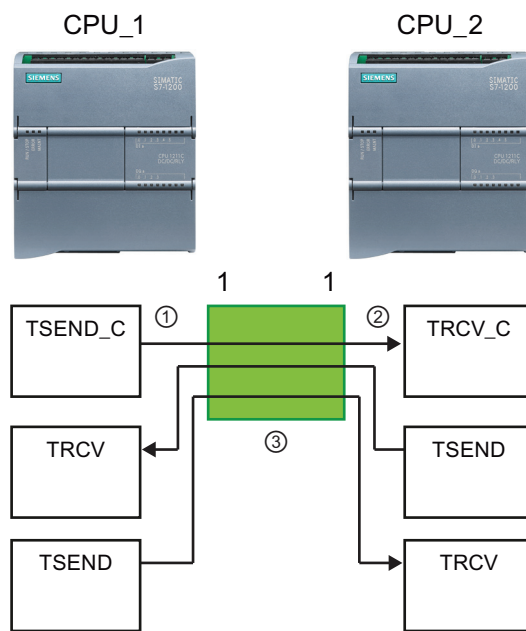
L'exemple suivant montre la communication entre deux CPU qui utilisent une même liaison pour émettre et recevoir des données.

- Chaque CPU utilise une instruction TCON pour configurer la liaison entre les deux CPU.
- L'instruction TSEND dans la CPU\_1 se connecte à l'instruction TRCV dans la CPU\_2 par le biais de l'ID de liaison ("ID de liaison 1") qui a été configurée par l'instruction TCON dans la CPU\_1. L'instruction TRCV dans la CPU\_2 se connecte à l'instruction TSEND dans la CPU\_1 par le biais de l'ID de liaison ("ID de liaison 1") qui a été configurée par l'instruction TCON dans la CPU\_2.
- L'instruction TSEND dans la CPU\_2 se connecte à l'instruction TRCV dans la CPU\_1 par le biais de l'ID de liaison ("ID de liaison 1") qui a été configurée par l'instruction TCON dans la CPU\_2. L'instruction TRCV dans la CPU\_1 se connecte à l'instruction TSEND dans la CPU\_2 par le biais de l'ID de liaison ("ID de liaison 1") qui a été configurée par l'instruction TCON dans la CPU\_1.



- ① TCON sur la CPU\_1 crée une liaison et affecte un ID de liaison à cette liaison sur la CPU\_1 (ID=1).
- ② TCON sur la CPU\_2 crée une liaison et affecte un ID de liaison à cette liaison sur la CPU\_2 (ID=1).
- ③ TSEND et TRCV sur la CPU\_1 utilisent l'ID de liaison créé par l'instruction TCON sur la CPU\_1 (ID=1). TSEND et TRCV sur la CPU\_2 utilisent l'ID de liaison créé par l'instruction TCON sur la CPU\_2 (ID=1).

Comme le montre l'exemple suivant, vous pouvez également utiliser des instructions TSEND et TRCV individuelles pour communiquer par le biais d'une liaison créée par une instruction TSEND\_C ou TRCV\_C. Les instructions TSEND et TRCV ne créent pas elles-mêmes de nouvelle liaison de sorte qu'elles doivent utiliser le DB et l'ID de liaison créés par une instruction TSEND\_C, TRCV\_C ou TCON.



- ① TSEND\_C sur la CPU\_1 crée une liaison et affecte un ID de liaison à cette liaison (ID=1).
- ② TRCV\_C sur la CPU\_2 crée une liaison et affecte un ID de liaison à cette liaison sur la CPU\_2 (ID=1).
- ③ TSEND et TRCV sur la CPU\_1 utilisent l'ID de liaison créé par l'instruction TSEND\_C sur la CPU\_1 (ID=1).  
TSEND et TRCV sur la CPU\_2 utilisent l'ID de liaison créé par l'instruction TRCV\_C sur la CPU\_2 (ID=1).

### Voir aussi

Configuration du routage local/partenaire (Page 136)

### 10.2.2.2 Protocoles

Le port PROFINET intégré de la CPU prend en charge plusieurs normes de communication via un réseau Ethernet :

- Transport Control Protocol (TCP)
- ISO sur TCP (RFC 1006)
- User Datagram Protocol (UDP)

Tableau 10- 1 Protocoles et leurs instructions de communication respectives

Protocole	Exemples d'utilisation	Entrée des données dans la zone de réception	Instructions de communication	Type d'adressage
TCP	Communication CPU à CPU	Mode ad hoc	Seulement TRCV_C et TRCV	Affecte des numéros de port aux appareils local (actif) et partenaire (passif).
	Transport de trames	Réception de données avec longueur indiquée	TSEND_C, TRCV_C, TCON, TDISCON, TSEND et TRCV	
ISO sur TCP	Communication CPU à CPU	Mode ad hoc	Seulement TRCV_C et TRCV	Affecte des TSAP aux appareils local (actif) et partenaire (passif).
	Fragmentation et réassemblage des messages	Gestion par le protocole	TSEND_C, TRCV_C, TCON, TDISCON, TSEND et TRCV	
UDP	Communication CPU à CPU Communication du programme utilisateur	User Datagram Protocol (protocole de datagramme utilisateur)	TUSEND et TURCV	Affecte des numéros de port aux appareils local (actif) et partenaire (passif), mais ce n'est pas une liaison spécialisée.
Communication S7	Communication CPU à CPU Lecture/écriture de données depuis/vers une CPU	Emission et réception de données avec longueur indiquée	GET et PUT	Affecte des TSAP aux appareils local (actif) et partenaire (passif).
PROFINET RT	Communication CPU à périphérique PROFINET IO	Emission et réception de données avec longueur indiquée	Intégrées	Intégré

### 10.2.2.3 Mode ad hoc

Typiquement, TCP et ISO sur TCP reçoivent des paquets de données d'une longueur indiquée, comprise entre 1 et 8192 octets. Toutefois, les instructions de communication TRCV\_C et TRCV fournissent aussi un mode de communication "ad hoc" qui permet de recevoir des paquets de données de longueur variable allant de 1 à 1472 octets.

---

#### Remarque

Si vous stockez les données dans un DB "optimisé" (symbolique uniquement), vous ne pouvez recevoir les données qu'en tableaux de types de données Byte, Char, USInt et SInt.

---

Pour activer le mode ad hoc de l'instruction TRCV\_C ou TRCV, donnez la valeur 65535 (0xFFFF) au paramètre LEN.

Si vous n'appelez pas souvent l'instruction TRCV\_C ou TRCV en mode ad hoc, vous recevrez peut-être plus d'un paquet lors d'un appel. Ainsi, si vous deviez recevoir cinq paquets de 100 octets avec un appel, TCP transmettrait ces cinq paquets en un paquet de 500 octets alors que ISO sur TCP restructurerait les paquets en cinq paquets de 100 octets.

#### 10.2.2.4 TCP et ISO sur TCP

TCP est un protocole standard décrit par RFC 793 : Transmission Control Protocol. L'objectif principal de TCP est de fournir un service de liaison sûr et fiable entre des paires de processus. Ce protocole présente les caractéristiques suivantes :

- Protocole de communication efficace car il est étroitement lié au matériel
- Convenant à des quantités moyennes à élevées de données (jusqu'à 8192 octets)
- Fournit considérablement plus de fonctions pour les applications, notamment la reprise sur erreur, le contrôle de flux et la fiabilité.
- Protocole orienté connexion
- Utilisable très facilement avec des systèmes tiers supportant exclusivement TCP
- Apte au routage
- Longueurs de données statiques uniquement
- Accusé de réception des messages
- Adressage des applications à l'aide de numéros de port
- La plupart des protocoles d'application utilisateur, tels que TELNET et FTP, utilisent TCP.
- Un effort de programmation est nécessaire pour la gestion des données en raison de l'interface de programmation EMISSION / RECEPTION.

ISO sur TCP (International Standards Organization on Transport Control Protocol ; RFC 1006) est un mécanisme qui permet aux applications ISO d'être portées sur le réseau TCP/IP. Ce protocole présente les caractéristiques suivantes :

- Protocole de communication efficace étroitement lié au matériel
- Convenant à des quantités moyennes à élevées de données (jusqu'à 8192 octets)
- Contrairement à TCP, les messages comportent une identification de fin de données et sont orientés messages.
- Apte au routage ; peut être utilisé dans des réseaux étendus
- Longueurs de données dynamiques possibles
- Un effort de programmation est nécessaire pour la gestion des données en raison de l'interface de programmation EMISSION / RECEPTION.

Grâce à des points d'accès au service transport (TSAP), le protocole TCP autorise plusieurs liaisons à une adresse IP unique (jusqu'à 64K liaisons). Avec RFC 1006, les TSAP identifient de manière unique ces liaisons de noeud d'extrémité à une adresse IP.

#### TSEND\_C et TRCV\_C

L'instruction TSEND\_C combine les fonctions des instructions TCON, TDISCON et TSEND .  
L'instruction TRCV\_C combine les fonctions des instructions TCON, TDISCON et TRCV.  
Reportez-vous à "TCON, TDISCON, TSEND et TRCV (Page 467)" pour plus d'informations sur ces instructions.

La taille de données minimale que vous pouvez envoyer (TSEND\_C) ou recevoir (TRCV\_C) est un octet ; la taille maximale est 8192 octets. TSEND\_C n'accepte pas l'envoi de données à partir d'adresses booléennes et TRCV\_C ne reçoit pas de données dans des adresses booléennes. Pour plus d'informations sur le transfert de données à l'aide de ces instructions, reportez-vous au paragraphe sur la cohérence des données (Page 163).

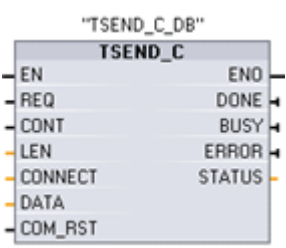
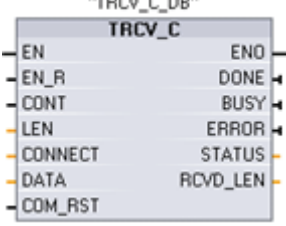
### Remarque

#### Initialisation des paramètres de communication

Une fois que vous avez inséré l'instruction TSEND\_C ou TRCV\_C, utilisez les "Propriétés" de l'instruction (Page 136) pour configurer les paramètres de communication. Lorsque vous entrez les paramètres pour les partenaires de communication dans la fenêtre d'inspection, STEP 7 entre les données correspondantes dans le DB pour l'instruction.

Si vous voulez utiliser un DB multi-instance, vous devez configurer le DB manuellement dans les deux CPU.

Tableau 10- 2 Instructions TSEND\_C et TRCV\_C

CONT/LOG	SCL	Description
	<pre>"TSEND_C_DB" (     req:=_bool_in_,     cont:=_bool_in_,     len:=_uint_in_,     done=&gt;_bool_out_,     busy=&gt;_bool_out_,     error=&gt;_bool_out_,     status=&gt;_word_out_,     connect:=_struct_inout_,     data:=_variant_inout_,     com rst:=_bool_inout_);</pre>	<p>TSEND_C établit une liaison de communication TCP ou ISO sur TCP à une station partenaire, envoie des données et peut mettre fin à la liaison. Une fois la liaison configurée et établie, elle est automatiquement conservée et surveillée par la CPU.</p>
	<pre>"TRCV_C_DB" (     en_r:=_bool_in_,     cont:=_bool_in_,     len:=_uint_in_,     done=&gt;_bool_out_,     busy=&gt;_bool_out_,     error=&gt;_bool_out_,     status=&gt;_word_out_,     rcvd_len=&gt;_uint_out_,     connect:=_struct_inout_,     data:=_variant_inout_,     com rst:=_bool_inout_);</pre>	<p>TRCV_C établit une liaison de communication TCP ou ISO sur TCP à une CPU partenaire, reçoit des données et peut mettre fin à la liaison. Une fois la liaison configurée et établie, elle est automatiquement conservée et surveillée par la CPU.</p>

<sup>1</sup> STEP 7 crée automatiquement le DB lorsque vous insérez l'instruction.

Tableau 10- 3 Types de données pour les paramètres de TSEND\_C et TRCV\_C

Paramètre et type		Type de données	Description
REQ (TSEND_C)	IN	Bool	Le paramètre de commande REQ lance la tâche d'émission avec la liaison décrite dans CONNECT en présence d'un front montant.
EN_R (TRCV_C)	IN	Bool	Paramètre de commande Activé pour recevoir : Lorsque EN_R est égal à 1, TRCV_C est prêt à recevoir. La tâche de réception est traitée.
CONT	IN	Bool	<ul style="list-style-type: none"> <li>0: Déconnexion</li> <li>1: Etablissement et maintien de la liaison</li> </ul>
LEN	IN	UInt	<p>Nombre maximum d'octets à envoyer (TSEND_C) ou à recevoir (TRCV_C) :</p> <ul style="list-style-type: none"> <li>Valeur par défaut = 0 : Le paramètre DATA détermine la longueur des données à envoyer (TSEND_C) ou à recevoir (TRCV_C).</li> <li>Mode ad hoc = 65535 : Une longueur de données variable est définie pour la réception (TRCV_C).</li> </ul>
CONNECT	IN_OUT	TCON_Param	Pointeur désignant la description de la liaison
DATA	IN_OUT	Variant	<ul style="list-style-type: none"> <li>Contient l'adresse et la longueur des données à envoyer (TSEND_C)</li> <li>Contient l'adresse de début et la longueur maximale des données reçues (TRCV_C).</li> </ul>
COM_RST	IN_OUT	Bool	<p>Permet le redémarrage de l'instruction :</p> <ul style="list-style-type: none"> <li>0: Sans objet</li> <li>1: Redémarrage complet du bloc fonctionnel avec coupure de la liaison existante</li> </ul>
DONE	OUT	Bool	<ul style="list-style-type: none"> <li>0: La tâche n'a pas encore commencé ou est encore en cours d'exécution.</li> <li>1: Tâche achevée sans erreur</li> </ul>
BUSY	OUT	Bool	<ul style="list-style-type: none"> <li>0: Tâche achevée</li> <li>1: La tâche n'est pas encore achevée. Il n'est pas possible de déclencher une nouvelle tâche.</li> </ul>
ERROR	OUT	Bool	<p>Paramètre d'état aux valeurs suivantes :</p> <ul style="list-style-type: none"> <li>0: Pas d'erreur</li> <li>1: Une erreur s'est produite pendant le traitement. STATUS contient des informations détaillées sur le type d'erreur.</li> </ul>
STATUS	OUT	Word	Information d'état incluant l'information d'erreur (voir le tableau "Paramètres Error et Status" ci-après)
RCVD_LEN (TRCV_C)	OUT	Int	Quantité de données effectivement reçue en octets

---

### Remarque

L'instruction TSEND\_C requiert un front montant du paramètre d'entrée REQ pour démarrer une tâche d'émission. Le paramètre BUSY est mis à 1 pendant le traitement. L'achèvement de la tâche d'émission est signalé par la mise à 1 du paramètre DONE ou ERROR pendant un cycle. Pendant ce temps, aucun front montant du paramètre d'entrée REQ n'est pris en compte.

---

### Remarque

Avec la valeur par défaut du paramètre LEN (LEN = 0), le paramètre DATA est utilisé pour déterminer la longueur des données transmises. Assurez-vous que les données DATA envoyées par l'instruction TSEND\_C ont la même taille que le paramètre DATA de l'instruction TRCV\_C.

---

## Fonctionnement de TSEND\_C

L'instruction TSEND\_C fonctionne de la manière suivante :

- Pour établir une liaison, exécutez TSEND\_C avec CONT = 1.
- Une fois la liaison établie avec succès, TSEND\_C met le paramètre DONE à 1 pour un cycle.
- Pour mettre fin à la liaison de communication, exécutez TSEND\_C avec CONT = 0. La liaison est immédiatement interrompue. Cela affecte également la station réceptrice. La connexion y est également coupée et il peut y avoir perte de données dans la mémoire tampon de réception.
- Pour envoyer des données par le biais d'une liaison établie, exécutez TSEND\_C avec un front montant en REQ. Après une opération d'envoi réussie, TSEND\_C met le paramètre DONE à 1 pour un cycle.
- Pour établir une liaison et envoyer des données, exécutez TSEND\_C avec CONT = 1 et REQ = 1. Après une opération d'envoi réussie, TSEND\_C met le paramètre DONE à 1 pour un cycle.

## Fonctionnement de TRCV\_C

L'instruction TRCV\_C fonctionne de la manière suivante :

- Pour établir une liaison, exécutez TRCV\_C avec le paramètre CONT = 1.
- Pour recevoir des données, exécutez TRCV\_C avec le paramètre EN\_R = 1. TRCV\_C reçoit les données en continu lorsque les paramètres EN\_R = 1 et CONT = 1.
- Pour mettre fin à la liaison, exécutez TRCV\_C avec le paramètre CONT = 0. La liaison est immédiatement interrompue et une perte des données est possible.

TRCV\_C gère les mêmes modes de réception que l'instruction TRCV. Le tableau suivant montre comment les données sont entrées dans la zone de réception.

Tableau 10- 4 Entrée des données dans la zone de réception

Variante de protocole	Entrée des données dans la zone de réception	Paramètre "connection_type"	Valeur du paramètre LEN	Valeur du paramètre RCVD_LEN (octets)
TCP	Mode ad hoc	B#16#11	65535	1 à 1472
TCP	Réception de données avec longueur indiquée	B#16#11	0 (recommandé) ou 1 à 8192, 65535 excepté	1 à 8192
ISO sur TCP	Mode ad hoc	B#16#12	65535	1 à 1472
ISO sur TCP	Gestion par le protocole	B#16#12	0 (recommandé) ou 1 à 8192, 65535 excepté	1 à 8192

---

### Remarque

#### Mode ad hoc

Le "mode ad hoc" existe avec les variantes de protocole TCP et ISO sur TCP. Vous paramétrez le "mode ad hoc" en affectant 65535 au paramètre LEN. La zone de réception est identique à la zone formée par DATA. La longueur des données reçues est transmise au paramètre RCVD\_LEN.

Si vous stockez les données dans un DB "optimisé" (symbolique uniquement), vous ne pouvez recevoir les données qu'en tableaux de types de données Byte, Char, USInt et SInt.

---

### Remarque

#### Importation dans le S7-1200 de projets STEP 7 S7-300/400 utilisant le "mode ad hoc"

Dans les projets STEP 7 S7-300/400, le "mode ad hoc" est sélectionné par affectation de 0 au paramètre LEN. Dans le S7-1200, vous paramétrez le "mode ad hoc" en affectant 65535 au paramètre LEN.

Si vous importez dans le S7-1200 un projet STEP 7 S7-300/400 utilisant le "mode ad hoc", vous devez donner la valeur 65535 au paramètre LEN.

---

### Remarque

En raison du traitement asynchrone de TSEND\_C, vous devez faire en sorte que les données dans la zone d'émission restent cohérentes jusqu'à ce que le paramètre DONE ou le paramètre ERROR prenne la valeur VRAI.

Pour TSEND\_C, un état VRAI du paramètre DONE signifie que les données ont été envoyées avec succès. Cela ne signifie pas que la CPU partenaire de liaison ait effectivement lu la mémoire tampon de réception.

En raison du traitement asynchrone de TRCV\_C, les données dans la zone de réception ne sont cohérentes que lorsque le paramètre DONE est égal à 1.

---



Tableau 10- 5 Paramètres BUSY, DONE et ERROR des instructions TSEND\_C et TRCV\_C

BUSY	DONE	ERROR	Description
VRAI	Sans objet	Sans objet	La tâche est en cours de traitement.
FAUX	VRAI	FAUX	La tâche a été achevée avec succès.
FAUX	FAUX	VRAI	La tâche s'est terminée avec une erreur. La cause de l'erreur figure dans le paramètre STATUS.
FAUX	FAUX	FAUX	Aucune nouvelle tâche n'a été affectée.

## Paramètres Error et Status

Tableau 10- 6 Codes d'erreur pour les paramètres ERROR et STATUS de TSEND\_C et TRCV\_C

ERROR	STATUS	Description
0	0000	Tâche exécutée sans erreur
0	7000	Pas de traitement de tâche actif
0	7001	Démarrage du traitement de la tâche, établissement de la liaison, attente du partenaire de liaison
0	7002	Données en cours d'émission ou de réception
0	7003	Coupure de la liaison
0	7004	Liaison établie et surveillée ; pas de traitement de tâche actif
1	8085	Le paramètre LEN est supérieur à la plus grande valeur autorisée.
1	8086	Le paramètre CONNECT se situe hors de la plage permise.
1	8087	Nombre maximal de liaisons atteint ; aucune liaison supplémentaire n'est possible.
1	8088	Le paramètre LEN n'est pas valide pour la zone de mémoire indiquée dans DATA.
1	8089	Le paramètre CONNECT ne pointe pas sur un bloc de données.
1	8091	Profondeur d'imbrication maximale dépassée
1	809A	Le paramètre CONNECT pointe sur un champ qui ne correspond pas à la longueur de la description de liaison.
1	809B	L'identificateur local_device_id dans la description de la liaison ne correspond pas à la CPU.
1	80A1	Erreur de communication : <ul style="list-style-type: none"> <li>• La liaison indiquée n'a pas encore été établie.</li> <li>• La liaison indiquée est en train d'être coupée ; une transmission via cette liaison n'est pas possible.</li> <li>• L'interface est en cours de réinitialisation.</li> </ul>
1	80A3	Tentative de mettre fin à une liaison inexistante
1	80A4	L'adresse IP de la liaison au partenaire éloigné est invalide. Par exemple, l'adresse IP du partenaire éloigné est identique à l'adresse IP du partenaire local.
1	80A5	ID de liaison déjà utilisé
1	80A7	Erreur de communication : Vous avez appelé TDISCON avant que TSEND_C ne soit achevé.
1	80B2	Le paramètre CONNECT pointe sur un bloc de données qui a été généré avec le mot-clé UNLINKED.

ERROR	STATUS	Description
1	80B3	Paramètres incohérents : <ul style="list-style-type: none"> <li>• Erreur dans la description de la liaison</li> <li>• Le port local (paramètre local_tsap_id) est déjà présent dans une autre description de liaison.</li> <li>• ID dans la description de liaison différent de l'ID indiqué comme paramètre.</li> </ul>
1	80B4	Lors de l'utilisation de ISO sur TCP (connection_type = B#16#12) pour établir une liaison passive, le code d'erreur 80B4 vous avertit que le TSAP entré n'était pas conforme à l'une des exigences suivantes eu égard à l'adresse : <ul style="list-style-type: none"> <li>• Pour une longueur TSAP locale égale à 2 et une valeur ID TSAP égale à E0 ou E1 (hexadécimal) pour le premier octet, le deuxième octet doit être 00 ou 01.</li> <li>• Pour une longueur TSAP locale supérieure ou égale à 3 et une valeur ID TSAP égale à E0 ou E1 (hexadécimal) pour le premier octet, le deuxième octet doit être 00 ou 01 et tous les autres octets doivent être des caractères ASCII valides.</li> <li>• Pour une longueur TSAP locale supérieure ou égale à 3 et une valeur ID TSAP différente de E0 et E1 (hexadécimal) pour le premier octet, tous les octets de l'ID TSAP doivent être des caractères ASCII valides.</li> </ul> Les caractères ASCII valides ont des valeurs d'octet allant de 20 à 7E (hexadécimal).
1	80B7	Type de données et/ou longueur des données émises incompatibles avec la zone de la CPU partenaire dans laquelle elles doivent être écrites.
1	80C3	Toutes les ressources de liaison sont utilisées.
1	80C4	Erreur de communication temporaire : <ul style="list-style-type: none"> <li>• La liaison ne peut pas être établie en ce moment.</li> <li>• L'interface est en train de recevoir de nouveaux paramètres.</li> <li>• La liaison configurée est en cours de suppression par TDISCON.</li> </ul>
1	8722	Paramètre CONNECT : Zone source invalide : la zone n'existe pas dans le DB.
1	873A	Paramètre CONNECT : Impossible d'accéder à la description de la liaison (DB non disponible par exemple)
1	877F	Paramètre CONNECT : Erreur interne telle qu'une référence ANY invalide
1	893A	Le paramètre contient le numéro d'un DB qui n'est pas chargé.

## Protocoles de communication Ethernet

Chaque CPU comprend un port PROFINET intégré qui prend en charge la communication PROFINET standard. Les instructions TSEND\_C, TRCV\_C, TSEND et TRCV prennent toutes en charge les protocoles Ethernet TCP et ISO sur TCP.

Reportez-vous à "Configuration des appareils, Configuration du routage local/partenaire (Page 136)" pour plus d'informations.

## Voir aussi

Paramètres pour la liaison PROFINET (Page 138)

## TCON, TDISCON, TSEND et TRCV

### Communication Ethernet à l'aide des protocoles TCP et ISO sur TCP

---

#### Remarque

#### Instructions TSEND\_C et TRCV\_C

Pour simplifier la programmation de la communication PROFINET/Ethernet, l'instruction TSEND\_C et l'instruction TRCV\_C combinent les fonctionnalités des instructions TCON, TDISCON, TSEND et TRCV :

- TSEND\_C combine les instructions TCON, TDISCON et TSEND.
  - TRCV\_C combine les instructions TCON, TDISCON et TRCV.
- 

Les instructions suivantes gèrent le processus de communication :

- TCON établit une liaison TCP/IP entre un PC client et serveur (CPU).
- TSEND et TRCV envoient et reçoivent les données.
- TDISCON coupe la liaison.

La taille de données minimale que vous pouvez envoyer (TSEND) ou recevoir (TRCV) est un octet ; la taille maximale est 8192 octets. TSEND n'accepte pas l'envoi de données à partir d'adresses booléennes et TRCV ne reçoit pas de données dans des adresses booléennes. Pour plus d'informations sur le transfert de données à l'aide de ces instructions, reportez-vous au paragraphe sur la cohérence des données (Page 163).

TCON, TDISCON, TSEND et TRCV opèrent de manière asynchrone, ce qui signifie que le traitement de la tâche s'étend sur plusieurs exécutions de l'instruction. Par exemple, vous lancez une tâche pour la configuration et l'établissement d'une liaison en exécutant une instruction TCON avec le paramètre REQ = 1. Puis, vous réexécutez TCON pour surveiller l'avancement de la tâche et tester l'achèvement de la tâche via le paramètre DONE.

Le tableau suivant montre les relations entre BUSY, DONE et ERROR. Servez-vous-en pour déterminer l'état de la tâche en cours.

Tableau 10- 7 Interactions entre les paramètres BUSY, DONE et ERROR

BUSY	DONE	ERROR	Description
VRAI	Sans objet	Sans objet	La tâche est en cours de traitement.
FAUX	VRAI	FAUX	La tâche a été achevée avec succès.
FAUX	FAUX	VRAI	La tâche s'est terminée avec une erreur. La cause de l'erreur figure dans le paramètre STATUS.
FAUX	FAUX	FAUX	Aucune nouvelle tâche n'a été affectée.

## TCON et TDISCON

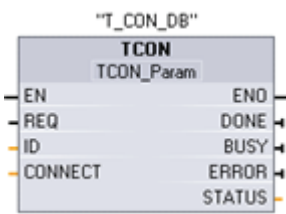
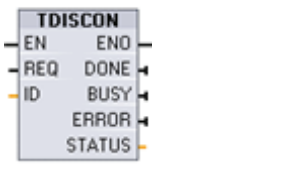
### Remarque

#### Initialisation des paramètres de communication

Une fois que vous avez inséré l'instruction TCON, utilisez les "Propriétés" de l'instruction (Page 136) pour configurer les paramètres de communication. Lorsque vous entrez les paramètres pour les partenaires de communication dans la fenêtre d'inspection, STEP 7 entre les données correspondantes dans le DB d'instance pour l'instruction.

Si vous voulez utiliser un DB multi-instance, vous devez configurer le DB manuellement dans les deux CPU.

Tableau 10- 8 Instructions TCON et TDISCON

CONT/LOG		Description
	<pre>"TCON_DB" (   req:=_bool_in_,   ID:=_undef_in_,   done=&gt;_bool_out_,   busy=&gt;_bool_out_,   error=&gt;_bool_out_,   status=&gt;_word_out_,   connect:= struct inout );</pre>	TCP et ISO sur TCP : TCON établit une liaison de communication de la CPU vers un partenaire de communication.
	<pre>"TDISCON_DB" (   req:=_bool_in_,   ID:=_word_in_,   done=&gt;_bool_out_,   busy=&gt;_bool_out_,   error=&gt;_bool_out_,   status=&gt;_word_out_);</pre>	TCP et ISO sur TCP : TDISCON met fin à une liaison de communication de la CPU vers un partenaire de communication.

<sup>1</sup> STEP 7 crée automatiquement le DB lorsque vous insérez l'instruction.

Tableau 10- 9 Types de données pour les paramètres de TCON et TDISCON

Paramètre et type	Type de données	Description
REQ	IN	Bool
ID	IN	CONN_OUC (Word)
CONNECT (TCON)	IN_OUT	TCON_Param

Paramètre et type		Type de données	Description
DONE	OUT	Bool	<ul style="list-style-type: none"> <li>0: La tâche n'a pas encore commencé ou est encore en cours d'exécution.</li> <li>1: Tâche achevée sans erreur</li> </ul>
BUSY	OUT	Bool	<ul style="list-style-type: none"> <li>0: Tâche achevée</li> <li>1: La tâche n'est pas encore achevée. Il n'est pas possible de déclencher une nouvelle tâche.</li> </ul>
ERROR	OUT	Bool	Paramètre d'état aux valeurs suivantes : <ul style="list-style-type: none"> <li>0: Pas d'erreur</li> <li>1: Une erreur s'est produite pendant le traitement. STATUS contient des informations détaillées sur le type d'erreur.</li> </ul>
STATUS	OUT	Word	Information d'état incluant l'information d'erreur (voir les codes d'erreur pour ERROR et STATUS dans le tableau ci-après)

Les deux partenaires de communication exécutent l'instruction TCON pour configurer et établir la liaison de communication. Vous vous servez de paramètres pour indiquer les partenaires de noeud d'extrémité actif et passif. Une fois la liaison configurée et établie, elle est automatiquement conservée et surveillée par la CPU.

Si la liaison est interrompue en raison d'une coupure de ligne ou à cause du partenaire de communication éloigné par exemple, le partenaire actif tente de rétablir la liaison configurée. Vous n'avez pas besoin de réexécuter TCON.

Une liaison existante est coupée et la liaison configurée est supprimée lors de l'exécution de l'instruction TDISCON ou lorsque la CPU passe à l'état ARRET. Vous devez réexécuter TCON pour reconfigurer et rétablir la liaison.

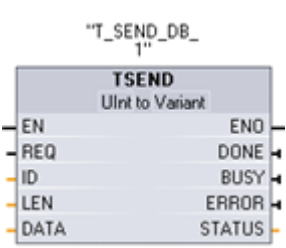
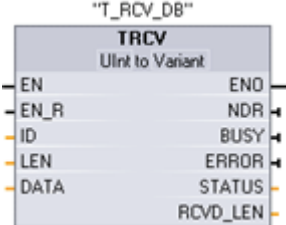
Tableau 10- 10 Codes d'erreur pour les paramètres ERROR et STATUS de TCON et TDISCON

ERROR	STATUS	Description
0	0000	La liaison a été établie avec succès.
0	7000	Pas de traitement de tâche actif
0	7001	Démarrage du traitement de la tâche ; établissement de la liaison (TCON) ou interruption de la liaison (TDISCON)
0	7002	Appel intermédiaire (REQ sans objet) ; établissement de la liaison (TCON) ou interruption de la liaison (TDISCON)
1	8086	Le paramètre ID se situe hors de la plage d'adresses permise.
1	8087	TCON : Nombre maximal de liaisons atteint ; aucune liaison supplémentaire n'est possible.
1	809B	TCON : L'identificateur local_device_id dans la description de la liaison ne correspond pas à la CPU.
1	80A1	TCON : Liaison ou port déjà occupé par un utilisateur
1	80A2	TCON : Port local ou éloigné occupé par le système
1	80A3	Tentative de rétablir une liaison existante (TCON) ou de mettre fin à une liaison inexistante (TDISCON)

ERROR	STATUS	Description
1	80A4	TCON : L'adresse IP du noeud d'extrémité de la liaison éloignée est invalide ; il est possible qu'elle corresponde à l'adresse IP locale.
1	80A5	TCON : ID de liaison déjà utilisé
1 ( )	80A7	TCON : Erreur de communication : Vous avez appelé TDISCON avant que TCON ne soit achevé. TDISCON doit d'abord couper complètement la liaison référencée par l'ID.
1	80B4	TCON : Lors de l'utilisation de ISO sur TCP ( <code>connection_type = B#16#12</code> ) pour établir une liaison passive, le code d'erreur 80B4 vous avertit que le TSAP entré n'était pas conforme à l'une des exigences suivantes eu égard à l'adresse : <ul style="list-style-type: none"> <li>• Pour une longueur TSAP locale égale à 2 et une valeur ID TSAP égale à E0 ou E1 (hexadécimal) pour le premier octet, le deuxième octet doit être 00 ou 01.</li> <li>• Pour une longueur TSAP locale supérieure ou égale à 3 et une valeur ID TSAP égale à E0 ou E1 (hexadécimal) pour le premier octet, le deuxième octet doit être 00 ou 01 et tous les autres octets doivent être des caractères ASCII valides.</li> <li>• Pour une longueur TSAP locale supérieure ou égale à 3 et une valeur ID TSAP différente de E0 et E1 (hexadécimal) pour le premier octet, tous les octets de l'ID TSAP doivent être des caractères ASCII valides.</li> </ul> Les caractères ASCII valides ont des valeurs d'octet allant de 20 à 7E (hexadécimal).
1	80B6	TCON : Erreur de paramétrage dans le paramètre <code>connection_type</code>
1	80B7	TCON : Type de données et/ou longueur des données émises incompatibles avec la zone de la CPU partenaire dans laquelle elles doivent être écrites.
1)	80B8	TCON : Le paramètre dans la description de liaison locale et le paramètre ID sont différents.
1	80C3	TCON : Toutes les ressources de liaison sont utilisées.
1	80C4	Erreur de communication temporaire : <ul style="list-style-type: none"> <li>• La liaison ne peut pas être établie en ce moment (TCON).</li> <li>• La liaison configurée est en cours de suppression par TDISCON (TCON).</li> <li>• La liaison est en cours d'établissement (TDISCON).</li> <li>• L'interface est en train de recevoir de nouveaux paramètres (TCON et TDISCON).</li> </ul>

## TSEND et TRCV

Tableau 10- 11 Instructions TSEND et TRCV

CONT/LOG	SCL	Description
<p>"T_SEND_DB_1"</p> 	<pre>"TSEND_DB" (   req:=_bool_in_,   ID:=_word_in_,   len:=_uint_in_,   done=&gt;_bool_out_,   busy=&gt;_bool_out_,   error=&gt;_bool_out_,   status=&gt;_word_out_,   data:=_variant_inout_);</pre>	<p>TCP et ISO sur TCP : TSEND envoie des données par le biais d'une liaison de communication de la CPU vers une station partenaire.</p>
<p>"T_RCV_DB"</p> 	<pre>"TRCV_DB" (   en_r:=_bool_in_,   ID:=_word_in_,   len:=_uint_in_,   ndr=&gt;_bool_out_,   busy=&gt;_bool_out_,   error=&gt;_bool_out_,   status=&gt;_word_out_,   rcvd_len=&gt;_uint_out_,   data:=_variant_inout_);</pre>	<p>TCP et ISO sur TCP : TRCV reçoit des données par le biais d'une liaison de communication d'une station partenaire vers la CPU.</p>

1 STEP 7 crée automatiquement le DB lorsque vous insérez l'instruction.

Tableau 10- 12 Types de données pour les paramètres de TSEND et TRCV

Paramètre et type		Type de données	Description
REQ	IN	Bool	TSEND : Lance la tâche d'émission en présence d'un front montant. Les données sont prises dans la zone indiquée par DATA et LEN.
EN_R	IN	Bool	TRCV : Valide la réception par la CPU ; lorsque EN_R = 1, TRCV est prêt à recevoir. La tâche de réception est traitée.
ID	IN	CONN_OUC (Word)	Référence à la liaison associée. L'ID doit être identique au paramètre ID associé dans la description de la liaison locale. Plage de valeurs : W#16#0001 à W#16#0FFF
LEN	IN	UInt	Nombre maximum d'octets à envoyer (TSEND) ou à recevoir (TRCV) : <ul style="list-style-type: none"> <li>Valeur par défaut = 0 : Le paramètre DATA détermine la longueur des données à envoyer (TSEND) ou à recevoir (TRCV).</li> <li>Mode ad hoc = 65535 : Une longueur de données variable est définie pour la réception (TRCV).</li> </ul>
DATA	IN_OUT	Variant	Pointeur désignant la zone de données à envoyer (TSEND) ou à recevoir (TRCV). La zone de données contient l'adresse et la longueur. L'adresse fait référence à de la mémoire I, de la mémoire Q, de la mémoire M ou à un DB.

Paramètre et type		Type de données	Description
DONE	OUT	Bool	TSEND: <ul style="list-style-type: none"> <li>0: La tâche n'a pas encore commencé ou est encore en cours d'exécution.</li> <li>1: Tâche exécutée sans erreur</li> </ul>
NDR	OUT	Bool	TRCV: <ul style="list-style-type: none"> <li>NDR = 0 : Tâche pas encore commencée ou encore en cours d'exécution</li> <li>NDR = 1 : Tâche achevée avec succès</li> </ul>
BUSY	OUT	Bool	<ul style="list-style-type: none"> <li>BUSY = 1 : La tâche n'est pas encore achevée. Il n'est pas possible de déclencher une nouvelle tâche.</li> <li>BUSY = 0 : Tâche achevée</li> </ul>
ERROR	OUT	Bool	ERROR = 1 : Une erreur s'est produite pendant le traitement. STATUS contient des informations détaillées sur le type d'erreur.
STATUS	OUT	Word	Information d'état incluant l'information d'erreur (voir les codes d'erreur pour ERROR et STATUS dans le tableau ci-après)
RCVD_LEN	OUT	Int	TRCV : Quantité de données effectivement reçue en octets

---

#### Remarque

L'instruction TSEND requiert un front montant du paramètre d'entrée REQ pour démarrer une tâche d'émission. Le paramètre BUSY est mis à 1 pendant le traitement. L'achèvement de la tâche d'émission est signalé par la mise à 1 du paramètre DONE ou ERROR pendant un cycle. Pendant ce temps, aucun front montant du paramètre d'entrée REQ n'est pris en compte.

---

#### Fonctionnement de TRCV

L'instruction TRCV écrit les données reçues dans une zone de réception qui est définie par les deux variables suivantes :

- Pointeur désignant le début de la zone
- Longueur de la zone ou valeur fournie à l'entrée LEN si différente de 0

---

#### Remarque

Avec la valeur par défaut du paramètre LEN (LEN = 0), le paramètre DATA est utilisé pour déterminer la longueur des données transmises. Assurez-vous que les données DATA envoyées par l'instruction TSEND ont la même taille que le paramètre DATA de l'instruction TRCV.

---

Dès que toutes les données de la tâche ont été reçues, TRCV les transfère dans la zone de réception et met NDR à 1.



Tableau 10- 13 Entrée des données dans la zone de réception

Variante de protocole	Entrée des données dans la zone de réception	Paramètre "connection_type"	Valeur du paramètre LEN	Valeur du paramètre RCVD_LEN (octets)
TCP	Mode ad hoc	B#16#11	65535	1 à 1472
TCP	Réception de données avec longueur indiquée	B#16#11	0 (recommandé) ou 1 à 8192, 65535 excepté	1 à 8192
ISO sur TCP	Mode ad hoc	B#16#12	65535	1 à 1472
ISO sur TCP	Gestion par le protocole	B#16#12	0 (recommandé) ou 1 à 8192, 65535 excepté	1 à 8192

**Remarque**

**Mode ad hoc**

Le "mode ad hoc" existe avec les variantes de protocole TCP et ISO sur TCP. Vous paramétrez le "mode ad hoc" en affectant 65535 au paramètre LEN. La zone de réception est identique à la zone formée par DATA. La longueur des données reçues est transmise au paramètre RCVD\_LEN. Tout de suite après la réception d'un bloc de données, TRCV entre les données dans la zone de réception et met NDR à 1.

Si vous stockez les données dans un DB "optimisé" (symbolique uniquement), vous ne pouvez recevoir les données qu'en tableaux de types de données Byte, Char, USInt et SInt.

**Remarque**

**Importation dans le S7-1200 de projets STEP 7 S7-300/400 utilisant le "mode ad hoc"**

Dans les projets STEP 7 S7-300/400, le "mode ad hoc" est sélectionné par affectation de 0 au paramètre LEN. Dans le S7-1200, vous paramétrez le "mode ad hoc" en affectant 65535 au paramètre LEN.

Si vous importez dans le S7-1200 un projet STEP 7 S7-300/400 utilisant le "mode ad hoc", vous devez donner la valeur 65535 au paramètre LEN.

Tableau 10- 14 Codes d'erreur pour les paramètres ERROR et STATUS de TSEND et TRCV

ERROR	STATUS	Description
0	0000	<ul style="list-style-type: none"> <li>Tâche d'émission achevée sans erreur (TSEND)</li> <li>Nouvelles données acceptées : La longueur en cours des données reçues est indiquée dans RCVD_LEN (TRCV).</li> </ul>
0	7000	<ul style="list-style-type: none"> <li>Pas de traitement de tâche actif (TSEND)</li> <li>Bloc pas prêt à recevoir (TRCV)</li> </ul>
0	7001	<ul style="list-style-type: none"> <li>Démarrage du traitement de la tâche, données en cours d'envoi : le système d'exploitation accède aux données dans la zone d'émission DATA pendant ce traitement (TSEND).</li> <li>Bloc prêt à recevoir ; une tâche de réception a été activée (TRCV).</li> </ul>

ERROR	STATUS	Description
0	7002	<ul style="list-style-type: none"> <li>Exécution intermédiaire de l'instruction (REQ sans objet), tâche en cours de traitement : le système d'exploitation accède aux données dans la zone d'émission DATA pendant ce traitement (TSEND).</li> <li>Exécution intermédiaire de l'instruction, tâche de réception en cours de traitement : des données sont écrites dans la zone de réception pendant ce traitement. C'est pourquoi une erreur pourrait être à l'origine de données incohérentes dans la zone de réception (TRCV).</li> </ul>
1	8085	<ul style="list-style-type: none"> <li>Le paramètre LEN est supérieur à la plus grande valeur autorisée (TSEND et TRCV).</li> <li>Paramètre LEN ou DATA modifié depuis la première exécution de l'instruction (TRCV).</li> </ul>
1	8086	Le paramètre ID ne se situe pas dans la plage d'adresses permise.
1	8088	Le paramètre LEN est plus grand que la zone de mémoire indiquée dans DATA.
1	80A1	Erreur de communication : <ul style="list-style-type: none"> <li>La liaison indiquée n'a pas encore été établie (TSEND et TRCV).</li> <li>La liaison indiquée est en train d'être coupée. Une tâche d'émission ou de réception via cette liaison n'est pas possible (TSEND et TRCV).</li> <li>L'interface est en cours de réinitialisation (TSEND).</li> <li>L'interface est en train de recevoir de nouveaux paramètres (TRCV).</li> </ul>
1	80C3	Manque interne de ressources : Un bloc avec cet ID est déjà en cours de traitement dans une classe de priorité différente.
1	80C4	Erreur de communication temporaire : <ul style="list-style-type: none"> <li>La liaison au partenaire de communication ne peut pas être établie en ce moment.</li> <li>L'interface est en train de recevoir de nouveaux paramètres ou la liaison est en cours d'établissement.</li> </ul>

### Protocoles de communication Ethernet

Chaque CPU comprend un port PROFINET intégré qui prend en charge la communication PROFINET standard. Les instructions TSEND\_C, TRCV\_C, TSEND et TRCV prennent toutes en charge les protocoles Ethernet TCP et ISO sur TCP.

Reportez-vous à "Configuration des appareils, Configuration du routage local/partenaire (Page 136)" pour plus d'informations.

### Voir aussi

Paramètres pour la liaison PROFINET (Page 138)

### 10.2.2.5 UDP

UDP est un protocole standard décrit par RFC 768 : User Datagram Protocol (protocole de datagramme utilisateur). UDP fournit un mécanisme permettant à une application d'envoyer un datagramme à une autre application. Toutefois, la distribution des données n'est pas garantie. Ce protocole présente les caractéristiques suivantes :

- Protocole de communication rapide, car très proche du matériel
- Convenant à des quantités faibles à moyennes de données (jusqu'à 2048 octets)
- UDP est un protocole de commande de transport plus simple que TCP avec une couche fine présentant des trafics de service faibles.
- Utilisable très souplement avec de nombreux systèmes tiers
- Apte au routage
- Utilisation de numéros de port pour diriger les datagrammes
- Il n'y a pas d'accusé de réception des messages. C'est à l'application qu'il incombe de gérer la reprise sur erreur et la sécurité.
- Un effort de programmation est nécessaire pour la gestion des données en raison de l'interface de programmation EMISSION / RECEPTION.

UDP prend en charge la diffusion générale. Pour utiliser la diffusion générale, vous devez configurer la partie "adresse IP" du paramètre ADDR. Ainsi, une CPU ayant l'adresse IP 192.168.2.10 et le masque de sous-réseau 255.255.255.0 utilisera l'adresse de diffusion générale 192.168.2.255.

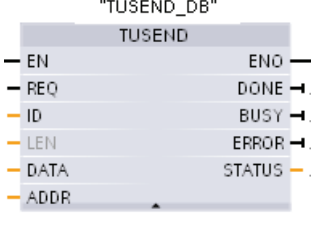

### TUSEND et TURCV

Les instructions suivantes gèrent le processus de communication UDP :

- TCON établit la communication entre le PC client et serveur (CPU).
- TUSEND et TURCV envoient et reçoivent les données.
- TDISCON coupe la communication entre le client et le serveur.

Reportez-vous à TCON, TDISCON, TSEND et TRCV (Page 467) dans la rubrique "TCP et ISO sur TCP" pour plus d'informations sur les instructions de communication TCON et TDISCON.

Tableau 10- 15 Instructions TUSEND et TURCV

CONT/LOG	SCL	Description
<p>"TUSEND_DB"</p> 	<pre>"TUSEND_DB" (     req:=_bool_in_,     ID:=_word_in_,     len:=_uint_in_,     done=&gt;_bool_out_,     busy=&gt;_bool_out_,     error=&gt;_bool_out_,     status=&gt;_word_out_,     data:=_variant_inout_);</pre>	<p>L'instruction TUSEND envoie des données via UDP au partenaire éloigné indiqué par le paramètre ADDR.</p> <p>Pour démarrer la tâche d'envoi des données, appelez l'instruction TUSEND avec REQ = 1.</p>
<p>"TURCV_DB"</p> 	<pre>"TURCV_DB" (     en_r:=_bool_in_,     ID:=_word_in_,     len:=_uint_in_,     ndr=&gt;_bool_out_,     busy=&gt;_bool_out_,     error=&gt;_bool_out_,     status=&gt;_word_out_,     rcvd_len=&gt;_uint_out_,     data:=_variant_inout_);</pre>	<p>L'instruction TURCV reçoit des données via UDP. Le paramètre ADDR indique l'adresse de l'émetteur. Lorsque l'exécution de TURCV s'est achevée avec succès, le paramètre ADDR contient l'adresse du partenaire éloigné (l'émetteur).</p> <p>TURCV ne prend pas en charge le mode ad hoc.</p> <p>Pour démarrer la tâche de réception des données, appelez l'instruction TURCV avec EN_R = 1.</p>

1 STEP 7 crée automatiquement le DB lorsque vous insérez l'instruction.

TCON, TDISCON, TUSEND et TURCV opèrent de manière asynchrone, ce qui signifie que le traitement de la tâche s'étend sur plusieurs exécutions de l'instruction.

Tableau 10- 16 Types de données pour les paramètres de TUSEND et TURCV

Paramètre et type	Type de données	Description	
REQ (TUSEND)	IN	Bool	Lance la tâche d'émission en présence d'un front montant. Les données sont prises dans la zone indiquée par DATA et LEN.
EN_R (TURCV)	IN	Bool	<ul style="list-style-type: none"> <li>0: La CPU ne peut pas recevoir.</li> <li>1: Valide la réception par la CPU. L'instruction TURCV est prête à recevoir et la tâche de réception est traitée.</li> </ul>
ID	IN	Word	Référence à la liaison associée entre le programme utilisateur et la couche de communication du système d'exploitation. ID doit être identique au paramètre ID associé dans la description de la liaison locale. Plage de valeurs : W#16#0001 à W#16#0FFF.
LEN	IN	UInt	Nombre d'octets à envoyer (TUSEND) ou à recevoir (TURCV) : <ul style="list-style-type: none"> <li>Default = 0. La paramètre DATA détermine la longueur des données à envoyer ou à recevoir.</li> <li>Sinon, plage de valeurs : 1 à 1472</li> </ul>
DONE (TUSEND)	IN	Bool	Paramètre d'état DONE (TUSEND) : <ul style="list-style-type: none"> <li>0: La tâche n'a pas encore commencé ou est encore en cours d'exécution.</li> <li>1: Tâche achevée sans erreur</li> </ul>

Paramètre et type		Type de données	Description
NDR (TURCV)	OUT	Bool	Paramètre d'état NDR (TURCV) : <ul style="list-style-type: none"> <li>0: La tâche n'a pas encore commencé ou est encore en cours d'exécution.</li> <li>1: La tâche a été achevée avec succès.</li> </ul>
BUSY	OUT	Bool	<ul style="list-style-type: none"> <li>1: La tâche n'est pas encore achevée. Il n'est pas possible de déclencher une nouvelle tâche.</li> <li>0: La tâche est achevée.</li> </ul>
ERROR	OUT	Bool	Paramètre d'état aux valeurs suivantes : <ul style="list-style-type: none"> <li>0: Pas d'erreur</li> <li>1: Une erreur s'est produite pendant le traitement. STATUS contient des informations détaillées sur le type d'erreur.</li> </ul>
STATUS	OUT	Word	Information d'état incluant l'information d'erreur (voir les codes d'erreur pour ERROR et STATUS dans le tableau ci-après)
DATA	IN_OUT	Variante	Adresse de la zone d'émission (TUSEND) ou de la zone de réception (TURCV) : <ul style="list-style-type: none"> <li>la mémoire image des entrées</li> <li>la mémoire image des sorties</li> <li>un memento</li> <li>un bloc de données</li> </ul>
ADDR	IN_OUT	Variante	Pointeur désignant l'adresse du récepteur (pour TUSEND) ou de l'émetteur (pour TURCV) (par exemple, P#DB100.DBX0.0 byte 8). Le pointeur peut désigner n'importe quelle zone de mémoire. Une structure de 8 octets du type suivant est requise : <ul style="list-style-type: none"> <li>Les 4 premiers octets contiennent l'adresse IP éloignée.</li> <li>Les 2 octets suivants indiquent le numéro du port partenaire.</li> <li>Les 2 derniers octets sont réservés.</li> </ul>

L'état de la tâche est indiqué dans les paramètres de sortie BUSY et STATUS. STATUS correspond au paramètre de sortie RET\_VAL des instructions à fonctionnement asynchrone.

Le tableau suivant montre les relations entre BUSY, DONE (TUSEND), NDR (TURCV) et ERROR. A l'aide de ce tableau, vous pouvez déterminer l'état en cours de l'instruction (TUSEND ou TURCV) ou le moment où l'opération d'émission / réception est achevée.

Tableau 10- 17 Etat des paramètres BUSY, DONE (TUSEND) / NDR (TURCV) et ERROR

BUSY	DONE / NDR	ERROR	Description
VRAI	Sans objet	Sans objet	La tâche est en cours de traitement.
FAUX	VRAI	FAUX	La tâche a été achevée avec succès.

BUSY	DONE / NDR	ERROR	Description
FAUX	FAUX	VRAI	La tâche s'est terminée avec une erreur. La cause de l'erreur figure dans le paramètre STATUS.
FAUX	FAUX	FAUX	Aucune (nouvelle) tâche n'a été affectée à l'instruction.

<sup>1</sup> En raison du traitement asynchrone des instructions : Pour TUSEND, vous devez faire en sorte que les données dans la zone d'émission restent cohérentes jusqu'à ce que le paramètre DONE ou le paramètre ERROR prenne la valeur VRAI. Pour TURCV, les données dans la zone de réception ne sont cohérentes que lorsque le paramètre NDR prend la valeur VRAI.

Tableau 10- 18 Codes d'erreur pour les paramètres ERROR et STATUS de TUSEND et TURCV

ERROR	STATUS	Description
0	0000	<ul style="list-style-type: none"> <li>Tâche d'émission achevée sans erreur (TUSEND).</li> <li>De nouvelles données ont été acceptées : La longueur en cours des données reçues est indiquée dans RCVD_LEN (TURCV).</li> </ul>
0	7000	<ul style="list-style-type: none"> <li>Pas de traitement de tâche actif (TUSEND)</li> <li>Bloc pas prêt à recevoir (TURCV)</li> </ul>
0	7001	<ul style="list-style-type: none"> <li>Démarrage du traitement de la tâche, données en cours d'envoi (TUSEND) : le système d'exploitation accède aux données dans la zone d'émission DATA pendant ce traitement.</li> <li>Bloc prêt à recevoir ; une tâche de réception a été activée (TURCV).</li> </ul>
0	7002	<ul style="list-style-type: none"> <li>Exécution intermédiaire de l'instruction (REQ sans objet), tâche en cours de traitement (TUSEND) : le système d'exploitation accède aux données dans la zone d'émission DATA pendant ce traitement.</li> <li>Exécution intermédiaire de l'instruction, tâche en cours de traitement : l'instruction TURCV écrit les données dans la zone de réception pendant ce traitement. C'est pourquoi une erreur pourrait être à l'origine de données incohérentes dans la zone de réception.</li> </ul>
1	8085	Le paramètre LEN est supérieur à la plus grande valeur autorisée ou a la valeur 0 (TUSEND) ou vous avez modifié la valeur du paramètre LEN ou DATA depuis la première exécution de l'instruction (TURCV).
1	8086	Le paramètre ID ne se situe pas dans la plage d'adresses permise.
1	8088	<ul style="list-style-type: none"> <li>Le paramètre LEN est plus grand que la zone d'émission (TUSEND) ou la zone de réception (TURCV) indiquée dans DATA.</li> <li>La zone de réception est trop petite (TURCV).</li> </ul>
1	8089	Le paramètre ADDR ne pointe pas sur un bloc de données.
1	80A1	<p>Erreur de communication :</p> <ul style="list-style-type: none"> <li>La liaison indiquée entre le programme utilisateur et la couche de communication du système d'exploitation n'a pas encore été établie.</li> <li>La liaison indiquée entre le programme utilisateur et la couche de communication du système d'exploitation est en train d'être coupée. Une tâche d'émission (TUSEND) ou de réception (TURCV) via cette liaison n'est pas possible.</li> <li>L'interface est en cours de réinitialisation.</li> </ul>

ERROR	STATUS	Description
1	80A4	L'adresse IP du noeud d'extrémité de la liaison éloignée est invalide ; il est possible qu'elle corresponde à l'adresse IP locale (TUSEND).
1	80B3	<ul style="list-style-type: none"> <li>La variante de protocole paramétrée (paramètre connection_type dans la description de la liaison) n'est pas UDP. Veuillez utiliser l'instruction TSEND ou TRCV.</li> <li>Paramètre ADDR : paramétrage invalide pour le numéro de port (TUSEND)</li> </ul>
1	80C3	<ul style="list-style-type: none"> <li>Un bloc avec cet ID est déjà en cours de traitement dans une classe de priorité différente.</li> <li>Manque interne de ressources</li> </ul>
1	80C4	Erreur de communication temporaire : <ul style="list-style-type: none"> <li>La liaison entre le programme utilisateur et la couche de communication du système d'exploitation ne peut pas être établie en ce moment (TUSEND).</li> <li>L'interface est en train de recevoir de nouveaux paramètres (TUSEND).</li> <li>La liaison est en cours de réinitialisation (TURCV).</li> </ul>

## Protocoles de communication Ethernet

Chaque CPU comprend un port PROFINET intégré qui prend en charge la communication PROFINET standard. Les instructions TUSEND et TURCV prennent en charge le protocole Ethernet UDP.

Reportez-vous à "Configuration du routage local/partenaire" (Page 136) dans le chapitre "Configuration des appareils" pour plus d'informations.

## Fonctionnement

Les deux partenaires sont passifs dans la communication UDP. Des valeurs initiales de paramètre typiques pour le type de données "TCON\_Param" sont présentées dans le tableau suivant. Les numéros de port (LOCAL\_TSAP\_ID) sont écrits en format à deux octets. Tous les ports sont autorisés à l'exception de 161, 34962, 34963 et 34964.

Tableau 10- 19 "Valeurs de paramètres du type de données "TCON\_Param"

Instruction TCON		"UDP Conn DB" de TCON																																																																																																	
		<table border="1"> <thead> <tr> <th>Nom</th> <th>Type de données</th> <th>Décalage</th> <th>Valeur initiale</th> <th>Commentaire</th> </tr> </thead> <tbody> <tr> <td>1</td> <td>Static</td> <td></td> <td></td> <td></td> </tr> <tr> <td>2</td> <td>Params</td> <td>TCON_Param</td> <td>0.0</td> <td></td> </tr> <tr> <td>3</td> <td>BLOCK_LENGTH</td> <td>UInt</td> <td>0.0</td> <td>-4</td> <td>byte length of SDT</td> </tr> <tr> <td>4</td> <td>ID</td> <td>CONN_OUC</td> <td>2.0</td> <td>1</td> <td>reference to the connection</td> </tr> <tr> <td>5</td> <td>CONNECTION_TYPE</td> <td>USInt</td> <td>4.0</td> <td>19</td> <td>17: TCP/IP, 18: ISO on TCP</td> </tr> <tr> <td>6</td> <td>ACTIVE_EST</td> <td>Bool</td> <td>5.0</td> <td>false</td> <td>active/passive connection establishment</td> </tr> <tr> <td>7</td> <td>LOCAL_DEVICE_ID</td> <td>USInt</td> <td>6.0</td> <td>1</td> <td>1: local I/E interface</td> </tr> <tr> <td>8</td> <td>LOCAL_TSAP_ID_LEN</td> <td>USInt</td> <td>7.0</td> <td>2</td> <td>byte length of local TSAP id/port number</td> </tr> <tr> <td>9</td> <td>REM_SUBNET_ID_LEN</td> <td>USInt</td> <td>8.0</td> <td>0</td> <td>byte length of remote subnet id</td> </tr> <tr> <td>10</td> <td>REM_STADDR_LEN</td> <td>USInt</td> <td>9.0</td> <td>0</td> <td>byte length of remote IP address</td> </tr> <tr> <td>11</td> <td>REM_TSAP_ID_LEN</td> <td>USInt</td> <td>10.0</td> <td>0</td> <td>byte length of remote port/TSAP id</td> </tr> <tr> <td>12</td> <td>NEXT_STADDR_LEN</td> <td>USInt</td> <td>11.0</td> <td>0</td> <td>byte length of next station address</td> </tr> <tr> <td>13</td> <td>LOCAL_TSAP_ID</td> <td>Array[1..16] of Byte</td> <td>12.0</td> <td></td> <td>TSAP id/local port number</td> </tr> <tr> <td>14</td> <td>LOCAL_TSAP_ID[1]</td> <td>Byte</td> <td></td> <td>#16#07</td> <td></td> </tr> <tr> <td>15</td> <td>LOCAL_TSAP_ID[2]</td> <td>Byte</td> <td></td> <td>#16#00</td> <td></td> </tr> </tbody> </table>					Nom	Type de données	Décalage	Valeur initiale	Commentaire	1	Static				2	Params	TCON_Param	0.0		3	BLOCK_LENGTH	UInt	0.0	-4	byte length of SDT	4	ID	CONN_OUC	2.0	1	reference to the connection	5	CONNECTION_TYPE	USInt	4.0	19	17: TCP/IP, 18: ISO on TCP	6	ACTIVE_EST	Bool	5.0	false	active/passive connection establishment	7	LOCAL_DEVICE_ID	USInt	6.0	1	1: local I/E interface	8	LOCAL_TSAP_ID_LEN	USInt	7.0	2	byte length of local TSAP id/port number	9	REM_SUBNET_ID_LEN	USInt	8.0	0	byte length of remote subnet id	10	REM_STADDR_LEN	USInt	9.0	0	byte length of remote IP address	11	REM_TSAP_ID_LEN	USInt	10.0	0	byte length of remote port/TSAP id	12	NEXT_STADDR_LEN	USInt	11.0	0	byte length of next station address	13	LOCAL_TSAP_ID	Array[1..16] of Byte	12.0		TSAP id/local port number	14	LOCAL_TSAP_ID[1]	Byte		#16#07		15	LOCAL_TSAP_ID[2]	Byte		#16#00	
Nom	Type de données	Décalage	Valeur initiale	Commentaire																																																																																															
1	Static																																																																																																		
2	Params	TCON_Param	0.0																																																																																																
3	BLOCK_LENGTH	UInt	0.0	-4	byte length of SDT																																																																																														
4	ID	CONN_OUC	2.0	1	reference to the connection																																																																																														
5	CONNECTION_TYPE	USInt	4.0	19	17: TCP/IP, 18: ISO on TCP																																																																																														
6	ACTIVE_EST	Bool	5.0	false	active/passive connection establishment																																																																																														
7	LOCAL_DEVICE_ID	USInt	6.0	1	1: local I/E interface																																																																																														
8	LOCAL_TSAP_ID_LEN	USInt	7.0	2	byte length of local TSAP id/port number																																																																																														
9	REM_SUBNET_ID_LEN	USInt	8.0	0	byte length of remote subnet id																																																																																														
10	REM_STADDR_LEN	USInt	9.0	0	byte length of remote IP address																																																																																														
11	REM_TSAP_ID_LEN	USInt	10.0	0	byte length of remote port/TSAP id																																																																																														
12	NEXT_STADDR_LEN	USInt	11.0	0	byte length of next station address																																																																																														
13	LOCAL_TSAP_ID	Array[1..16] of Byte	12.0		TSAP id/local port number																																																																																														
14	LOCAL_TSAP_ID[1]	Byte		#16#07																																																																																															
15	LOCAL_TSAP_ID[2]	Byte		#16#00																																																																																															

L'instruction TUSEND envoie des données par le biais d'UDP au partenaire éloigné indiqué dans le type de données "TADDR\_Param". L'instruction TURCV reçoit des données par le biais d'UDP. Lorsque l'exécution de TURCV s'est achevée avec succès, le type de données "TADDR\_Param" montre l'adresse du partenaire éloigné (l'émetteur).

Tableau 10- 20 "Valeurs de paramètres du type de données "TADDR\_Param"

Instruction TUSEND		"UDP ADDR DB" de TUSEND																																																								
		<table border="1"> <thead> <tr> <th colspan="5">Send_UDP_ADDR</th> </tr> <tr> <th>Nom</th> <th>Type de données</th> <th>Décalage</th> <th>Valeur initiale</th> <th>Commentaire</th> </tr> </thead> <tbody> <tr> <td>1</td> <td>Static</td> <td></td> <td></td> <td></td> </tr> <tr> <td>2</td> <td>Addr_Data</td> <td>TADDR_Param</td> <td>0.0</td> <td></td> </tr> <tr> <td>3</td> <td>REM_IP_ADDR</td> <td>Array[1..4] of USint</td> <td>0.0</td> <td>remote station address</td> </tr> <tr> <td>4</td> <td>REM_IP_ADDR[1]</td> <td>USint</td> <td>0</td> <td></td> </tr> <tr> <td>5</td> <td>REM_IP_ADDR[2]</td> <td>USint</td> <td>0</td> <td></td> </tr> <tr> <td>6</td> <td>REM_IP_ADDR[3]</td> <td>USint</td> <td>0</td> <td></td> </tr> <tr> <td>7</td> <td>REM_IP_ADDR[4]</td> <td>USint</td> <td>0</td> <td></td> </tr> <tr> <td>8</td> <td>REM_PORT_NR</td> <td>Ushort</td> <td>4.0</td> <td>remote port number</td> </tr> <tr> <td>9</td> <td>RESERVED</td> <td>Word</td> <td>6.0</td> <td>unused, has to be 0</td> </tr> </tbody> </table>		Send_UDP_ADDR					Nom	Type de données	Décalage	Valeur initiale	Commentaire	1	Static				2	Addr_Data	TADDR_Param	0.0		3	REM_IP_ADDR	Array[1..4] of USint	0.0	remote station address	4	REM_IP_ADDR[1]	USint	0		5	REM_IP_ADDR[2]	USint	0		6	REM_IP_ADDR[3]	USint	0		7	REM_IP_ADDR[4]	USint	0		8	REM_PORT_NR	Ushort	4.0	remote port number	9	RESERVED	Word	6.0	unused, has to be 0
Send_UDP_ADDR																																																										
Nom	Type de données	Décalage	Valeur initiale	Commentaire																																																						
1	Static																																																									
2	Addr_Data	TADDR_Param	0.0																																																							
3	REM_IP_ADDR	Array[1..4] of USint	0.0	remote station address																																																						
4	REM_IP_ADDR[1]	USint	0																																																							
5	REM_IP_ADDR[2]	USint	0																																																							
6	REM_IP_ADDR[3]	USint	0																																																							
7	REM_IP_ADDR[4]	USint	0																																																							
8	REM_PORT_NR	Ushort	4.0	remote port number																																																						
9	RESERVED	Word	6.0	unused, has to be 0																																																						

### 10.2.2.6 T\_CONFIG

L'instruction T\_CONFIG permet de modifier les paramètres de configuration IP du port PROFINET à partir du programme utilisateur, autorisant ainsi la modification ou la configuration permanente des fonctionnalités suivantes :

- Nom de station
- Adresse IP
- Masque de sous-réseau
- Adresse du routeur



### Remarque

Le bouton radio "Dériver l'adresse IP d'une autre source" (Page 486) dans la page "Adresses Ethernet" des "Propriétés" de la CPU vous permet de modifier l'adresse IP en ligne ou à l'aide de l'instruction "T\_CONFIG" après le chargement du programme. La méthode d'affectation de l'adresse IP concerne la CPU uniquement.

L'option "Dériver le nom d'appareil PROFINET d'une autre source" (Page 487) dans la page "Adresses Ethernet" des "Propriétés" de la CPU vous permet de modifier le nom d'appareil PROFINET en ligne ou à l'aide de l'instruction "T\_CONFIG" après le chargement du programme. La méthode d'affectation du nom d'appareil PROFINET concerne la CPU uniquement.

### ATTENTION

Une fois que vous avez utilisé T\_CONFIG pour modifier un paramètre de configuration IP, la CPU redémarre. Elle passe à l'état ARRET, effectue un démarrage à chaud et repasse à l'état MARCHE.

Les appareils de commande peuvent présenter des défaillances dans des situations non sûres et provoquer un fonctionnement inattendu des appareils pilotés pouvant entraîner la mort, des blessures graves et des dommages matériels importants.

Assurez-vous que votre processus entrera dans un état sûr lors du redémarrage de la CPU provoqué par l'exécution de l'instruction T\_CONFIG.

Tableau 10- 21 Instruction T\_CONFIG

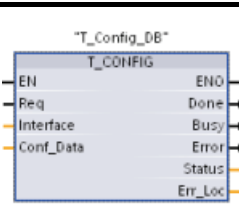
CONT/LOG	SCL	Description
	<pre>"T_CONFIG_DB" (   req:=_bool_in_,   interface:=_word_in_,   conf_Data:=_variant_in_,   done=&gt;_bool_out_,   busy=&gt;_bool_out_,   error=&gt;_bool_out_,   status=&gt;_dword_out_,   err_loc=&gt;_word_out_);</pre>	<p>Utilisez l'instruction T_CONFIG pour modifier les paramètres de configuration IP à partir de votre programme utilisateur.</p> <p>T_CONFIG fonctionne de manière asynchrone. Son exécution s'étend sur plusieurs appels.</p>

Tableau 10- 22 Types de données pour les paramètres

Paramètre et type	Type de données	Description
REQ Input	Bool	Démarre l'instruction en présence d'un front montant.
INTERFACE Input	HW_Interface	ID de l'interface de réseau
CONF_DATA Input	Variant	Référence à la structure des données de configuration ; CONF_DATA est défini par un type de données système (SDT).
DONE Output	Bool	<ul style="list-style-type: none"> <li>0: La tâche n'a pas encore commencé ou est encore en cours d'exécution.</li> <li>1: Tâche exécutée sans erreur</li> </ul>

Paramètre et type		Type de données	Description
BUSY	Output	Bool	<ul style="list-style-type: none"> <li>0: La tâche est achevée.</li> <li>1: La tâche n'est pas encore achevée. Il n'est pas possible de déclencher une nouvelle tâche.</li> </ul>
ERROR	Output	Bool	Paramètre d'état aux valeurs suivantes : <ul style="list-style-type: none"> <li>0: Pas d'erreur</li> <li>1: Une erreur s'est produite pendant le traitement. STATUS contient des informations détaillées sur le type d'erreur.</li> </ul>
STATUS	Output	DWord	Information d'état incluant l'information d'erreur (voir les codes d'erreur pour ERROR et STATUS dans le tableau ci-après)
ERR_LOC	Output	DWord	Emplacement de l'erreur (ID de bloc et ID de sous-bloc du paramètre erroné)

Les informations de configuration IP sont placées dans le bloc de données CONF\_DATA conjointement avec un pointeur Variant désignant le paramètre CONF\_DATA référencé ci-avant. L'exécution réussie de l'instruction T\_CONFIG se termine avec la transmission des données de configuration IP à l'interface de réseau. Les erreurs sont affectées au paramètre de sortie STATUS.

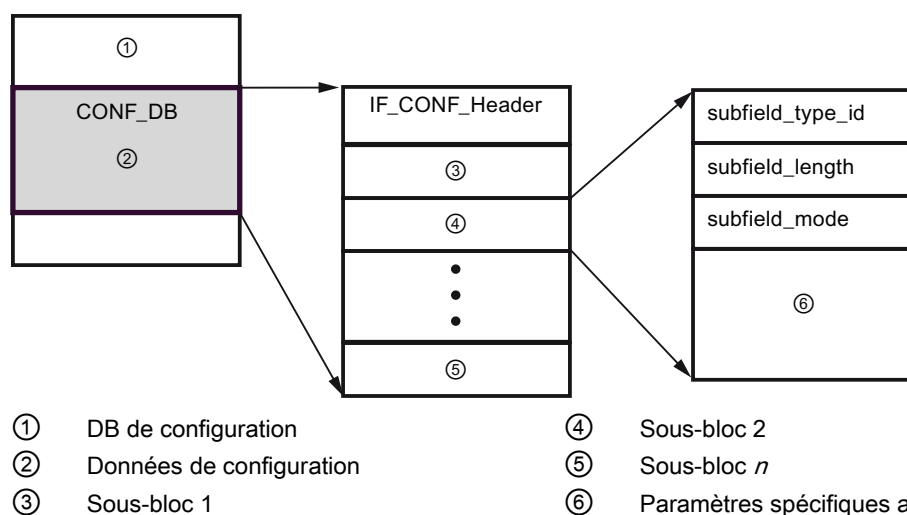
Tableau 10- 23 Codes d'erreur pour ERROR et STATUS

ERROR	STATUS	Description
0	00000000	Pas d'erreur
0	00700000	La tâche n'est pas achevée (BUSY = 1).
0	00700100	Début de l'exécution de la tâche
0	00700200	Appel intermédiaire (REQ non significatif)
1	C08xyy00	Défaillance générale
1	C0808000	Les paramètres LADDR d'identification de l'interface sont invalides.
1	C0808100	Les paramètres LADDR d'identification de l'interface correspondent à une interface matérielle non prise en charge.
1	C0808200	Paramètre CONF_DATA erroné : Le type de données du pointeur Variant ne correspond pas au type de données Byte.
1	C0808300	Paramètre CONF_DATA erroné : Le pointeur de zone n'est pas dans le DB du pointeur Variant.
1	C0808400	Paramètre CONF_DATA erroné : Le pointeur Variant a une longueur incorrecte.
1	C0808600	Réservé
1	C0808700	Incohérence dans la longueur du bloc de données CONF_DATA par rapport à la configuration IP
1	C0808800	Les paramètres field_type_id du bloc de données CONF_DATA sont invalides (seul field_type_id = 0 est autorisé).
1	C0808900	Les paramètres field_type_id du bloc de données CONF_DATA sont invalides ou ont été utilisés plusieurs fois.
1	C0808A00	Erreurs dans la longueur LEN des paramètres de configuration IP ou dans subfield_cnt
1	C0808B00	Le paramètre ID de la configuration IP est invalide ou n'est pas pris en charge.

ERROR	STATUS	Description
1	C0808C00	Le sous-bloc de la configuration IP est mal placé (sous-bloc erroné, ordre erroné ou utilisation à plusieurs reprises).
1	C0808D00	La longueur LEN d'un sous-bloc est invalide.
1	C0808E00	La valeur du paramètre dans le mode d'un sous-bloc est invalide.
1	C0808F00	Conflit de sous-bloc entre la configuration IP et un sous-bloc précédent.
1	C0809000	Les paramètres du sous-bloc sont protégés en écriture (par exemple, paramètres spécifiés par configuration ou mode PNIO activé).
1	C0809100	Réservé
1	C0809400	Un paramètre dans le sous-bloc Configuration IP n'a pas été défini ou ne peut pas être utilisé.
1	C0809500	Il y a une incompatibilité entre un paramètre du sous-bloc Configuration IP et d'autres paramètres.
1	C080C200	L'instruction ne peut pas être exécutée. Cette erreur se produit, par exemple, si la communication avec l'interface a été perdue.
1	C080C300	Il n'y a pas suffisamment de ressources. Cette erreur peut se produire, par exemple, si l'instruction est appelée plusieurs fois avec des paramètres différents.
1	C080C400	Echec de la communication. Cette erreur peut se produire temporairement et nécessite une répétition du programme utilisateur.
1	C080D200	L'exécution de l'instruction n'est pas prise en charge par l'interface PROFINET.

### Bloc de données CONF\_DATA

Le schéma suivant montre comment les données de configuration à transférer sont stockées dans le DB de configuration.



Les données de configuration de CONF\_DB comprennent un bloc contenant un en-tête (IF\_CONF\_Header) et plusieurs sous-blocs. IF\_CONF\_Header fournit les éléments suivants :

- field\_type\_id (type de données UInt) : Zéro
- field\_id (type de données UInt) : Zéro
- subfield\_cnt (type de données UInt) : Nombre de sous-blocs

Chaque sous-bloc est constitué à son tour d'un en-tête (subfield\_type\_id, subfield\_length, subfield\_mode) et des paramètres spécifiques du sous-bloc. Chaque sous-bloc doit comporter un nombre pair d'octets. "subfield\_mode" accepte une valeur de 1.

### Remarque

Un seul bloc (IF\_CONF\_Header) est autorisé actuellement. Ses paramètres field\_type\_id et field\_id doivent avoir la valeur zéro. D'autres blocs ayant des valeurs différentes pour field\_type\_id et field\_id sont envisageables dans le futur.

Dans le bloc IF\_CONF\_Header, seuls deux sous-blocs, "addr" (adresse IP) et "nos" (Name of station), sont autorisés actuellement.

Tableau 10- 24 Sous-blocs pris en charge

subfield_type_id	Type de données	Explication
30	IF_CONF_V4	Paramètres IP : adresse IP, masque de sous-réseau, adresse du routeur
40	IF_CONF_NOS	Nom de station PROFINET IO (Name of station)

Tableau 10- 25 Eléments du type de données IF\_CONF\_V4

Nom	Type de données	Valeur initiale	Description	
Id	UInt	30	subfield_type_id	
len	UInt	18	subfield_length	
mode	UInt	1	subfield_mode (1: permanent)	
InterfaceAddress	IP_V4	-	Adresse d'interface	
ADDR	Array [1..4] of Byte			
ADDR[1]	Byte	b#16#C8		Adresse IP, octet de poids fort : 200
ADDR[2]	Byte	b#16#0C		Adresse IP, octet de poids fort : 12
ADDR[3]	Byte	b#16#01		Adresse IP, octet de poids faible : 1
ADDR[4]	Byte	b#16#90	Adresse IP, octet de poids faible : 144	
SubnetMask	IP_V4	-	Masque de sous-réseau	
ADDR	Array [1..4] of Byte			
ADDR[1]	Byte	b#16#FF		Masque de sous-réseau, octet de poids fort : 255
ADDR[2]	Byte	b#16#FF		Masque de sous-réseau, octet de poids fort : 255
ADDR[3]	Byte	b#16#FF		Masque de sous-réseau, octet de poids faible : 255
ADDR[4]	Byte	b#16#00	Masque de sous-réseau, octet de poids faible : 0	
DefaultRouter	IP_V4	-	Routeur par défaut	
ADDR	Array [1..4] of Byte			
ADDR[1]	Byte	b#16#C8	Routeur, octet de poids fort : 200	

Nom	Type de données	Valeur initiale	Description
ADDR[2]	Byte	b#16#0C	Routeur, octet de poids fort : 12
ADDR[3]	Byte	b#16#01	Routeur, octet de poids faible : 1
ADDR[4]	Byte	b#16#01	Routeur, octet de poids faible : 1

Tableau 10- 26 Eléments du type de données IF\_CONF\_NOS

Nom	Type de données	Valeur initiale	Description
id	UInt	40	subfield_type_id
len	UInt	246	subfield_length
mode	UInt	1	subfield_mode (1: permanent)
Nos (Name of station)	Array[1..240] of Byte	0	Nom de station : Vous devez occuper le tableau ARRAY à partir du premier octet. Si le tableau ARRAY est plus long que le nom de station à affecter, vous devez entrer un octet nul après le nom de station réel (conformément à CEI 61158-6-10). Sinon, le paramètre "nos" est rejeté et l'instruction T_CONFIG (Page 480) inscrit le code d'erreur DW#16#C0809400 dans STATUS. Si vous écrivez zéro dans le premier octet, le nom de la station est effacé.

Les noms de station sont soumis aux limitations suivantes :

- Un nom d'élément à l'intérieur du nom de station, c'est-à-dire une chaîne de caractères entre deux points, ne doit pas dépasser 63 caractères.
- Les caractères spéciaux, tels que tréma, crochet, trait de soulignement, barre oblique, espace, etc., sont interdits. Le seul caractère spécial permis est le trait d'union.
- Le nom de station ne doit pas commencer ni se terminer par un trait d'union (-).
- Le nom de station ne doit pas commencer par un chiffre.
- La forme n.n.n.n (n = 0, ... 999) du nom de station n'est pas permise.
- Le nom de station ne doit pas commencer par la chaîne "port-xyz" ou "port-xyz-abcde" (a, b, c, d, e, x, y, z = 0, ... 9).

#### Remarque

Vous pouvez également créer un tableau ARRAY "nos" qui est plus court que 240 octets, mais il ne doit pas faire moins de 2 octets. Dans ce cas, vous devez ajuster en conséquence la variable "len" (longueur du sous-bloc).

### Comment modifier les paramètres IP

Dans l'exemple suivant, l'adresse IP "InterfaceAddress", le masque de sous-réseau "SubnetMask" et le routeur IP "DefaultRouter" sont modifiés dans le sous-bloc "addr". Le bouton radio "Dériver l'adresse IP d'une autre source" dans la page "Adresses Ethernet" des "Propriétés" de la CPU doit être sélectionné pour que vous puissiez modifier l'adresse IP à l'aide de l'instruction "T\_CONFIG" après le chargement du programme.

Tableau 10- 27 Comment modifier les paramètres IP

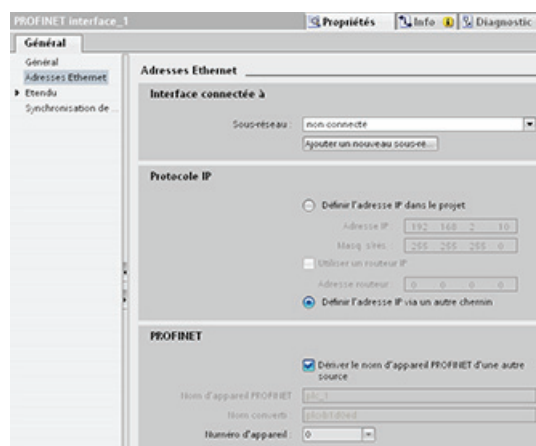
CONF_DATA_1				PROFINET interface_1	
	Nom	Type de données	Valeur de départ		
1	Static			Général	
2	Conf_data	Struct		Adresses Ethernet	
3	header	IF_CONF_Header		Etabli	
4	FieldType	UInt	0	Synchronisation de	
5	Fieldid	UInt	0		
6	SubfieldCount	UInt	1		
7	addr	IF_CONF_v4		Adresses Ethernet	
8	Id	UInt	30	Interface connectée à	
9	Length	UInt	18	Sous-réseau non connecté	
10	Mode	UInt	1	Ajouter un nouveau sous-...	
11	InterfaceAddress	IP_V4		Protocole IP	
12	ADDR	array [1..4] of Byte		<input type="radio"/> Définir l'adresse IP dans le projet Adresse IP 192.168.2.10 Masq. sous 255.255.255.0 <input type="checkbox"/> Utiliser un routeur IP Adresse routeur 0.0.0.0 <input checked="" type="radio"/> Définir l'adresse IP via un autre chemin	
13	ADDR[1]	Byte	192	PROFINET	
14	ADDR[2]	Byte	168	<input type="checkbox"/> Dériver le nom d'appareil PROFINET d'une autre source Nom d'appareil PROFINET plc_1 Nom converti plcB1-0ed Numéro d'appareil 0	
15	ADDR[3]	Byte	2		
16	ADDR[4]	Byte	30		
17	SubnetMask	IP_V4			
18	ADDR	array [1..4] of Byte			
19	ADDR[1]	Byte	255		
20	ADDR[2]	Byte	255		
21	ADDR[3]	Byte	255		
22	ADDR[4]	Byte	0		
23	DefaultRouter	IP_V4			
24	ADDR	array [1..4] of Byte			
25	ADDR[1]	Byte	192		
26	ADDR[2]	Byte	168		
27	ADDR[3]	Byte	2		
28	ADDR[4]	Byte	1		

## Comment modifier les paramètres IP et les noms des stations PROFINET IO

Dans l'exemple suivant, les sous-blocs "addr" et "nos" (Name of station) sont tous deux modifiés. L'option "Dériver le nom d'appareil PROFINET d'une autre source" dans la page "Adresses Ethernet" des "Propriétés" de la CPU doit être sélectionnée pour que vous puissiez modifier le nom d'appareil PROFINET à l'aide de l'instruction "T\_CONFIG" après le chargement du programme.

Tableau 10- 28 Comment modifier les paramètres IP et les noms des stations PROFINET IO

CONF_DATA_2			
	Nom	Type de données	Valeur de départ
1	Static		
2	Conf_data	Struct	
3	header	IF_CONF_Header	
4	FieldType	UInt	0
5	FieldId	UInt	0
6	SubfieldCount	UInt	2
7	addr	IF_CONF_v4	
8	Id	UInt	30
9	Length	UInt	18
10	Mode	UInt	1
11	InterfaceAddress	IP_V4	
12	ADDR	array [1..4] of Byte	
13	SubnetMask	IP_V4	
14	ADDR	array [1..4] of Byte	
15	DefaultRouter	IP_V4	
16	ADDR	array [1..4] of Byte	
17	nos	IF_CONF_IOS	
18	Id	UInt	40
19	Length	UInt	246
20	Mode	UInt	1
21	IOS	array [1..240] of Byte	



### 10.2.2.7 Paramètres communs des instructions

#### Paramètre d'entrée REQ

De nombreuses instructions de communication ouverte (Open User Communication) utilisent une entrée REQ pour déclencher l'opération en présence d'une transition du niveau bas au niveau haut. L'entrée REQ doit être au niveau haut (VRAI) pour une exécution de l'instruction, mais elle peut rester à VRAI aussi longtemps que désiré. L'instruction ne déclenche pas de nouvelle opération tant qu'elle n'a pas été exécutée avec l'entrée REQ à FAUX de manière à ce qu'elle réinitialise l'historique de l'entrée REQ. Cela est nécessaire pour que l'instruction puisse à nouveau détecter une transition du niveau bas au niveau haut afin de déclencher l'opération suivante.

Lorsque vous insérez l'une de ces instructions dans votre programme, STEP 7 vous demande d'identifier le DB d'instance. Utilisez un DB unique pour chaque appel d'instruction. Cela garantit que chaque instruction gère correctement les entrées telles que REQ.

### Paramètre entrée ID

Ce paramètre fait référence à "ID local (hexa)" dans la "Vue du réseau" de "Appareils & Réseaux" dans STEP 7 ; il s'agit de l'ID du réseau que vous voulez utiliser pour ce bloc de communication. L'ID doit être identique au paramètre ID associé dans la description de la liaison locale.

### Paramètres de sortie DONE, NDR, ERROR et STATUS

Ces instructions fournissent des sorties décrivant l'état d'achèvement.

Tableau 10- 29 Paramètres de sortie des instructions pour la communication ouverte

Paramètre	Type de données	Valeur par défaut	Description
DONE	Bool	FAUX	Est mis à VRAI pour un cycle pour indiquer que la dernière demande s'est achevée sans erreur. FAUX sinon.
NDR	Bool	FAUX	Est mis à VRAI pour un cycle pour indiquer que la dernière action demandée s'est achevée sans erreur et que de nouvelles données ont été reçues. FAUX sinon.
BUSY	Bool	FAUX	Est mis à VRAI pour indiquer que : <ul style="list-style-type: none"> <li>• La tâche n'est pas encore achevée.</li> <li>• Il n'est pas possible de déclencher une nouvelle tâche.</li> </ul> Est mis à FAUX quant la tâche est achevée.
ERROR	Bool	FAUX	Est mis à VRAI pour un cycle pour indiquer que la dernière demande s'est achevée avec des erreurs, le code d'erreur correspondant étant contenu dans STATUS. FAUX sinon.
STATUS	Word	0	Etat du résultat : <ul style="list-style-type: none"> <li>• Si le bit DONE ou NDR est à 1, STATUS est mis à 0 ou reçoit un code informatif.</li> <li>• Si le bit ERROR est à 1, STATUS aura la valeur du code d'erreur.</li> <li>• Si aucun des bits ci-dessus n'est à 1, l'instruction renvoie des résultats d'état qui décrivent l'état en cours de la fonction.</li> </ul> STATUS conserve sa valeur pendant la durée de l'exécution de la fonction.

---

#### Remarque

Notez que DONE, NDR et ERROR sont à 1 pour un cycle uniquement.

---

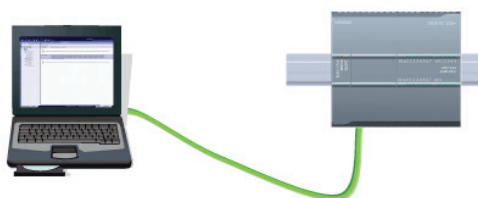


### TSAP et numéros de port limités pour la communication TCP et ISO passive

Si vous vous servez de l'instruction TCON pour configurer et établir la liaison de communication passive, n'utilisez pas les adresses de port suivantes qui sont réservées :

- TSAP ISO (passif) :
  - 01.00, 01.01, 02.00, 02.01, 03.00, 03.01
  - 10.00, 10.01, 11.00, 11.01, ... BF.00, BF.01
- Port TCP (passif) : 5001, 102, 123, 20, 21, 25, 34962, 34963, 34964, 80
- Port UDP (passif) : 161, 34962, 34963, 34964

### 10.2.3 Communication avec une console de programmation



Une CPU peut communiquer avec une console de programmation STEP 7 dans un réseau.

Tenez compte des points suivants lorsque vous établissez une communication entre une CPU et une console de programmation :

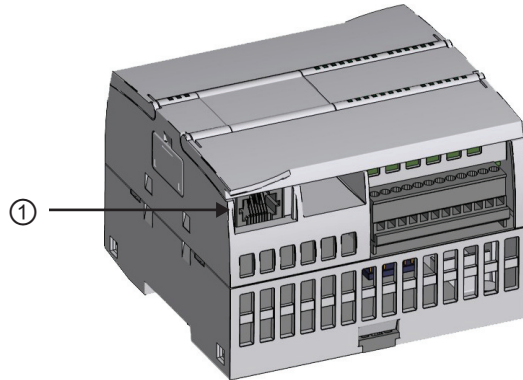
- Configuration/installation : Une configuration matérielle est requise.
- Aucun commutateur Ethernet n'est nécessaire pour la communication un à un ; un commutateur Ethernet est obligatoire pour plus de deux unités dans un réseau.

#### 10.2.3.1 Etablissement de la liaison de communication matérielle

Les interfaces PROFINET établissent les connexions physiques entre une console de programmation et une CPU. Comme la fonction Auto-Cross-Over est intégrée à la CPU, un câble Ethernet standard ou croisé peut être utilisé pour l'interface. Un commutateur Ethernet n'est pas nécessaire pour connecter une console de programmation directement à une CPU.

Procédez comme suit pour créer la liaison matérielle entre une console de programmation et une CPU :

1. Installez la CPU (Page 50).
2. Branchez le câble Ethernet dans le port PROFINET montré ci-dessous.
3. Connectez le câble Ethernet à la console de programmation.



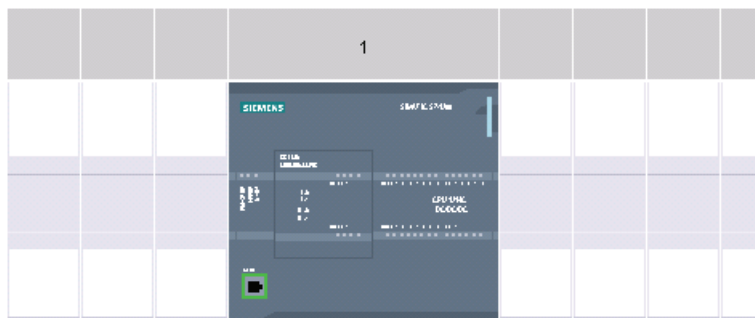
① Port PROFINET

Un serre-câble optionnel est disponible pour renforcer la connexion PROFINET.

### 10.2.3.2 Configuration des appareils

Si vous avez déjà créé un projet avec une CPU, ouvrez votre projet dans STEP 7.

Si ce n'est pas le cas, créez un projet et insérez une CPU (Page 128) dans le châssis. Dans le projet ci-dessous, vous voyez une CPU dans la "Vue des appareils".



### 10.2.3.3 Affectation d'adresses IP (Internet Protocol)

#### Affectation des adresses IP

Dans un réseau PROFINET, chaque appareil doit également comporter une adresse IP (protocole Internet). Cette adresse permet à l'appareil de transmettre des données dans un réseau routé plus complexe :

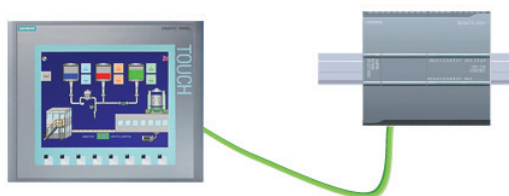
- Si vos consoles de programmation ou autres périphériques réseau utilisent une carte adaptateur intégrée reliée à votre réseau local d'usine ou une carte adaptateur Ethernet à USB reliée à un réseau isolé, vous devez leur affecter des adresses IP. Pour plus d'informations, reportez-vous à "Affectation d'adresses IP aux consoles de programmation et périphériques réseau" (Page 141).
- Vous pouvez également affecter une adresse IP à une CPU ou un périphérique réseau en ligne. Cela s'avère particulièrement utile lors d'une configuration d'appareil initiale. Pour plus d'informations, reportez-vous à "Affectation d'une adresse IP à une CPU en ligne (Page 141)".
- Une fois que vous avez configuré votre CPU ou périphérique réseau dans votre projet, vous pouvez configurer les paramètres pour l'interface PROFINET, dont son adresse IP. Reportez-vous à "Configuration d'une adresse IP pour une CPU dans votre projet" (Page 144) pour plus d'informations.

### 10.2.3.4 Test de votre réseau PROFINET

Une fois la configuration achevée, vous devez charger le projet dans la CPU. Toutes les adresses IP sont configurées lorsque vous chargez le projet dans la CPU.

La boîte de dialogue "Chargement étendu" de la fonction "Charger dans l'appareil" de la CPU peut montrer tous les dispositifs réseau accessibles en précisant si des adresses IP uniques ont été affectées ou non à tous les appareils. Reportez-vous à "Test du réseau PROFINET" (Page 149) pour plus d'informations.

## 10.2.4 Communication IHM vers automate



La CPU prend en charge les liaisons de communication PROFINET vers des IHM (interfaces homme-machine). Il faut tenir compte des exigences suivantes lors de la configuration de la communication entre CPU et IHM :

Configuration/installation :

- Le port PROFINET de la CPU doit être configuré pour la connexion avec l'IHM.
- L'IHM doit être installée et configurée.

- Les informations de configuration IHM font partie du projet CPU et peuvent être configurées et chargées de l'intérieur du projet.
- Aucun commutateur Ethernet n'est nécessaire pour la communication un à un ; un commutateur Ethernet est obligatoire pour plus de deux unités dans un réseau.

---

**Remarque**

Vous pouvez utiliser le commutateur Ethernet à 4 ports CSM1277 monté sur châssis pour connecter vos CPU et appareils IHM. Le port PROFINET de la CPU ne contient pas de dispositif de commutation Ethernet.

---

Fonctions prises en charge :

- L'IHM peut lire/écrire des données dans la CPU.
- Des messages peuvent être lancés, sur la base d'informations récupérées dans la CPU.
- Diagnostic système

Tableau 10- 30 Etapes requises pour configurer la communication entre une IHM et une CPU

Etape	Tâche
1	<p>Etablissement de la liaison de communication matérielle</p> <p>Une interface PROFINET établit la connexion physique entre une IHM et une CPU. Comme la fonction Auto-Cross-Over est intégrée à la CPU, vous pouvez utiliser un câble Ethernet standard ou croisé pour l'interface. Un commutateur Ethernet n'est pas nécessaire pour connecter une IHM et une CPU.</p> <p>Reportez-vous à "Communication avec une console de programmation, Etablissement de la liaison de communication matérielle" (Page 489) pour plus d'informations.</p>
2	<p>Configuration des appareils</p> <p>Reportez-vous à "Communication avec une console de programmation, Configuration des appareils" (Page 490) pour plus d'informations.</p>
3	<p>Configuration des liaisons réseau logiques entre une IHM et une CPU</p> <p>Reportez-vous à "Communication IHM vers API, Configuration des liaisons réseau logiques entre deux appareils" (Page 493) pour plus d'informations.</p>
4	<p>Configuration d'une adresse IP dans votre projet</p> <p>Utilisez le même processus de configuration ; vous devez toutefois configurer des adresses IP pour l'IHM et la CPU.</p> <p>Reportez-vous à "Configuration des appareils, Configuration d'une adresse IP pour une CPU dans votre projet" (Page 145) pour plus d'informations.</p>
5	<p>Test du réseau PROFINET</p> <p>Vous devez charger la configuration pour chaque CPU et appareil IHM.</p> <p>Reportez-vous à "Configuration des appareils, Test du réseau PROFINET" (Page 149) pour plus d'informations.</p>

### 10.2.4.1 Configuration des liaisons réseau logiques entre deux appareils

Une fois que vous avez configuré le châssis avec la CPU, vous être prêt à configurer vos liaisons réseau.

Dans le portail "Appareils & Réseaux", utilisez la "Vue du réseau" pour créer les liaisons réseau entre les appareils dans votre projet. Cliquez d'abord sur l'onglet "Liaisons", puis sélectionnez le type de liaison dans la liste déroulante juste à droite (une liaison ISO sur TCP, par exemple).

Pour créer une liaison PROFINET, cliquez sur le carré vert (PROFINET) du premier appareil et tracez une ligne vers le carré PROFINET du deuxième appareil. Relâchez le bouton de la souris : la liaison PROFINET est créée.

Reportez-vous à "Configuration des appareils, Création d'une liaison réseau" (Page 135) pour plus d'informations.

### 10.2.5 Communication API-API



Une CPU peut communiquer avec une autre CPU dans un réseau à l'aide des instructions TSEND\_C et TRCV\_C.

Tenez compte des points suivants lors de la configuration de la communication entre deux CPU :

- Configuration/installation : Une configuration matérielle est requise.
- Fonctions prises en charge : Lecture/écriture de données vers une CPU homologue
- Aucun commutateur Ethernet n'est nécessaire pour la communication un à un ; un commutateur Ethernet est obligatoire pour plus de deux unités dans un réseau.

Tableau 10- 31 Etapes requises pour configurer la communication entre deux CPU

Etape	Tâche
1	<p>Etablissement de la liaison de communication matérielle</p> <p>Une interface PROFINET établit la connexion physique entre deux CPU. Comme la fonction Auto-Cross-Over est intégrée à la CPU, vous pouvez utiliser un câble Ethernet standard ou croisé pour l'interface. Un commutateur Ethernet n'est pas nécessaire pour connecter les deux CPU.</p> <p>Reportez-vous à "Communication avec une console de programmation, Etablissement de la liaison de communication matérielle" (Page 489) pour plus d'informations.</p>
2	<p>Configuration des appareils</p> <p>Vous devez configurer deux CPU dans votre projet.</p> <p>Reportez-vous à "Communication avec une console de programmation, Configuration des appareils" (Page 490) pour plus d'informations.</p>

Étape	Tâche
3	Configuration des liaisons réseau logiques entre deux CPU Reportez-vous à "Communication API-API, Configuration des liaisons réseau logiques entre deux appareils" (Page 494) pour plus d'informations.
4	Configuration d'une adresse IP dans votre projet Utilisez le même processus de configuration ; vous devez toutefois configurer des adresses IP pour les deux CPU (par exemple, API_1 et API_2). Reportez-vous à "Configuration des appareils, Configuration d'une adresse IP pour une CPU dans votre projet" (Page 145) pour plus d'informations.
5	Configuration des paramètres d'émission et de réception Vous devez configurer des instructions TSEND_C et TRCV_C dans les deux CPU pour activer la communication entre elles. Reportez-vous à "Configuration de la communication entre deux CPU, Configuration des paramètres d'émission et de réception" (Page 495) pour plus d'informations.
6	Test du réseau PROFINET Vous devez charger la configuration pour chaque CPU. Reportez-vous à "Configuration des appareils, Test du réseau PROFINET" (Page 149) pour plus d'informations.

### 10.2.5.1 Configuration des liaisons réseau logiques entre deux appareils

Une fois que vous avez configuré le châssis avec la CPU, vous être prêt à configurer vos liaisons réseau.

Dans le portail "Appareils & Réseaux", utilisez la "Vue du réseau" pour créer les liaisons réseau entre les appareils dans votre projet. Cliquez d'abord sur l'onglet "Liaisons", puis sélectionnez le type de liaison dans la liste déroulante juste à droite (une liaison ISO sur TCP, par exemple).

Pour créer une liaison PROFINET, cliquez sur le carré vert (PROFINET) du premier appareil et tracez une ligne vers le carré PROFINET du deuxième appareil. Relâchez le bouton de la souris : la liaison PROFINET est créée.

Reportez-vous à "Configuration des appareils, Création d'une liaison réseau" (Page 135) pour plus d'informations.

### 10.2.5.2 Configuration du routage local/partenaire entre deux appareils

#### Configuration des paramètres généraux

Vous indiquez les paramètres de communication dans la boîte de dialogue de configuration "Propriétés" de l'instruction de communication. Cette boîte de dialogue apparaît au bas de la page lorsque vous sélectionnez n'importe quelle partie de l'instruction.

Reportez-vous à "Configuration des appareils, Configuration du routage local/partenaire (Page 136)" pour plus d'informations.

Vous définissez les TSAP ou ports à utiliser dans la section "Détails de l'adresse" de la boîte de dialogue "Paramètres de liaison". Vous indiquez le point TSAP ou le port d'une liaison dans la CPU dans la zone "TSAP local". Vous indiquez le point TSAP ou le port affecté à la liaison dans votre CPU partenaire dans la zone "TSAP partenaire".

### 10.2.5.3 Configuration des paramètres d'émission et de réception

On utilise des blocs de communication (par exemple, TSEND\_C et TRCV\_C) pour établir des liaisons entre deux CPU. Avant que la CPU ne puisse entamer une communication PROFINET, vous devez configurer les paramètres pour l'émission et le réception de messages. Ces paramètres conditionnent la façon dont la communication fonctionne lorsque des messages sont envoyés à un appareil cible ou reçu d'un appareil cible.

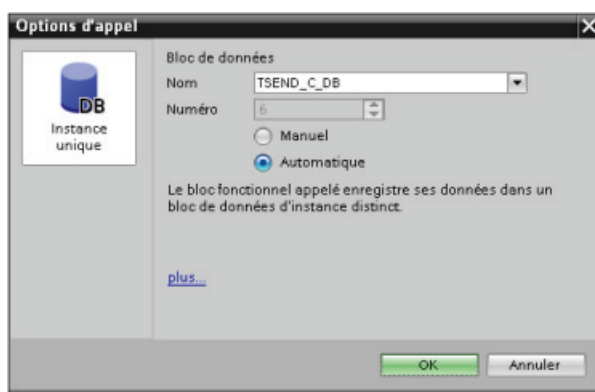
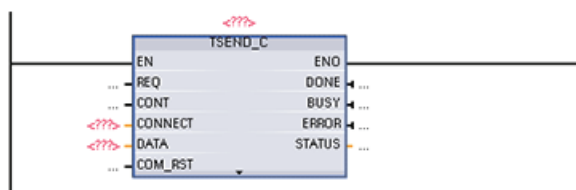
### Configuration des paramètres d'émission de l'instruction TSEND\_C

#### TSEND\_C, instruction

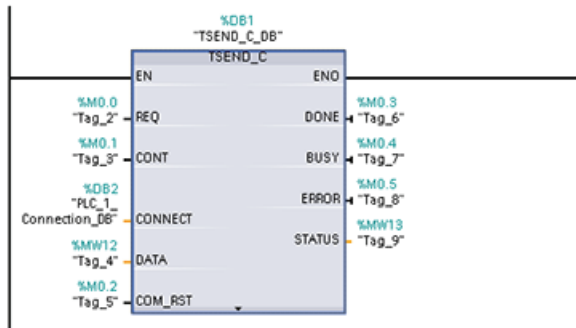
L'instruction TSEND\_C (Page 460) crée une liaison de communication vers une station partenaire. La liaison est configurée, établie et automatiquement surveillée jusqu'à ce que sa coupure soit demandée par l'instruction. L'instruction TSEND\_C combine les fonctions des instructions TCON, TDISCON et TSEND.

Vous pouvez configurer comment une instruction TSEND\_C envoie des données dans la Configuration des appareils de STEP 7. Pour commencer, vous insérez l'instruction dans le programme à partir du dossier "Communication" dans la task card "Instructions".

L'instruction TSEND\_C s'affiche avec la boîte de dialogue d'options d'appel dans laquelle vous indiquez un DB pour le stockage des paramètres de l'instruction.



Vous pouvez affecter des adresses de variables aux entrées et sorties comme illustré dans la figure ci-après :



### Configuration des paramètres généraux

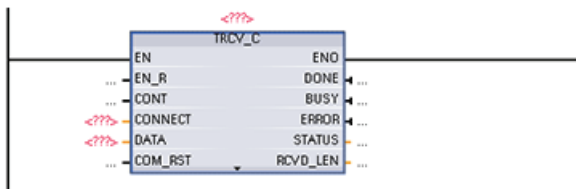
Vous indiquez les paramètres dans la boîte de dialogue de configuration Propriétés de l'instruction TSEND\_C. Cette boîte de dialogue apparaît au bas de la page lorsque vous sélectionnez n'importe quelle partie de l'instruction TSEND\_C.

### Configuration des paramètres de réception de l'instruction TRCV\_C

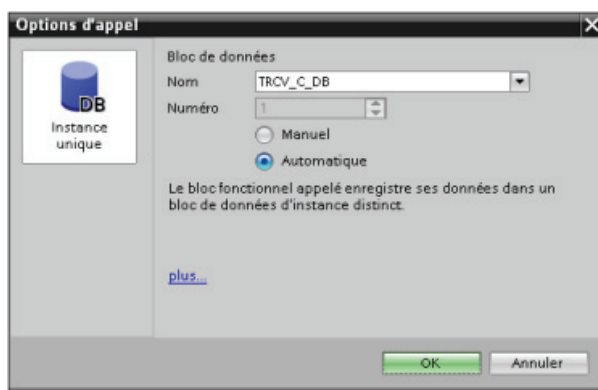
#### Instruction TRCV\_C

L'instruction TRCV\_C (Page 460) crée une liaison de communication vers une station partenaire. La liaison est configurée, établie et automatiquement surveillée jusqu'à ce que sa coupure soit demandée par l'instruction. L'instruction TRCV\_C combine les fonctions des instructions TCON, TDISCON et TRCV.

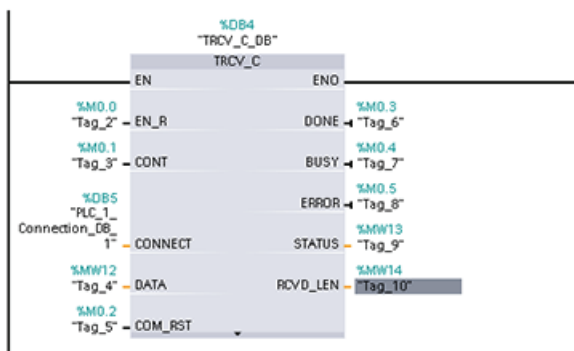
Vous pouvez configurer comment une instruction TRCV\_C reçoit des données dans la configuration CPU de STEP 7. Pour commencer, insérez l'instruction dans le programme à partir du dossier "Communication" dans la task card "Instructions". L'instruction TRCV\_C s'affiche avec la boîte de dialogue d'options d'appel dans laquelle vous indiquez un DB pour le stockage des paramètres de l'instruction.







Vous pouvez affecter des adresses de variables aux entrées et sorties comme illustré dans la figure ci-après :



### Configuration des paramètres généraux

Vous indiquez les paramètres dans la boîte de dialogue de configuration Propriétés de l'instruction TRCV\_C. Cette boîte de dialogue apparaît au bas de la page lorsque vous sélectionnez n'importe quelle partie de l'instruction TRCV\_C.

## 10.2.6 Configuration d'une CPU et d'un périphérique PROFINET IO

### Ajout d'un périphérique PROFINET IO

Dans le portail "Appareils & Réseaux", utilisez le catalogue du matériel pour ajouter des périphériques PROFINET IO.

---



#### Remarque

Vous pouvez utiliser STEP 7 Professional ou Basic, V11 ou une version supérieure, pour ajouter un périphérique PROFINET IO.

---

Développez p. ex. les dossiers suivants dans le catalogue du matériel pour ajouter un périphérique I/O ET200S : Périphérie décentralisée, ET200S, Coupleurs, et PROFINET. Vous pouvez alors sélectionner le coupleur dans la liste des périphériques ET200S (classés par numéro de référence) et ajouter le périphérique IO ET200S.

Tableau 10- 32 Ajout d'un périphérique IO ET200S à la configuration des appareils

Insérez le périphérique IO	Résultat
	

Vous pouvez maintenant connecter le périphérique PROFINET IO à la CPU :

1. Cliquez avec le bouton droit de la souris sur le lien "Non affecté" sur l'appareil et sélectionnez "Affecter un nouveau contrôleur IO" dans le menu contextuel pour afficher la boîte de dialogue "Sélectionner un contrôleur IO".
2. Sélectionnez votre CPU S7-1200 ("PLC\_1" dans cet exemple) dans la liste des contrôleurs IO du projet.
3. Cliquez sur "OK" pour créer la connexion réseau.

### Configuration de liaisons réseau logiques

Une fois que vous avez configuré le châssis avec la CPU, vous être prêt à configurer vos liaisons réseau.

Dans le portail "Appareils & Réseaux", utilisez la "Vue du réseau" pour créer les liaisons réseau entre les appareils dans votre projet. Pour créer une liaison PROFINET, cliquez sur le carré vert (PROFINET) du premier appareil et tracez une ligne vers le carré PROFINET du deuxième appareil. Relâchez le bouton de la souris : la liaison PROFINET est créée.

Reportez-vous à "Configuration des appareils, Création d'une liaison réseau" (Page 135) pour plus d'informations.

### Affectation de CPU et de noms d'appareils

Les liaisons réseau entre les appareils affectent également le périphérique PROFINET IO à la CPU, ce qui est nécessaire pour que cette CPU puisse piloter le périphérique. Pour modifier cette affectation, cliquez sur le nom d'API figurant sur le périphérique PROFINET IO. Une boîte de dialogue s'ouvre, permettant de déconnecter le périphérique PROFINET IO de la CPU en cours et de le réaffecter ou de le laisser sans affectation, au choix.

Vous devez affecter un nom aux appareils sur votre réseau PROFINET pour pouvoir vous connecter à la CPU. Servez-vous de la "Vue du réseau" pour affecter un nom à vos appareils PROFINET s'ils n'en ont pas déjà un ou s'il faut le modifier. Pour ce faire, cliquez avec le bouton droit de la souris sur le périphérique PROFINET IO et sélectionnez "Affecter un nom d'appareil".

Vous devez définir, pour chaque périphérique PROFINET IO, le même nom dans le projet STEP 7 et, à l'aide de l'outil "En ligne & diagnostic", dans la mémoire de configuration de périphérique PROFINET IO (par exemple, mémoire de configuration de coupleur ET200S). Si un nom manque ou ne coïncide pas à l'un ou l'autre endroit, le mode d'échange de données PROFINET IO ne fonctionnera pas. Reportez-vous à "Outils en ligne et de diagnostic, Affectation d'un nom à un appareil PROFINET en ligne (Page 724)" pour plus d'informations.

## Affectation des adresses IP

Dans un réseau PROFINET, chaque appareil doit également comporter une adresse IP (protocole Internet). Cette adresse permet à l'appareil de transmettre des données dans un réseau routé plus complexe :

- Si vos consoles de programmation ou autres périphériques réseau utilisent une carte adaptateur intégrée reliée à votre réseau local d'usine ou une carte adaptateur Ethernet à USB reliée à un réseau isolé, vous devez leur affecter des adresses IP. Pour plus d'informations, reportez-vous à "Affectation d'adresses IP aux consoles de programmation et périphériques réseau" (Page 141).
- Vous pouvez également affecter une adresse IP à une CPU ou un périphérique réseau en ligne. Cela s'avère particulièrement utile lors d'une configuration d'appareil initiale. Pour plus d'informations, reportez-vous à "Affectation d'une adresse IP à une CPU en ligne (Page 144)".
- Une fois que vous avez configuré votre CPU ou périphérique réseau dans votre projet, vous pouvez configurer les paramètres pour l'interface PROFINET, dont son adresse IP. Reportez-vous à "Configuration d'une adresse IP pour une CPU dans votre projet" (Page 145) pour plus d'informations.

## Configuration du temps de cycle IO

Un périphérique PROFINET IO reçoit de nouvelles données de la CPU périodiquement selon un "cycle IO". Le temps d'actualisation peut être configuré séparément pour chaque périphérique et détermine l'intervalle de temps auquel des données sont transmises entre la CPU et le périphérique.

STEP 7 calcule automatiquement le temps d'actualisation "cycle IO" dans le paramétrage par défaut de chaque périphérique du réseau PROFINET, en tenant compte du volume de données à échanger et du nombre de périphériques affectés à ce contrôleur. Si vous ne voulez pas que le temps d'actualisation soit calculé automatiquement, vous pouvez modifier ce paramètre.

Vous indiquez les paramètres de "cycle IO" dans la boîte de dialogue de configuration "Propriétés" du périphérique PROFINET IO. Cette boîte de dialogue apparaît au bas de la page lorsque vous sélectionnez n'importe quelle partie de l'instruction.

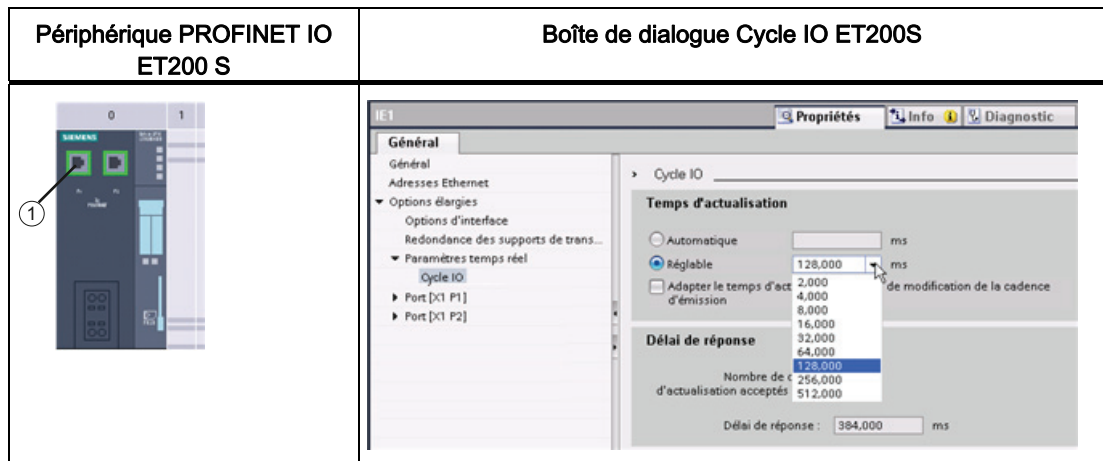
Cliquez sur le port PROFINET dans la "Vue des appareils" du périphérique PROFINET IO. Dans la boîte de dialogue "Interface PROFINET", accédez aux paramètres "Cycle IO" au moyen des sélections de menu suivantes :

- "Options élargies"
- "Paramètres temps réel"
- "Cycle IO"

Définissez le "Temps d'actualisation" du cycle IO avec les sélections suivantes :

- Sélectionnez "Calculé automatiquement" pour avoir un temps d'actualisation approprié calculé automatiquement.
- Pour définir le temps d'actualisation vous-même, sélectionnez "Réglable" et entrez le temps d'actualisation requis en ms.
- Pour assurer la compatibilité entre la cadence d'émission et le temps d'actualisation, activez l'option "Adapter le temps d'actualisation en cas de modification de la cadence d'émission". Cette option garantit que le temps d'actualisation ne sera pas inférieur à la cadence d'émission.

Tableau 10- 33 Configuration du temps de cycle PROFINET IO ET200S



① Port PROFINET

## 10.2.7 Diagnostic

### Bloc d'organisation d'alarme de diagnostic (OB82)

Si un module apte au diagnostic à alarme de diagnostic activée détecte une modification dans son état de diagnostic, il envoie une demande d'alarme de diagnostic à la CPU dans les situations suivantes :

- Un problème a été détecté par ce module (rupture de fil, par exemple), un composant requiert une maintenance ou les deux (événement apparaissant).
- Le problème a été corrigé ou n'existe plus et aucun autre composant ne requiert de maintenance (événement disparaissant).

Si l'OB82 n'existe pas, ces erreurs sont écrites dans la mémoire tampon de diagnostic. La CPU n'exécute aucune action ni ne passe à l'état ARRET.

Si l'OB82 existe, le système d'exploitation peut appeler l'OB82 en réponse à un événement apparaissant. Vous devez créer l'OB82 et cet OB vous permet de configurer le traitement d'erreur local et une réaction plus détaillée aux événements apparaissants.

Si vous utilisez une CPU à fonctionnalité DPV1, vous pouvez obtenir des informations supplémentaires sur l'alarme à l'aide de l'instruction RALRM qui fournit des informations plus spécifiques que les informations de déclenchement de l'OB82.

### Alarmes d'accès à la périphérie

Ces erreurs sont écrites dans la mémoire tampon de diagnostic. La CPU n'exécute aucune action ni ne passe à l'état ARRET. Les erreurs inscrites dans la mémoire tampon de diagnostic comprennent :

- Défaillances de module
- Incompatibilité de module
- Module manquant

### Erreurs d'accès aux E/S

Ces erreurs sont écrites dans la mémoire tampon de diagnostic. La CPU n'exécute aucune action ni ne passe à l'état ARRET.

## 10.2.8 Instructions de périphérie décentralisée

Reportez-vous à "Périphérie décentralisée (PROFINET, PROFIBUS ou AS-i)" (Page 291) pour obtenir des informations sur la manière d'utiliser les instructions de périphérie décentralisée avec ces réseaux de communication.

### 10.2.9 Instructions de diagnostic

Reportez-vous à "Diagnostic (PROFINET ou PROFIBUS)", "Instructions de diagnostic" (Page 317) pour obtenir des informations sur la manière d'utiliser ces instructions avec ces réseaux de communication.

### 10.2.10 Evénements de diagnostic pour la périphérie décentralisée

Reportez-vous à "Diagnostic (PROFINET ou PROFIBUS)", "Evénements de diagnostic pour la périphérie décentralisée" (Page 317) pour obtenir des informations sur la manière d'utiliser ces informations de diagnostic avec ces réseaux de communication.

## 10.3 PROFIBUS

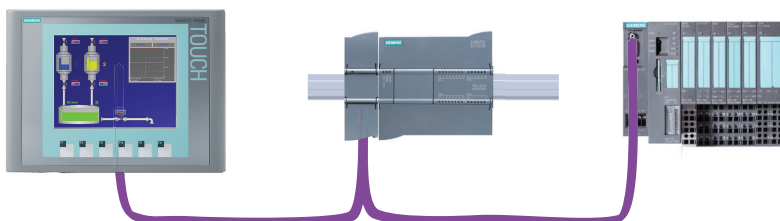
Un système PROFIBUS utilise un maître du bus pour interroger des esclaves répartis en mode multipoint sur un bus série RS485. Un esclave PROFIBUS est n'importe quel périphérique (transducteur d'E/S, vanne, entraînement de moteur ou autre appareil de mesure) qui traite des informations et envoie ses sorties au maître. L'esclave constitue une station passive sur le réseau puisqu'il n'a pas de droits d'accès au bus et peut seulement accuser réception des messages reçus ou envoyer des messages sur demande du maître. Tous les esclaves PROFIBUS ont la même priorité et le maître est à l'origine de toutes les communications de réseau.

Un maître PROFIBUS constitue une station active sur le réseau. PROFIBUS DP définit deux classes de maîtres. Un maître de classe 1 (normalement un automate programmable industriel API ou un PC exécutant un logiciel spécial) gère la communication normale ou l'échange de données avec les esclaves qui lui sont affectés. Un maître de classe 2 (généralement un appareil de configuration, tel qu'un ordinateur portable ou une console de programmation servant à la mise en service, à la maintenance ou au diagnostic) est un appareil spécial utilisé avant tout pour la mise en service des esclaves et à des fins de diagnostic.

Le S7-1200 est raccordé à un réseau PROFIBUS en tant qu'esclave DP avec le module de communication CM 1242-5. Le module CM 1242-5 (esclave DP) peut être le partenaire de communication de maîtres DP V0/V1. Dans la figure ci-dessous, le S7-1200 est un esclave DP affecté à un automate S7-300.



Le S7-1200 est raccordé à un réseau PROFIBUS en tant que maître DP avec le module de communication CM 1243-5. Le module CM 1243-5 (maître DP) peut être le partenaire de communication d'esclaves DP V0/V1. Dans la figure ci-dessous, le S7-1200 est un maître pilotant un esclave DP ET200S.



Si on a installé à la fois un CM 1242-5 et un CM 1243-5, un S7-1200 peut opérer simultanément comme esclave d'un système maître DP de niveau supérieur et comme maître d'un système maître DP de niveau inférieur.



Dans V3.0, vous pouvez configurer un maximum de trois CM PROFIBUS par station, toute combinaison de CM maîtres ou esclaves DP étant possible. Dans une implémentation V3.0, les maîtres DP peuvent piloter chacun 32 esclaves au maximum.

Dans V2.2, vous pouvez configurer un maximum de trois CM PROFIBUS par station, dont seulement un peut être un maître DP. Dans une implémentation V2.2, un maître DP peut piloter 16 esclaves au maximum.

## 10.3.1 Modules de communication pour PROFIBUS

### 10.3.1.1 Connexion à PROFIBUS

#### Connexion du S7-1200 à PROFIBUS DP

Les modules de communication suivants permettent de connecter le S7-1200 à un système de bus de terrain PROFIBUS :

- CM 1242-5  
Fonction d'esclave DP
- CM 1243-5  
Fonction de maître DP de classe 1

Lorsqu'il est équipé d'un CM 1242-5 et d'un CM 1243-5, le S7-1200 peut assurer simultanément les fonctions suivantes :

- esclave d'un système maître DP de niveau supérieur
- et
- maître d'un système maître DP subordonné

### 10.3.1.2 Services de communication des CM PROFIBUS

#### Protocole de bus

Les CM PROFIBUS utilisent le protocole PROFIBUS DP-V1.

#### Partenaires de communication PROFIBUS du S7-1200

Les deux CM PROFIBUS-CM permettent au S7-1200 de transmettre des données aux partenaires de communication ci-après.

- CM 1242-5

Le CM 1242-5 (esclave DP) peut être partenaire de communication des maîtres DP-V0/V1 suivants :

- SIMATIC S7-1200, S7-300, S7-400, S7-Modular Embedded Controller
- Modules maître DP de la périphérie décentralisée SIMATIC ET200
- Stations PC SIMATIC
- SIMATIC NET IE/PB Link
- Automates de divers fabricants

- CM 1243-5

Le CM 1243-5 (maître DP) peut être partenaire de communication des esclaves DP-V0/V1 suivants :

- Périphérie décentralisée SIMATIC ET200
- CPU S7-1200 avec CM 1242-5
- CPU S7-200 avec module PROFIBUS DP EM 277
- variateurs SINAMICS
- Moteurs et actionneurs de divers fabricants
- Capteurs de divers fabricants
- CPU S7-300/400 avec interface PROFIBUS
- CPU S7-300/400 avec CP PROFIBUS (CP 342-5 p. ex.)
- Stations PC SIMATIC avec CP PROFIBUS



## Types de communication sous DP-V1

Les types de communication suivants sont disponibles sous DP-V1 :

- Communication cyclique (CM 1242-5 et CM 1243-5)

Les deux modules PROFIBUS prennent en charge la communication cyclique pour les échanges de données de process entre esclave DP et maître DP.

La communication cyclique est exécutée par le système d'exploitation de la CPU. L'emploi de blocs logiciels n'est pas nécessaire. Les données d'E/S sont lues et écrites directement dans la mémoire image de process de la CPU.

- Communication acyclique (uniquement CM 1243-5)

Le module maître DP prend également en charge la communication acyclique à l'aide de blocs logiciels :

- L'instruction "RALRM" est disponible pour le traitement des alarmes.
- Les instructions "RDREC" et "WRREC" servent à transmettre des données de configuration et de diagnostic.

Fonctions non prises en charge par le CM 1243-5 :

SYNC/FREEZE  
Get\_Master\_Diag

## Autres services de communication du CM 1243-5

Le modules maître DP CM 1243-5 prend en charge les services de communication additionnels suivants :

- Communication S7

- Services PUT/GET

Le maître DP fait office de client et de serveur pour les requêtes via PROFIBUS des autres automates S7 ou PC.

- Communication PG/OP

Les fonctions de PG permettent de charger des données de configuration et programmes utilisateur à partir de la PG et de transmettre des données de diagnostic à la PG.

Les partenaires potentiels de la communication OP sont les pupitres HMI, Panel PC SIMATIC avec WinCC flexible ou les systèmes SCADA qui prennent en charge la communication S7.

### 10.3.1.3 Autres propriétés des CM PROFIBUS

#### Configuration et échange de module

La configuration des modules, réseaux et liaisons s'effectue sous STEP 7 version V11.0 et suivantes.

Un fichier GSD, permettant de configurer le CM 1242-5 (esclave DP) dans un système d'une autre marque, est mis à disposition sur le CD fourni avec le module, de même que sur le site Internet Siemens Automation Customer Support.

Les données de configuration des CM PROFIBUS sont enregistrées sur la CPU locale. Ceci permet d'échanger facilement ces modules de communication en cas de panne.

Vous pouvez configurer au maximum trois CM PROFIBUS par station.

#### Connexions électriques

- Alimentation
  - Le CM 1242-5 est alimenté par le bus de fond de panier de la station SIMATIC.
  - Le CM 1243-5 est doté par ailleurs d'une connexion pour une alimentation DC 24 V.
- PROFIBUS

L'interface RS485 de la connexion PROFIBUS est une embase femelle SubD à 9 points.

Les réseaux PROFIBUS optiques peuvent être connectés via un Optical Bus Terminal OBT ou un Optical Link Module OLM.

#### Informations complémentaires

Les manuels des appareils contiennent des informations détaillées sur les CM PROFIBUS. Vous les trouverez sur Internet sur le site Siemens Industrial Automation Customer Support sous les numéros d'article suivants :

- CM 1242-5 :  
49852105 (<http://support.automation.siemens.com/WW/view/fr/49852105>)
- CM 1243-5 :  
49851842 (<http://support.automation.siemens.com/WW/view/fr/49851842>)

### 10.3.1.4 Exemples de configuration pour PROFIBUS

Vous trouverez ci-après des exemples de configuration mettant en œuvre le CM 1242-5 comme esclave PROFIBUS et le CM1243-5 comme maître PROFIBUS.

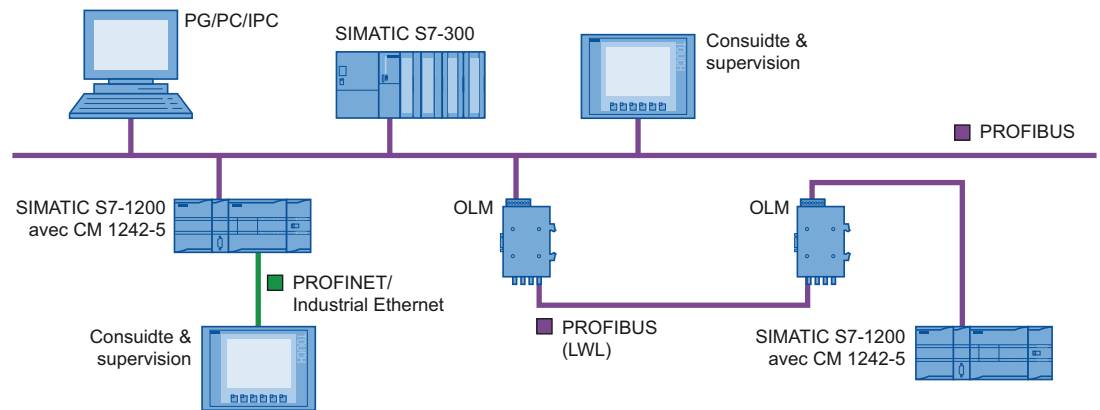


Figure 10-1 Exemple de configuration avec CM 1242-5 comme esclave PROFIBUS

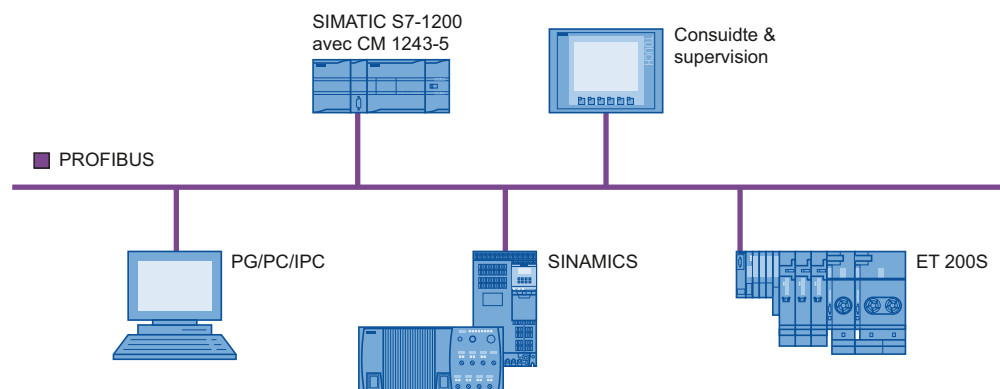


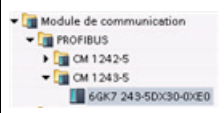
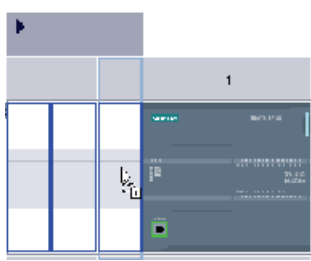
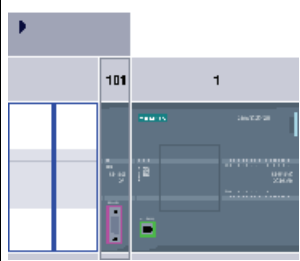
Figure 10-2 Exemple de configuration avec CM 1243-5 comme maître PROFIBUS

### 10.3.2 Configuration d'un maître et d'un esclave DP

#### 10.3.2.1 Ajout du module CM 1243-5 (maître DP) et d'un esclave DP

Dans le portail "Appareils & Réseaux", utilisez le catalogue du matériel pour ajouter des modules PROFIBUS à la CPU. Ces modules se raccordent sur le côté gauche de la CPU. Pour insérer un module dans la configuration matérielle, sélectionnez le module concerné dans le catalogue du matériel et double-cliquez ou faites glisser le module dans l'emplacement mis en évidence.

Tableau 10- 34 Ajout d'un module CM 1243-5 PROFIBUS (maître DP) à la configuration des appareils



Module	Sélectionnez le module	Insérez le module	Résultat
CM 1243-5 (maître DP)			

Utilisez également le catalogue du matériel pour ajouter des esclaves DP. Par exemple, pour ajouter un esclave DP ET200 S, affichez le détail des conteneurs suivants dans le Catalogue du matériel :

- Périphérie décentralisée
- ET200 S
- Coupleurs
- PROFIBUS

Puis, sélectionnez "6ES7 151-1BA02-0AB0" (IM151-1 HF) dans la liste des numéros de référence et ajoutez l'esclave DP ET200 S comme illustré dans la figure ci-dessous.

Tableau 10- 35 Ajout d'un esclave DP ET200 S à la configuration des appareils

Insérez l'esclave DP	Résultat
	

### 10.3.2.2 Configuration des liaisons réseau logiques entre deux appareils PROFIBUS

Maintenant que vous avez configuré le module CM 1243-5 (maître DP), vous êtes prêt à configurer vos liaisons réseau.

Dans le portail "Appareils & Réseaux", utilisez la "Vue du réseau" pour créer les liaisons réseau entre les appareils dans votre projet. Pour créer la liaison PROFIBUS, sélectionnez le carré violet (PROFIBUS) sur le premier appareil. Tracez une ligne vers le carré PROFIBUS sur le deuxième appareil. Relâchez le bouton de la souris : la liaison PROFIBUS est créée.

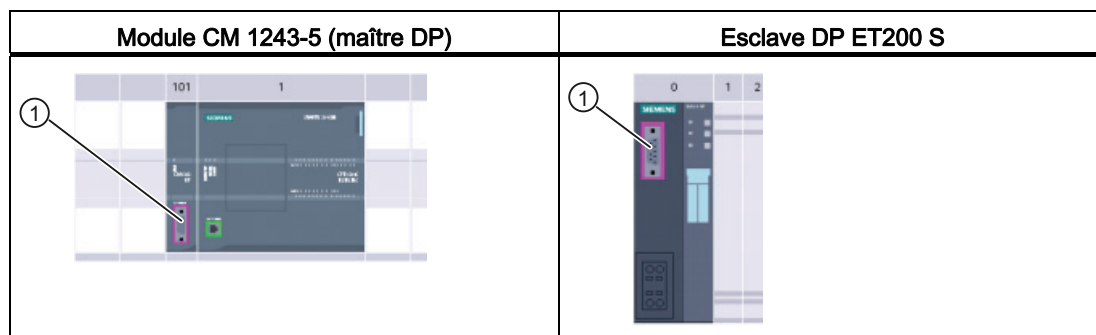
Reportez-vous à "Configuration des appareils, Création d'une liaison réseau" (Page 135) pour plus d'informations.

### 10.3.2.3 Affectation d'adresses PROFIBUS au module CM 1243-5 et à l'esclave DP

#### Configuration de l'interface PROFIBUS

Une fois que vous avez configuré les liaisons réseau logiques entre deux appareils PROFIBUS, vous pouvez configurer les paramètres des interfaces PROFIBUS. Pour ce faire, cliquez sur le carré PROFIBUS violet sur le module CM 1243-5. L'onglet "Propriétés" de la fenêtre d'inspection affiche l'interface PROFIBUS. L'interface PROFIBUS de l'esclave DP se configure de la même manière.

Tableau 10- 36 Configuration des interfaces PROFIBUS du module CM 1243-5 (maître DP) et de l'esclave DP ET200 S



① port PROFIBUS

#### Affectation de l'adresse PROFIBUS

Dans un réseau PROFIBUS, une adresse PROFIBUS est affectée à chaque appareil. Cette adresse est comprise entre 0 et 127, aux exceptions suivantes près :

- Adresse 0 : Réservee aux outils de configuration et/ou de programmation du réseau raccordés au bus
- Adresse 1 : Réservee au premier maître par Siemens

- Adresse 126 : Réservée aux appareils sortant de l'usine qui n'ont pas de réglage par commutateur et doivent être réadressés via le réseau
- Adresse 127 : Réservée aux messages à diffusion générale vers tous les appareils dans le réseau et ne pouvant pas être affectée à des appareils opérationnels.

Ainsi, les adresses pouvant être utilisées par des appareils PROFIBUS opérationnels vont de 2 à 125.

Dans la fenêtre Propriétés, sélectionnez l'entrée de configuration "Adresse PROFIBUS". STEP 7 affiche la boîte de dialogue de configuration de l'adresse PROFIBUS qui sert à définir l'adresse PROFIBUS de l'appareil.

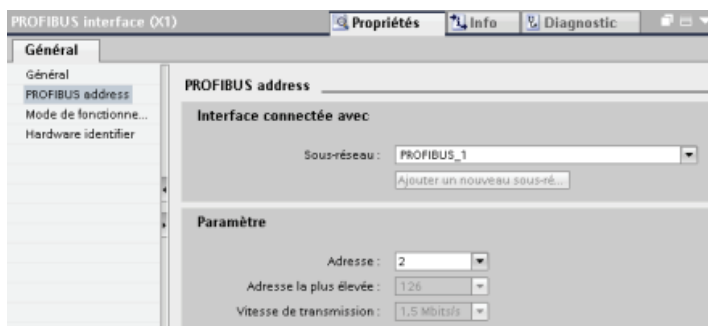


Tableau 10- 37 Paramètres pour l'adresse PROFIBUS

Paramètre	Description	
Sous-réseau	Nom du sous-réseau auquel l'appareil est connecté. Cliquez sur le bouton "Ajouter nouveau sous-réseau" pour créer un nouveau sous-réseau. "non connecté" est la valeur par défaut. Deux types de connexion sont possibles : <ul style="list-style-type: none"> <li>• La valeur par défaut "non connecté" fournit une liaison locale.</li> <li>• Un sous-réseau est nécessaire lorsque votre réseau contient au moins deux appareils.</li> </ul>	
Paramètres	Adresse	Adresse PROFIBUS affectée à l'appareil
	Adresse la plus élevée	L'adresse PROFIBUS la plus élevée est basée sur les stations actives dans le réseau PROFIBUS (par exemple, maître DP). Les esclaves DP passifs ont, indépendamment de cela, des adresses PROFIBUS allant de 1 à 125 même si, par exemple, l'adresse PROFIBUS la plus élevée est définie à 15. L'adresse PROFIBUS la plus élevée est pertinente pour le passage du jeton (c'est-à-dire des droits d'émission), le jeton étant uniquement transmis à des stations actives. Indiquer l'adresse PROFIBUS la plus élevée permet d'optimiser le bus.
	Vitesse de transmission	Vitesse de transmission du réseau PROFIBUS configuré : Les vitesses de transmission PROFIBUS vont de 9,6 Kbits/s à 12 Mbits/s. Le réglage de la vitesse de transmission dépend des propriétés des noeuds PROFIBUS utilisés. La vitesse de transmission ne doit pas dépasser la vitesse prise en charge par le noeud le plus lent.  La vitesse de transmission est normalement définie pour le maître sur le réseau PROFIBUS, tous les esclaves DP utilisant automatiquement la même vitesse de transmission (détection automatique de débit).

### 10.3.3 Instructions de périphérie décentralisée

Reportez-vous à "Périphérie décentralisée (PROFINET, PROFIBUS ou AS-i)" (Page 291) pour obtenir des informations sur la manière d'utiliser les instructions de périphérie décentralisée avec ces réseaux de communication.

### 10.3.4 Instructions de diagnostic

Reportez-vous à "Diagnostic (PROFINET ou PROFIBUS)", "Instructions de diagnostic" (Page 317) pour obtenir des informations sur la manière d'utiliser ces instructions avec ces réseaux de communication.

### 10.3.5 Événements de diagnostic pour périphérie décentralisée

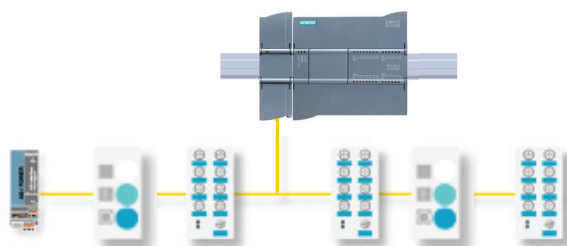
Reportez-vous à "Diagnostic (PROFINET ou PROFIBUS)", "Événements de diagnostic pour la périphérie décentralisée" (Page 317) pour obtenir des informations sur la manière d'utiliser ces informations de diagnostic avec ces réseaux de communication.

## 10.4 Interface AS-i

Le module CM 1243-2 maître AS-i S7-1200 permet le raccordement d'un réseau AS-i à une CPU S7-1200.

L'interface actionneur/capteur, interface AS-i, est un réseau d'interconnexion à un seul maître pour le niveau le plus bas dans les systèmes d'automatisation. Le module CM 1243-2 sert de maître AS-i pour le réseau. Avec un câble AS-i unique, il est possible de connecter les capteurs et actionneurs (esclaves AS-i) à la CPU par le biais du CM 1243-2. Le CM 1243-2 gère toute la coordination du réseau AS-i et transmet les données et informations d'état des actionneurs et capteurs à la CPU par le biais des adresses d'E/S qui lui sont affectées. Vous pouvez accéder aux valeurs binaires ou analogiques en fonction du type d'esclave. Les esclaves AS-i sont les voies d'entrée et de sortie du réseau AS-i et ne sont actifs que si le CM 1243-2 les appelle.

Dans la figure ci-dessous, le S7-1200 est un maître AS-i qui pilote un pupitre opérateur AS-i et des modules esclaves TOR/analogiques.



### 10.4.1 Configuration d'un maître et d'un esclave AS-i

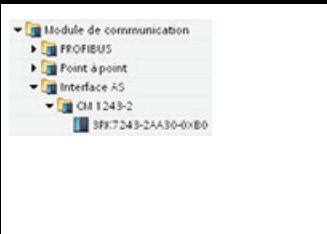


Le maître AS-i CM 1243-2 est intégré dans le système d'automatisation S7-1200 en tant que module de communication.

Vous trouverez des informations détaillées sur le module maître AS-i CM 1243-2 dans le manuel "Maître AS-i CM 1243-2 et module AS-i de découplage des données DCM 1271 pour SIMATIC S7-1200" (<http://support.automation.siemens.com/WW/view/en/50414115/133300>).

#### 10.4.1.1 Ajout du module maître AS-i CM 1243-2 et d'un esclave AS-i

Utilisez le catalogue du matériel pour ajouter des modules maître AS-i CM 1243-2 à la CPU. Ces modules se raccordent sur le côté gauche de la CPU, trois modules maître AS-i CM 1243-2 pouvant être utilisés au maximum. Pour insérer un module dans la configuration matérielle, sélectionnez le module concerné dans le catalogue du matériel et double-cliquez ou faites glisser le module dans l'emplacement mis en évidence.

Tableau 10- 38 Ajout d'un module maître AS-i CM 1243-2 à la configuration des appareils

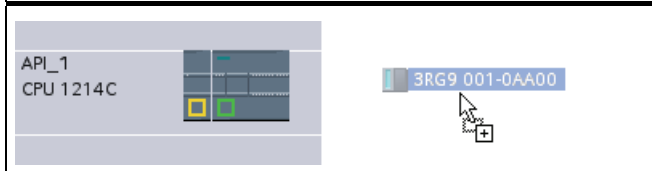
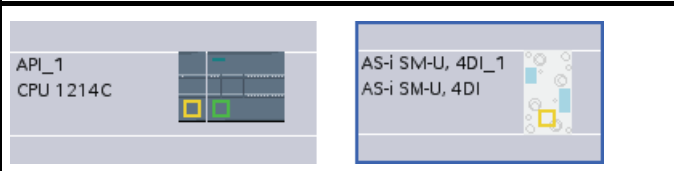
Module	Sélectionnez le module	Insérez le module	Résultat
CM 1243-2 maître AS-i			

Utilisez également le catalogue du matériel pour ajouter des esclaves AS-i. Par exemple, pour ajouter un esclave qui soit "module d'E/S, compact, TOR, entrées", affichez le détail des dossiers suivants dans le Catalogue du matériel :

- Appareils de terrain
- Esclaves AS Interface

Puis, sélectionnez "3RG9 001-0AA00" (AS-i SM-U, 4DI) dans la liste des numéros de référence et ajoutez l'esclave "module d'E/S, compact, TOR, entrées" comme illustré dans la figure ci-après.

Tableau 10- 39 Ajout d'un esclave AS-i à la configuration des appareils

Insérez l'esclave AS-i	Résultat
	



### 10.4.1.2 Configuration des liaisons réseau logiques entre deux appareils AS-i

Maintenant que vous avez configuré le module maître AS-i CM 1243-2, vous êtes prêt à configurer vos liaisons réseau.

Dans le portail "Appareils & Réseaux", utilisez la "Vue du réseau" pour créer les liaisons réseau entre les appareils dans votre projet. Pour créer la liaison AS-i, sélectionnez le carré jaune (AS-i) sur le premier appareil. Tracez une ligne vers le carré AS-i sur le deuxième appareil. Relâchez le bouton de la souris : la liaison AS-i est créée.

Reportez-vous à "Configuration des appareils, Création d'une liaison réseau" (Page 135) pour plus d'informations.

### 10.4.1.3 Configuration des propriétés du maître AS-i CM 1243-2

Pour paramétrer l'interface AS-i, cliquez sur le carré AS-i jaune sur le module maître AS-i CM 1243-2. L'onglet "Propriétés" de la fenêtre d'inspection affiche l'interface AS-i.

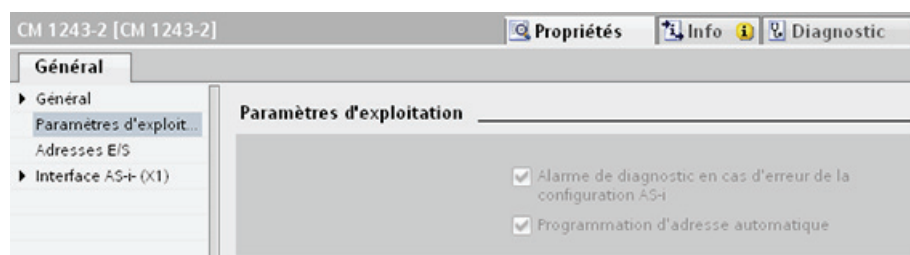
Dans la fenêtre d'inspection STEP 7, vous pouvez afficher, configurer et modifier les informations générales, les adresses et les paramètres de fonctionnement :

Tableau 10- 40 Propriétés du module maître AS-i CM 1243-2

Propriété	Description
Général	Nom du module maître AS-i CM 1243-2
Paramètres de fonctionnement	Paramètres pour la réponse du maître AS-i
Adresses E/S	Zone pour les adresses d'E/S des esclaves
Interface AS-i (X1)	Réseau AS-i affecté

#### Remarque

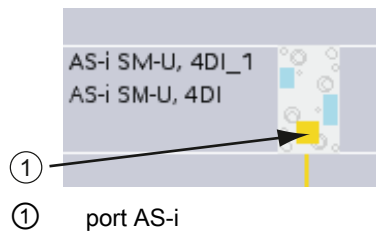
"Alarme de diagnostic pour erreurs dans la configuration AS-i" et "Programmation d'adresses automatique" sont toujours actifs et sont donc affichés en gris.



### 10.4.1.4 Affectation d'une adresse AS-i à un esclave AS-i

#### Configuration de l'interface esclave AS-i

Pour paramétrer l'interface AS-i, cliquez sur le carré AS-i jaune sur l'esclave AS-i. L'onglet "Propriétés" de la fenêtre d'inspection affiche l'interface AS-i.



#### Affectation de l'adresse esclave AS-i

Dans un réseau AS-i, une adresse d'esclave AS-i est affectée à chaque appareil. Cette adresse est comprise entre 0 et 31, l'adresse 0 étant toutefois réservée aux nouveaux esclaves.

Les adresses d'esclaves vont de 1(A ou B) à 31(A ou B) pour un total maximal de 62 esclaves. Toute adresse dans la plage de 1 à 31 peut être affectée à un esclave AS-i ; il est indifférent que les esclaves commencent à l'adresse 21 ou que le premier esclave reçoive effectivement l'adresse 1.

Entrez l'adresse d'esclave AS-i ici.

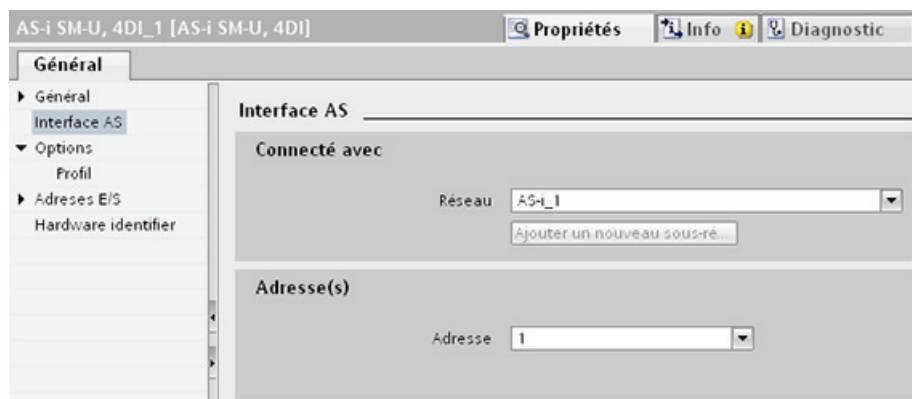


Tableau 10- 41 Paramètres pour l'interface AS-i

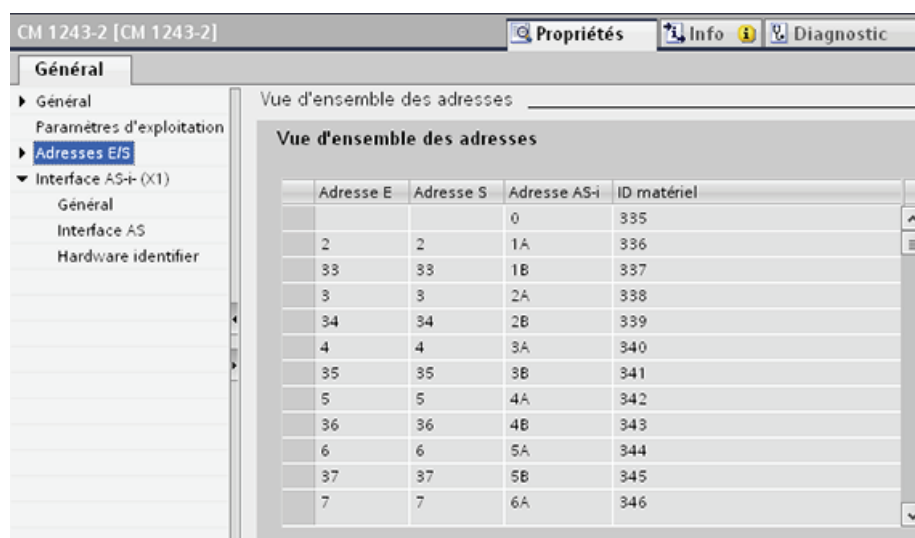
Paramètre	Description
Réseau	Nom du réseau auquel l'appareil est connecté.
Adresse(s)	Adresse AS-i affectée à l'esclave, comprise dans la plage allant de 1(A ou B) à 31(A ou B) pour un total maximal de 62 esclaves

## 10.4.2 Echange de données entre le programme utilisateur et les esclaves AS-i

### 10.4.2.1 Configuration STEP 7 de base

Le maître AS-i réserve une zone de données de 62 octets dans la zone d'E/S de la CPU. L'accès aux données TOR s'y fait en octets ; il y a un octet de données d'entrée et un octet de données de sortie pour chaque esclave.

L'affectation des liaisons AS-i des esclaves TOR AS-i aux bits de données de l'octet affecté est indiquée dans la fenêtre d'inspection du module CM 1243-2 maître AS-i.



Adresse E	Adresse S	Adresse AS-i	ID matériel
		0	335
2	2	1A	336
33	33	1B	337
3	3	2A	338
34	34	2B	339
4	4	3A	340
35	35	3B	341
5	5	4A	342
36	36	4B	343
6	6	5A	344
37	37	5B	345
7	7	6A	346

Vous pouvez accéder aux données des esclaves AS-i dans le programme utilisateur en utilisant les adresses d'E/S affichées avec les instructions logiques sur bits (par exemple, AND) ou les affectations de bits appropriées.

#### Remarque

"Affectation système" est automatiquement activé si vous ne configurez pas les esclaves AS-i avec STEP 7.

Si vous ne configurez pas d'esclaves, vous devez informer le module maître AS-i CM 1243-2 de la configuration de bus actuelle à l'aide de la fonction en ligne "Réelle > prévue".

#### Informations supplémentaires

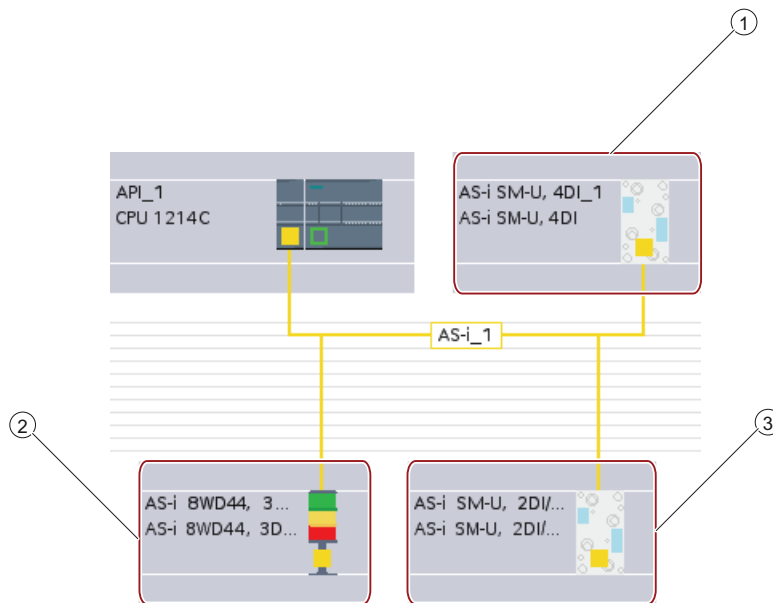
Vous trouverez des informations détaillées sur le module maître AS-i CM 1243-2 dans le manuel "Maître AS-i CM 1243-2 et module AS-i de découplage des données DCM 1271 pour SIMATIC S7-1200"

(<http://support.automation.siemens.com/WW/view/en/50414115/133300>).

### 10.4.2.2 Configuration d'esclaves avec STEP 7

#### Transfert de valeurs TOR AS-i

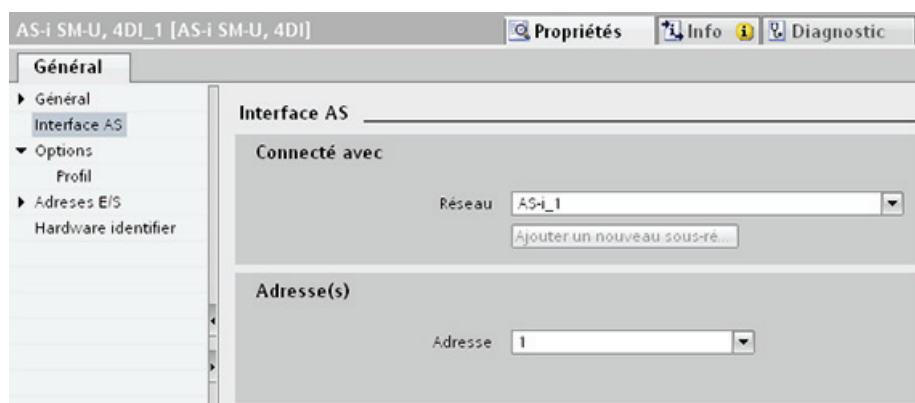
La CPU accède cycliquement aux entrées et sorties TOR des esclaves AS-i par le biais du module maître AS-i CM 1243-2. L'accès aux données se fait par les adresses d'E/S ou via un transfert d'enregistrements.



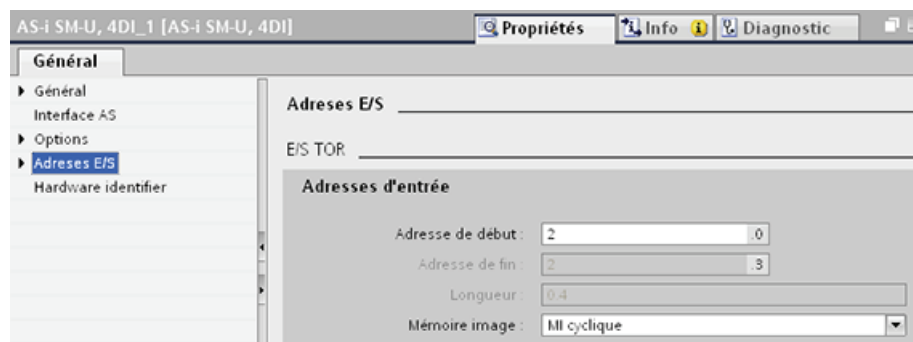
- ① Adresse esclave AS-i 1
- ② Adresse esclave AS-i 2A
- ③ Adresse esclave AS-i 3

L'accès aux données TOR s'y fait en octets (en d'autres termes, un octet est affecté à chaque esclave TOR AS-i). Lorsque vous configurez les esclaves AS-i dans STEP 7, l'adresse d'E/S pour l'accès aux données depuis le programme utilisateur apparaît dans la fenêtre d'inspection de l'esclave AS-i correspondant.

On a affecté l'adresse d'esclave 1 au module d'entrées TOR (AS-i SM-U, 4DI) dans le réseau AS-i ci-dessus. Lorsqu'on clique sur le module d'entrées TOR, l'onglet "Interface AS" des propriétés de l'appareil affiche l'adresse d'esclave, comme illustré ci-dessous :

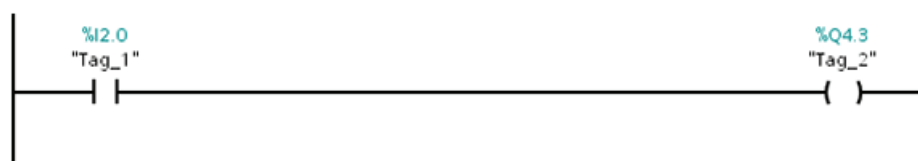


On a affecté l'adresse d'E/S 2 au module d'entrées TOR (AS-i SM-U, 4DI) dans le réseau AS-i ci-dessus. Lorsqu'on clique sur le module d'entrées TOR, l'onglet "Adresses E/S" des propriétés de l'appareil affiche l'adresse d'E/S, comme illustré ci-dessous :



Vous pouvez accéder aux données des esclaves AS-i dans le programme utilisateur en utilisant leurs adresses d'E/S avec les instructions logiques sur bits (par exemple, AND) ou les affectations de bits appropriées. Le programme simple suivant illustre comment fonctionne l'affectation :

L'entrée 2.0 est interrogée dans ce programme. Dans le système AS-i, cette entrée appartient à l'esclave 1 (octet d'entrée 2, bit 0). La sortie 4.3, qui est alors mise à 1, correspond à l'esclave AS-i (octet de sortie 4, bit 3).



## Transfert de valeurs analogiques AS-i

Vous pouvez accéder aux données analogiques d'un esclave AS-i via la mémoire image de la CPU si vous avez configuré cet esclave AS-i comme esclave analogique dans STEP 7.

Si vous n'avez pas configuré l'esclave analogique dans STEP 7, vous pouvez accéder aux données de l'esclave AS-i uniquement par le biais des fonctions acycliques (interface d'enregistrement). Dans le programme utilisateur de la CPU, la lecture et l'écriture des appels AS-i se font à l'aide des instructions de périphérie décentralisée RDREC (Lire enregistrement) et WRREC (Ecrire enregistrement).

---

### Remarque

Une configuration des esclaves AS-i définie via STEP 7 et chargée dans la station S7 est transférée par la CPU dans le module maître AS-i CM 1243-2 au démarrage de la station S7. Toute configuration existante déterminée via la fonction en ligne "Affectation système" (Page 515) (réelle -> prévue) est écrasée.

---

### Informations supplémentaires

Vous trouverez des informations détaillées sur le module maître AS-i CM 1243-2 dans le manuel "Maître AS-i CM 1243-2 et module AS-i de découplage des données DCM 1271 pour SIMATIC S7-1200" (<http://support.automation.siemens.com/WW/view/en/50414115/133300>).

### 10.4.3 Instructions de périphérie décentralisée

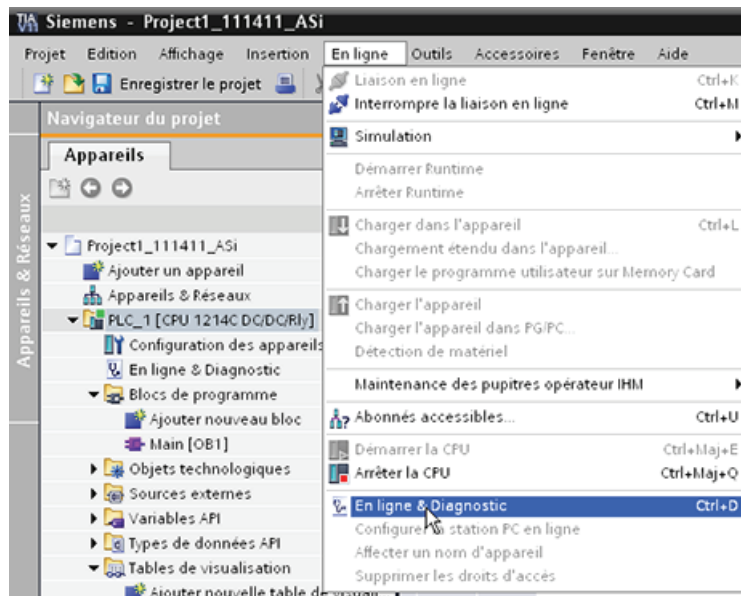
Reportez-vous à "Périphérie décentralisée (PROFINET, PROFIBUS ou AS-i)" (Page 291) pour obtenir des informations sur la manière d'utiliser les instructions de périphérie décentralisée avec ces réseaux de communication.

### 10.4.4 Utilisation d'outils en ligne AS-i

#### Changement des modes de fonctionnement AS-i en ligne

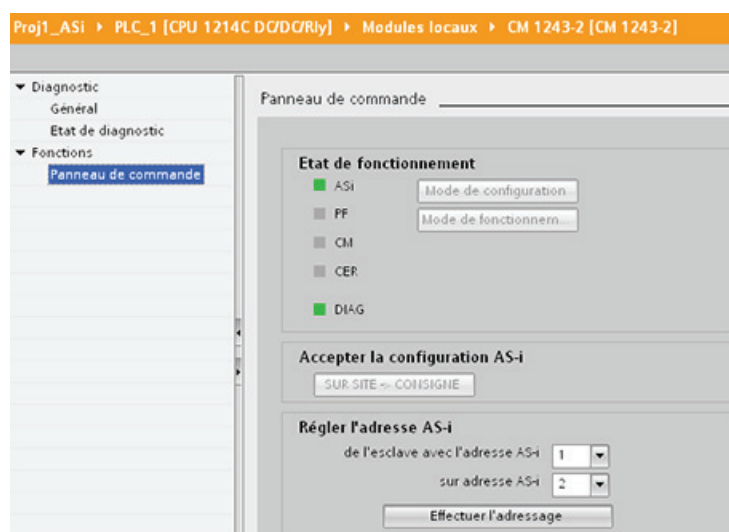
Vous devez aller en ligne pour afficher les modes de fonctionnement AS-i et en changer.

Pour aller en ligne, vous devez d'abord sélectionner le module maître AS-i CM1243-2 dans la configuration des appareils, puis cliquer sur le bouton "En ligne" dans la barre d'outils. Sélectionnez ensuite la commande "En ligne et diagnostic" du menu "En ligne".



Il existe deux modes de fonctionnement AS-i :

- le mode protégé :
  - Vous ne pouvez pas modifier les adresses d'esclave AS et d'E/S CPU.
  - La DEL "CM" verte est éteinte.
- le mode de configuration :
  - Vous pouvez apporter les modifications requises aux adresses d'esclave AS et d'E/S CPU.
  - La DEL "CM" verte est allumée.



Vous pouvez modifier l'adresse d'esclave AS-i dans la zone "Régler l'adresse AS-i". Un nouvel esclave n'ayant pas encore reçu d'adresse a toujours l'adresse 0. Il est identifié par le maître comme nouvel esclave auquel aucune adresse n'a encore été affectée et n'est pas inclus dans la communication normale tant qu'il n'a pas reçu d'adresse.

## Erreur de configuration

Lorsque la DEL "CER" jaune est allumée, il y a une erreur dans la configuration de l'esclave AS-i. Cliquez sur le bouton "Réelle > prévue" pour substituer à la configuration d'esclave du module maître AS-i CM 1243-2 la configuration d'esclave du réseau de terrain AS-i.

## 10.5 Communication S7

### 10.5.1 Instructions GET et PUT

Vous pouvez utiliser les instructions GET et PUT pour communiquer avec des CPU S7 via des liaisons PROFINET et PROFIBUS :

- Accès aux données dans une CPU éloignée : Une CPU S7-1200 peut utiliser uniquement des adresses absolues dans le champ d'entrée ADDR\_x pour accéder à des variables dans des CPU éloignées (S7-200/300/400/1200).
- Accès aux données dans un DB standard : Une CPU S7-1200 peut utiliser uniquement des adresses absolues dans le champ d'entrée ADDR\_x pour accéder aux variables d'un DB standard dans une CPU S7 éloignée.
- Accès aux données dans un DB optimisé : Une CPU S7-1200 ne peut pas accéder aux variables d'un DB optimisé dans une CPU S7-1200 éloignée.
- Accès aux données dans une CPU locale : Une CPU S7-1200 peut utiliser soit des adresses absolues, soit des adresses symboliques comme entrées pour les champs d'entrée RD\_x ou SD\_x des instructions GET et PUT respectivement.



Tableau 10- 42 Instructions GET et PUT

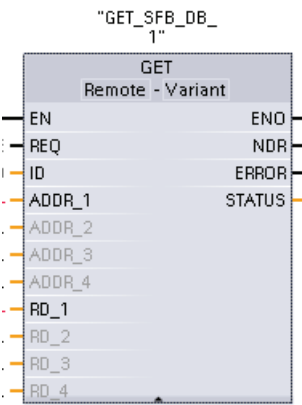
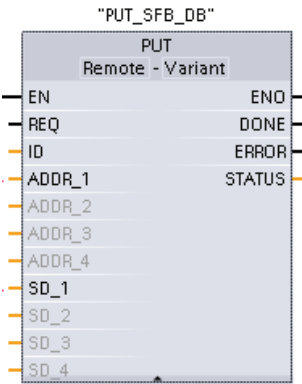
CONT/LOG	SCL	Description
<p>"GET_SFB_DB_1"</p> 	<pre>"GET_DB" (   req:= _bool_in_,   ID:= _word_in_,   ndr=&gt; _bool_out_,   error=&gt; _bool_out_,   status=&gt; _word_out_,   addr_1:= _remote_inout_,   [...addr_4:= _remote_inout_,]   rd_1:= _variant_inout_   [...rd_4:= _variant_inout_]);</pre>	<p>Utilisez l'instruction GET pour lire des données dans une CPU S7 éloignée. La CPU éloignée peut être à l'état MARCHÉ ou à l'état ARRÊT.</p> <p>STEP 7 crée automatiquement le DB lorsque vous insérez l'instruction.</p>
<p>"PUT_SFB_DB"</p> 	<pre>"PUT_DB" (   req:= _bool_in_,   ID:= _word_in_,   done=&gt; _bool_out_,   error=&gt; _bool_out_,   status=&gt; _word_out_,   addr_1:= _remote_inout_,   [...addr_4:= _remote_inout_,]   sd_1:= _variant_inout_,   [...sd_4:= _variant_inout_]);</pre>	<p>Utilisez l'instruction PUT pour écrire des données dans une CPU S7 éloignée. La CPU éloignée peut être à l'état MARCHÉ ou à l'état ARRÊT.</p> <p>STEP 7 crée automatiquement le DB lorsque vous insérez l'instruction.</p>

Tableau 10- 43 Types de données pour les paramètres

Paramètre et type	Type de données	Description
REQ	Input	Bool
ID	Input	CONN_PRG (Word)
NDR (GET)	Output	Bool
DONE (PUT)	Output	Bool

Un front montant déclenche l'opération.

ID de liaison S7 (hexa)

Nouvelles données prêtes (New Data Ready) :

- 0: La tâche n'a pas encore commencé ou est encore en cours d'exécution.
- 1: La tâche a été achevée avec succès.

DONE :

- 0: La tâche n'a pas encore commencé ou est encore en cours d'exécution.
- 1: La tâche a été achevée avec succès.

Paramètre et type		Type de données	Description
ERROR STATUS	Output Output	Bool Word	<ul style="list-style-type: none"> <li>• ERROR=0 Valeur de STATUS : <ul style="list-style-type: none"> <li>- 0000H : Ni erreur ni avertissement</li> <li>- &lt;&gt; 0000H : Avertissement ; STATUS fournit des informations détaillées.</li> </ul> </li> <li>• ERROR=1 Il y a une erreur. STATUS fournit des informations détaillées sur la nature de l'erreur.</li> </ul>
ADDR_1	InOut	Distant	Pointeur désignant les zones de mémoire dans la CPU éloignée qui contiennent les données à lire (GET) ou les données envoyées (PUT).
ADDR_2	InOut	Distant	
ADDR_3	InOut	Distant	
ADDR_4	InOut	Distant	
RD_1 (GET) SD_1 (PUT)	InOut	Variant	Pointeur désignant les zones de mémoire dans la CPU locale qui contiennent les données lues (GET) ou à envoyer (PUT).
RD_2 (GET) SD_2 (PUT)	InOut	Variant	Types de données autorisés : Bool (un bit unique autorisé), Byte, Char, Word, Int, DWord, DInt ou Real.
RD_3 (GET) SD_3 (PUT)	InOut	Variant	Remarque : Si le pointeur accède à un DB, vous devez indiquer l'adresse absolue, par exemple
RD_4 (GET) SD_4 (PUT)	InOut	Variant	P# DB10.DBX5.0 Byte 10 Dans ce cas, 10 représente le nombre d'octets à lire ou à envoyer.

Vous devez vous assurer que la longueur (nombre d'octets) et les types de données pour les paramètres ADDR\_x (CPU éloignée) et RD\_x ou SD\_x (CPU locale) coïncident. Le nombre après l'identificateur "Byte" est le nombre d'octets désignés par le paramètre ADDR\_x, RD\_x ou SD\_x .

### Remarque

Le nombre total d'octets reçus dans une instruction GET et le nombre total d'octets envoyés dans une instruction PUT sont limités. Ces limites dépendent de combien des quatre zones de mémoire et d'adresses possibles vous utilisez :

- Si vous utilisez uniquement ADDR\_1 et RD\_1/SD\_1, une instruction GET peut lire 222 octets et une instruction PUT peut envoyer 212 octets.
- Si vous utilisez ADDR\_1, RD\_1/SD\_1, ADDR\_2 et RD\_2/SD\_2, une instruction GET peut lire 218 octets au total et une instruction PUT peut envoyer 196 octets au total.
- Si vous utilisez ADDR\_1, RD\_1/SD\_1, ADDR\_2, RD\_2/SD\_2, ADDR\_3 et RD\_3/SD\_3 , une instruction GET peut lire 214 octets au total et une instruction PUT peut envoyer 180 octets au total.
- Si vous utilisez ADDR\_1, RD\_1/SD\_1, ADDR\_2, RD\_2/SD\_2, ADDR\_3, RD\_3/SD\_3, ADDR\_4, RD\_4/SD\_4, une instruction GET peut lire 210 octets au total et une instruction PUT peut envoyer 164 octets au total.

La somme des nombres d'octets de chacun de vos paramètres de zone de mémoire et d'adresses doit être inférieure ou égale aux limites définies. Si vous dépassez ces limites, l'instruction GET ou PUT renvoie une erreur.

En présence d'un front montant du paramètre REQ, l'opération de lecture (GET) ou l'opération d'écriture (PUT) charge les paramètres ID, ADDR\_1 et RD\_1 (GET) ou SD\_1 (PUT).

- Pour GET : La CPU éloignée renvoie les données demandées dans les zones de réception (RD\_x), en commençant au cycle suivant. Le paramètre NDR est mis à 1 lorsque l'opération de lecture s'est achevée sans erreur. Une nouvelle opération ne peut être lancée qu'une fois l'opération précédente achevée.
- Pour PUT : La CPU locale commence à envoyer les données (SD\_x) vers l'emplacement de mémoire (ADDR\_x) dans la CPU éloignée. La CPU éloignée envoie un accusé de réception lorsque l'opération d'écriture s'est achevée sans erreur. Le paramètre DONE de l'instruction PUT est alors mis à 1. Une nouvelle opération d'écriture ne peut être lancée qu'une fois l'opération précédente achevée.

#### Remarque

Pour garantir la cohérence des données, évaluez toujours le moment où l'opération s'est achevée (NDR = 1 pour GET ou DONE = 1 pour PUT) avant d'accéder aux données ou de déclencher une nouvelle opération de lecture ou d'écriture.

Les paramètres ERROR et STATUS fournissent des informations sur l'état de l'opération de lecture (GET) ou d'écriture (PUT).

Tableau 10- 44 Informations d'erreur

ERROR	STATUS (décimal)	Description
0	11	<ul style="list-style-type: none"> <li>• La nouvelle tâche ne peut démarrer, car la tâche précédente n'est pas encore achevée.</li> <li>• La tâche est en cours de traitement dans une classe de priorité de priorité inférieure.</li> </ul>
0	25	La communication a démarré. La tâche est en cours de traitement.
1	1	Problèmes de communication, tels que : <ul style="list-style-type: none"> <li>• Description de la liaison non chargée (locale ou éloignée)</li> <li>• Interruption de la liaison (par exemple, câble, CPU hors tension ou CM/CB/CP à l'état ARRET)</li> <li>• Liaison au partenaire non encore établie</li> </ul>
1	2	Accusé de réception négatif du partenaire. La tâche ne peut pas être exécutée.
1	4	Erreurs de longueur ou de type de données dans les pointeurs de zone de réception ou d'émission (RD_x pour GET ou SD_x pour PUT).
1	8	Erreur d'accès sur la CPU partenaire
1	10	Accès à la mémoire utilisateur locale impossible (par exemple, tentative d'accéder à un DB effacé)
1	12	A l'appel du SFB : <ul style="list-style-type: none"> <li>• Un DB d'instance qui n'appartient pas à GET ou PUT a été indiqué.</li> <li>• Ce n'est pas un DB d'instance qui était indiqué mais un DB global.</li> <li>• Le DB d'instance est resté introuvable (chargement d'un nouveau DB d'instance).</li> </ul>

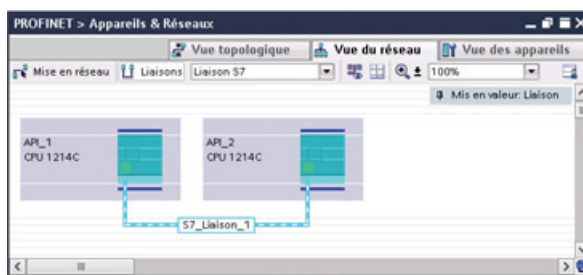
ERROR	STATUS (décimal)	Description
1	20	<ul style="list-style-type: none"> <li>Dépassement du nombre maximum de tâches/instances en parallèle</li> <li>Les instances ont été rechargées à l'état MARCHE de la CPU.</li> </ul> Cet état est possible à la première exécution de l'instruction GET ou PUT.
1	27	Il n'y a pas d'instruction GET ou PUT correspondante dans la CPU.

## 10.5.2 Création d'une liaison S7

Le type de liaison que vous sélectionnez crée une liaison de communication vers une station partenaire. La liaison est configurée, établie et automatiquement surveillée.

Dans le portail "Appareils & Réseaux", utilisez la "Vue du réseau" pour créer les liaisons réseau entre les appareils dans votre projet. Cliquez d'abord sur l'onglet "Liaisons", puis sélectionnez le type de liaison dans la liste déroulante juste à droite (une liaison S7, par exemple). Cliquez sur le carré vert (PROFINET) du premier appareil et tracez une ligne vers le carré PROFINET du deuxième appareil. Relâchez le bouton de la souris : la liaison PROFINET est créée.

Reportez-vous à "Création d'une liaison réseau" (Page 135) pour plus d'informations.



Cliquez sur le bouton "Mis en valeur : Liaison" pour accéder à la boîte de dialogue de configuration "Propriétés" de l'instruction de communication.

## 10.5.3 Configuration du routage local/partenaire entre deux appareils

### Configuration des paramètres généraux

Vous indiquez les paramètres de communication dans la boîte de dialogue de configuration "Propriétés" de l'instruction de communication. Cette boîte de dialogue apparaît au bas de la page lorsque vous sélectionnez n'importe quelle partie de l'instruction.

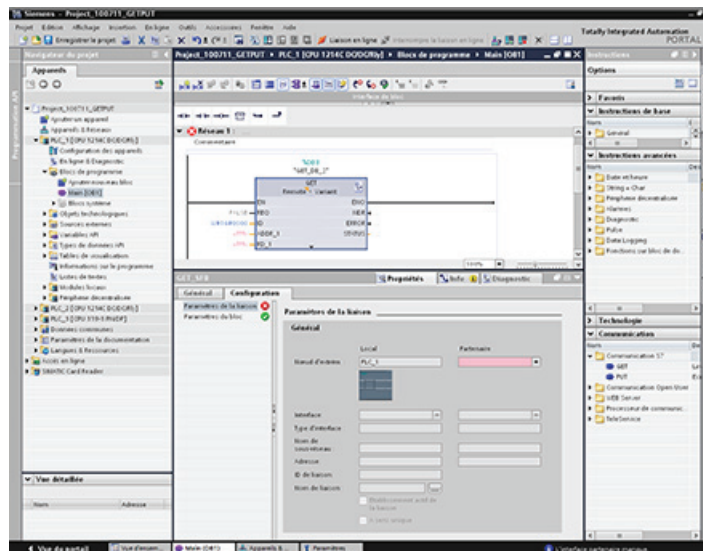
Reportez-vous à "Configuration des appareils, Configuration du routage local/partenaire (Page 136)" pour plus d'informations.

Vous définissez les TSAP ou ports à utiliser dans la section "Détails de l'adresse" de la boîte de dialogue "Paramètres de liaison". Vous indiquez le point TSAP ou le port d'une liaison dans la CPU dans la zone "TSAP local". Vous indiquez le point TSAP ou le port affecté à la liaison dans votre CPU partenaire dans la zone "TSAP partenaire".

## 10.5.4 Paramétrage de la liaison GET/PUT

Le paramétrage de la liaison pour instructions GET/PUT est une aide utilisateur qui vous assiste lors de la configuration de liaisons de communication CPU à CPU S7.

Après l'insertion d'un bloc GET ou PUT, le paramétrage de la liaison pour GET/PUT est démarré :



La fenêtre d'inspection affiche les propriétés de la liaison lorsque vous sélectionnez n'importe quelle partie de l'instruction. Vous pouvez indiquer les paramètres de communication dans l'onglet "Configuration" des "Propriétés" de l'instruction de communication.

### 10.5.4.1 Paramètres de la liaison

La page "Paramètres de la liaison" vous permet de configurer la liaison S7 nécessaire et de définir le paramètre "ID de liaison" auquel fait référence le paramètre "ID" du bloc GET/PUT. La page contient des informations sur le noeud d'extrémité local et vous permet de définir l'interface locale. Vous pouvez également définir le noeud d'extrémité partenaire.

La page "Paramètres du bloc" vous permet de configurer les autres paramètres du bloc.

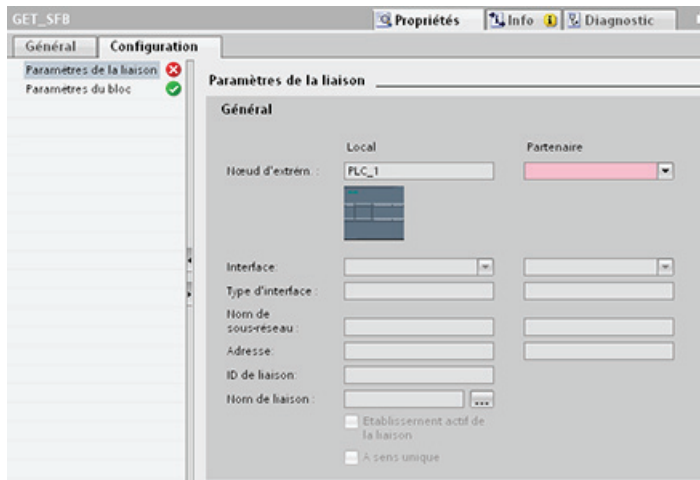


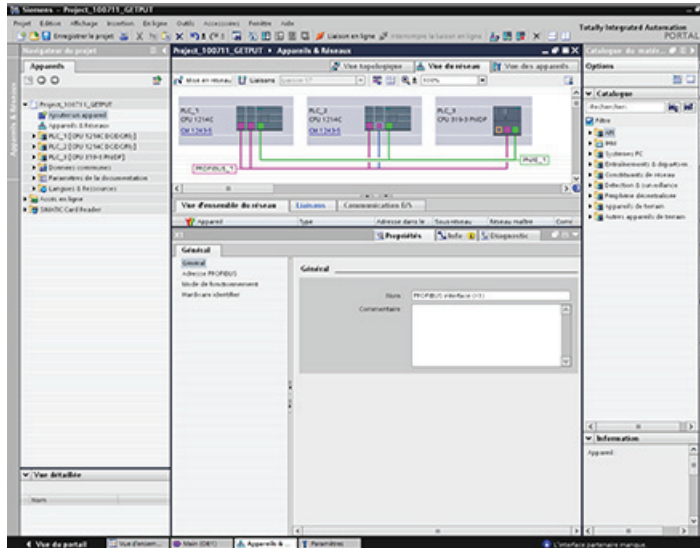
Tableau 10- 45 Paramètres de la liaison : définitions générales

Paramètre	Définition	
Paramètres de la liaison : Général	Noeud d'extrémité	Noeud d'extrémité local : Nom affecté à la CPU locale Noeud d'extrémité partenaire : Nom affecté à la CPU partenaire (éloignée) Remarque : Le système affiche tous les partenaires de liaison S7 potentiels du projet en cours ainsi que l'option "non spécifié" dans la liste déroulante "Noeud d'extrémité partenaire". Un partenaire non spécifié représente un partenaire de communication ne se trouvant pas actuellement dans le projet STEP 7 (par exemple, un partenaire de communication d'appareil tiers).
	Interface	Nom affecté aux interfaces Remarque : Vous pouvez modifier la liaison en changeant les interfaces locale et partenaire.
	Type d'interface	Type de l'interface
	Nom de sous-réseau	Nom affecté aux sous-réseaux
	Adresse	Adresses IP affectées Remarque : Vous pouvez indiquer l'adresse éloignée d'un appareil tiers pour un partenaire de communication "non spécifié".
	ID de liaison	Numéro d'identification : généré automatiquement par le paramétrage de la liaison pour GET/PUT
	Nom de la liaison	Emplacement de stockage des données des CPU locale et partenaire : généré automatiquement par le paramétrage de la liaison pour GET/PUT
	Initialisation de la liaison	Case à cocher pour sélectionner la CPU locale en tant que liaison active
A sens unique	Case en lecture seule indiquant s'il s'agit d'une liaison à sens unique ou à double sens.  Remarque : Dans une liaison GET/PUT PROFINET, l'appareil local et l'appareil partenaire peuvent tous deux agir comme client ou serveur. Cela permet une liaison à double sens, auquel cas la case "A sens unique" n'est pas cochée.  Dans certains cas pour une liaison GET/PUT PROFIBUS, l'appareil partenaire peut agir uniquement en tant que serveur (S7-300, par exemple) et la case "A sens unique" est cochée.	



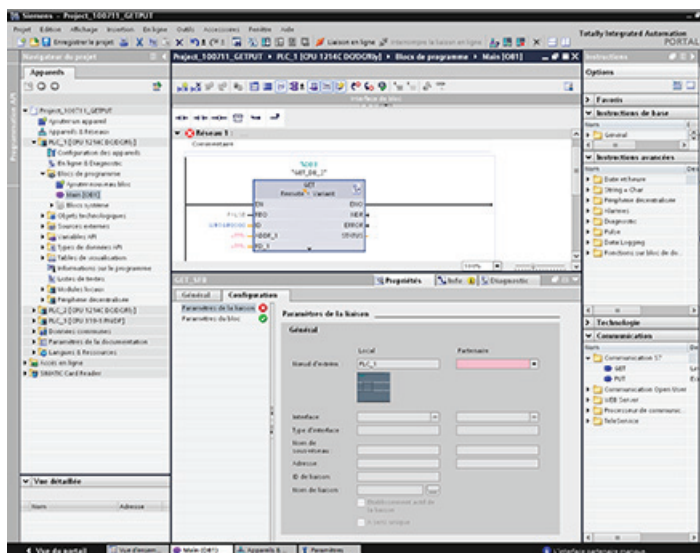
### 10.5.4.2 Configuration d'une liaison S7 CPU à CPU

Etant donné la configuration de PLC\_1, PLC\_2 et PLC\_3 illustrée dans la figure ci-dessous, insérez des blocs GET ou PUT pour "PLC\_1".



L'onglet "Propriétés" s'affiche automatiquement dans la fenêtre d'inspection avec les sélections de menu suivantes pour l'instruction GET ou PUT :

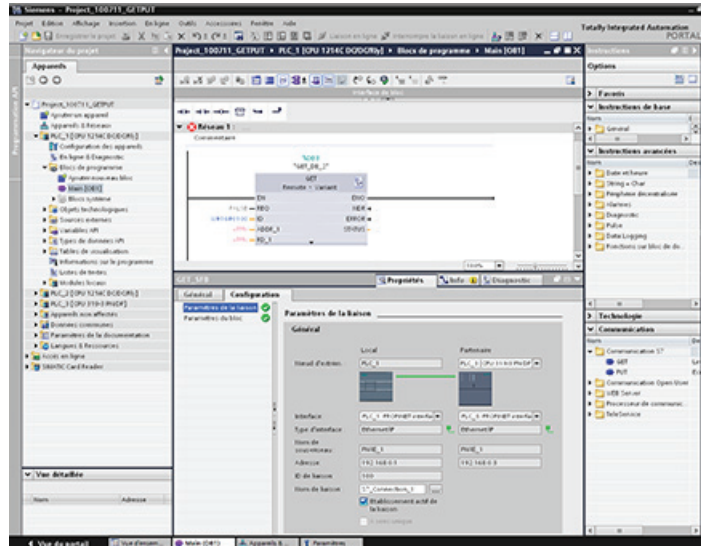
- "Configuration"
- "Paramètres de la liaison"





## Configuration d'une liaison S7 PROFINET

Sélectionnez "PLC\_3" pour "Noeud d'extrémité partenaire".



Le système réagit avec les modifications suivantes :

Tableau 10- 46 Valeurs dans Paramètres de la liaison, Général

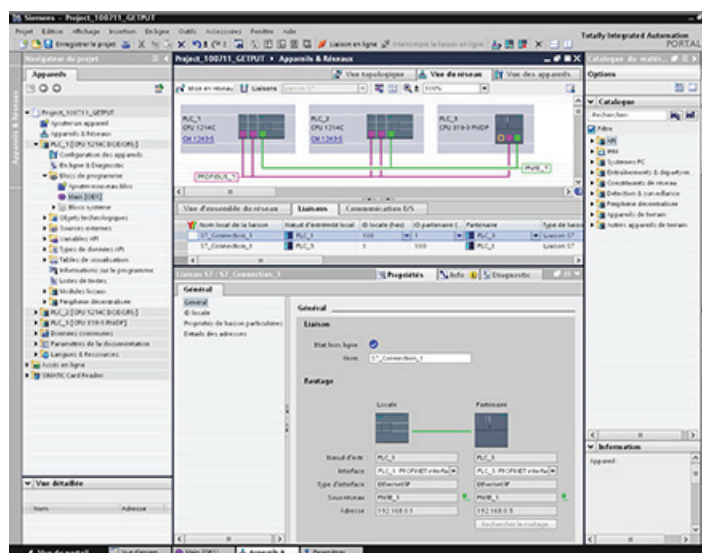
Paramètre	Définition
Paramètres de la liaison : Général	<p>Noeud d'extrémité</p> <p>"Noeud d'extrémité local" contient "PLC_1" en lecture seule. Le champ "Noeud d'extrémité partenaire" contient "PLC_3[CPU319-3PN/DP]".</p> <ul style="list-style-type: none"> <li>La couleur passe de rouge à blanc.</li> <li>L'image de l'appareil partenaire s'affiche.</li> <li>Une ligne de connexion apparaît entre l'image de "PLC_1" et l'image de "PLC_3" (ligne Ethernet verte).</li> </ul>
Interface	<p>"Interface" pour "Local" contient "CPU1214C DC/DC/DC, interface PROFINET (R0/S1)". "Interface" pour "Partenaire" contient "CPU319-3PN/DP, interface PROFINET (R0/S2)".</p>
Type d'interface	<p>"Type d'interface" pour "Local" contient "Ethernet/IP" ; le contrôle est en lecture seule. "Type d'interface" pour "Partenaire" contient "Ethernet/IP" ; le contrôle est en lecture seule. Des images des types d'interface sont montrées à droite des "types d'interface" local et partenaire (icône Ethernet verte).</p>
Nom de sous-réseau	<p>"Nom de sous-réseau" pour "Local" contient "PN/IE_1" ; le contrôle est en lecture seule. "Nom de sous-réseau" pour "Partenaire" contient "PN/IE_1" ; le contrôle est en lecture seule.</p>

Paramètre	Définition
Adresse	"Adresse" pour "Local" contient l'adresse IP locale ; le contrôle est en lecture seule. "Adresse" pour "Partenaire" contient l'adresse IP partenaire ; le contrôle est en lecture seule.
ID de liaison	"ID de liaison" contient "100". Dans le bloc Main [OB1] dans l'éditeur de programmes, le paramètre "ID de liaison" du bloc GET/PUT contient également la valeur "100".
Nom de la liaison	Ce champ contient le nom de liaison par défaut (par exemple, "S7_Connection_1") ; le contrôle est activé.
Initialisation de la liaison	Case activée et cochée pour sélectionner la CPU locale en tant que liaison active
A sens unique	Case en lecture seule et non cochée. Remarque : "PLC_1" (une CPU 1214CDC/DC/Relais S7-1200) et "PLC_3" (une CPU 319-3PN/DP S7-300) peuvent tous deux agir comme serveur et comme client dans une liaison GET/PUT PROFINET, ce qui permet une liaison à double sens.

La couleur de l'icône GET/PUT dans la vue des propriétés change également de rouge à vert.

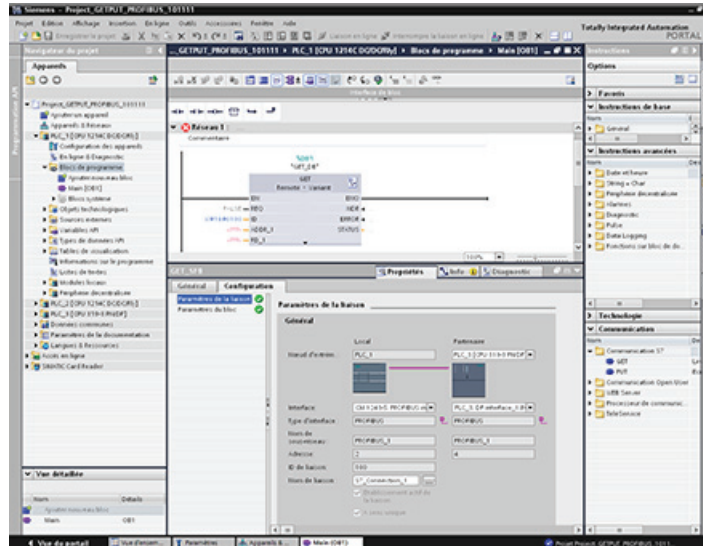
### Liaison S7 PROFINET achevée

Dans la vue du réseau, une liaison S7 bilatérale est visible dans la table des liaisons entre "PLC\_1" et "PLC\_3".



## Configuration d'une liaison S7 PROFIBUS

Sélectionnez "PLC\_3" pour "Noeud d'extrémité partenaire".



Le système réagit avec les modifications suivantes :

Tableau 10- 47 Valeurs dans Paramètres de la liaison, Général

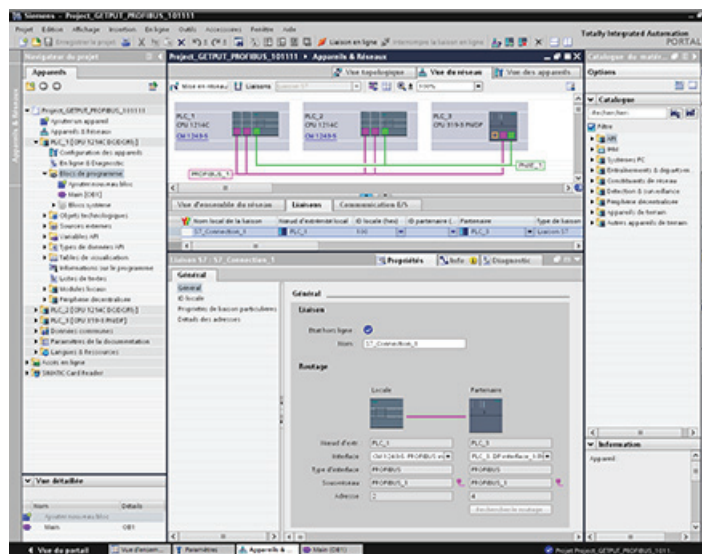
Paramètre	Définition	
Paramètres de la liaison : Général	Noeud d'extrémité	"Noeud d'extrémité local" contient "PLC_1" en lecture seule. Le champ "Noeud d'extrémité partenaire" contient "PLC_3[CPU319-3PN/DP]". <ul style="list-style-type: none"> <li>La couleur passe de rouge à blanc.</li> <li>L'image de l'appareil partenaire s'affiche.</li> <li>Une ligne de connexion apparaît entre l'image de "PLC_1" et l'image de "PLC_3" (ligne PROFIBUS violette).</li> </ul>
	Interface	"Interface" pour "Local" contient "CPU1214C DC/DC/DC, interface PROFIBUS (R0/S1)". "Interface" pour "Partenaire" contient "CPU319-3PN/DP, interface PROFIBUS (R0/S2)".
	Type d'interface	"Type d'interface" pour "Local" contient "PROFIBUS" ; le contrôle est en lecture seule. "Type d'interface" pour "Partenaire" contient "PROFIBUS" ; le contrôle est en lecture seule. Des images des types d'interface sont montrées à droite des "types d'interface" local et partenaire (icône PROFIBUS violette).
	Nom de sous-réseau	"Nom de sous-réseau" pour "Local" contient "PROFIBUS_1" ; le contrôle est en lecture seule. "Nom de sous-réseau" pour "Partenaire" contient "PROFIBUS_1" ; le contrôle est en lecture seule.

Paramètre	Définition
Adresse	"Adresse" pour "Local" contient l'adresse IP locale ; le contrôle est en lecture seule. "Adresse" pour "Partenaire" contient l'adresse IP partenaire ; le contrôle est en lecture seule.
ID de liaison	"ID de liaison" contient "100". Dans le bloc Main [OB1] dans l'éditeur de programmes, le paramètre "ID de liaison" du bloc GET/PUT contient également la valeur "100".
Nom de la liaison	Ce champ contient le nom de liaison par défaut (par exemple, "S7_Connection_1") ; le contrôle est activé.
Initialisation de la liaison	Case en lecture seule et cochée pour sélectionner la CPU locale en tant que liaison active
A sens unique	Case en lecture seule et cochée. Remarque : "PLC_3" (une CPU 319-3PN/DP S7-300) peut agir uniquement en tant que serveur (et ne peut pas être également client) dans une liaison GET/PUT PROFIBUS, ce qui permet uniquement une liaison à sens unique.

La couleur de l'icône GET/PUT dans la vue des propriétés change également de rouge à vert.

### Liaison S7 PROFIBUS achevée

Dans la vue du réseau, une liaison S7 à sens unique est visible dans la table des liaisons entre "PLC\_1" et "PLC\_3".



## Serveur Web

Le serveur Web pour le S7-1200 fournit un accès par pages Web aux données concernant votre CPU et aux données de processus contenues dans la CPU.

### Pages Web standard

Le S7-1200 comprend des pages Web standard auxquelles vous pouvez accéder depuis un navigateur Web sur votre PC (Page 535) :

- Introduction (Page 539) : point d'entrée dans les pages Web standard
- Page d'accueil (Page 540) : informations générales sur la CPU
- Identification (Page 541) : informations détaillées sur la CPU, y compris numéros de série, de référence et de version
- Informations modules (Page 542) : informations sur les modules dans le châssis local
- Communication (Page 545) : informations sur les adresses de réseau, les propriétés physiques des interfaces de communication et les statistiques de communication
- Tampon de diagnostic (Page 541) : mémoire tampon de diagnostic
- Etat des variables (Page 546) : variables et E/S de la CPU accessibles par adresse ou nom de variable API
- Journaux (Page 548) : fichiers journaux stockés en interne dans la CPU ou sur une carte mémoire
- Mise à jour du firmware (Page 551) : mise à jour du firmware dans votre CPU

Ces pages sont intégrées dans le S7-1200. Pour plus de détails sur les pages Web standard et sur la façon d'y accéder, consultez la partie Pages Web standard (Page 535).

### Pages Web personnalisées

Le S7-1200 fournit également une aide à la création de pages Web personnalisées qui peuvent accéder aux données de la CPU. Vous pouvez concevoir ces pages avec le logiciel auteur HTML de votre choix et inclure des commandes AWP (Automation Web Programming) prédéfinies dans votre code HTML pour accéder aux données de la CPU. Reportez-vous au chapitre Pages Web personnalisées (Page 553) pour obtenir des informations spécifiques sur la conception de pages Web personnalisées et sur la configuration et la programmation associées dans STEP 7.

## Exigences concernant le navigateur Web

Les navigateurs Web suivants prennent en charge le serveur Web :

- Internet Explorer 8.0 ou plus
- Mozilla Firefox 3.0 ou plus
- Opera 11.0 ou plus

Les restrictions liées au navigateur, en raison des interférences possibles avec l'affichage de pages Web standard ou personnalisées, sont présentées au paragraphe Contraintes (Page 593).

## 11.1 Activation du serveur Web

Vous activez le serveur Web dans STEP 7, dans la configuration d'appareil pour la CPU à laquelle vous avez l'intention de vous connecter.

Procédez comme suit pour activer le serveur Web :

1. Sélectionnez la CPU dans la vue Configuration des appareils.
2. Dans la fenêtre d'inspection, sélectionnez "Serveur Web" dans les propriétés de la CPU.
3. Cochez la case "Activer le serveur Web sur ce module".
4. Si vous avez besoin d'un accès sécurisé au serveur Web, cochez la case "Autoriser l'accès uniquement via HTTPS".

### ATTENTION

L'accès de personnes non autorisées à la CPU ou l'affectation de valeurs invalides aux variables API peut perturber le fonctionnement du processus, ce qui peut entraîner la mort, des blessures graves et des dommages matériels importants.

Comme la validation du serveur Web permet à des utilisateurs ayant les droits "admin" de changer l'état de fonctionnement, d'écrire dans des données de l'API et d'actualiser le firmware, Siemens vous recommande de mettre en oeuvre la politique de sécurité suivante :

- N'autorisez l'accès au serveur Web qu'avec le protocole HTTPS.
- Protégez la CPU (Page 175) avec un mot de passe fort. Les mots de passe forts contiennent au moins huit caractères, lettres, chiffres et caractères spéciaux mélangés, ne sont pas des mots du dictionnaire et ne sont pas des noms ou identifiants pouvant être déduits de vos données personnelles. Ne divulguez pas le mot de passe et changez-en fréquemment.
- Effectuez des recherches d'erreur et des vérifications de plage sur vos variables dans la logique de programme, car les utilisateurs des pages Web ont la possibilité de donner des valeurs invalides aux variables API.

Une fois que vous avez chargé la configuration d'appareil, vous pouvez utiliser les pages Web standard pour accéder à la CPU. Si vous activez l'actualisation automatique, les pages Web standard sont rafraîchies toutes les dix secondes.

Si vous avez créé des pages Web personnalisées, vous pouvez y accéder par le menu des pages Web standard.

---

**Remarque**

Si un "chargement dans la CPU à l'état MARCHE" (Page 739) est en cours, les valeurs de données ne sont pas actualisées et vous ne pouvez pas écrire de valeurs dans les pages Web standard et personnalisées tant que le chargement dans la CPU n'est pas achevé. Toute tentative d'écriture de valeurs de données alors que le chargement est en cours est rejetée.

---

## 11.2 Pages Web standard

### 11.2.1 Accès aux pages Web standard depuis le PC

Procédez comme suit pour accéder aux pages Web standard du S7-1200 depuis un PC :

1. Vérifiez que le S7-1200 et le PC se trouvent sur un réseau Ethernet commun ou sont reliés directement l'un à l'autre au moyen d'un câble Ethernet standard.
2. Ouvrez un navigateur Web et entrez l'URL "http://ww.xx.yy.zz", "ww.xx.yy.zz" correspondant à l'adresse IP de la CPU S7-1200.

Le navigateur Web ouvre la page Introduction.

---

**Remarque**

Si votre accès Web n'autorise pas la connexion directe à une adresse IP, contactez votre administrateur informatique. Votre système d'exploitation ou environnement Web peut également imposer d'autres contraintes (Page 593).

---

Vous pouvez également demander à votre navigateur Web d'accéder à une page Web standard spécifique. Pour ce faire, entrez l'URL sous la forme "http://ww.xx.yy.zz/<page>.html", <page> correspondant à l'une des pages Web standard :

- start (Page 540) : informations générales sur la CPU
- identification (Page 541) : informations détaillées sur la CPU, y compris numéros de série, de référence et de version
- module (Page 542) : informations sur les modules dans le châssis local
- communication (Page 545) : informations sur les adresses de réseau, les propriétés physiques des interfaces de communication et les statistiques de communication
- diagnostic (Page 541) : mémoire tampon de diagnostic
- variable (Page 546) : variables et E/S de la CPU accessibles par adresse ou nom de variable API

- datalog (Page 548) : fichiers journaux stockés en interne dans la CPU ou sur une carte mémoire
- updatefirmware (Page 551) : page permettant de mettre à jour le firmware dans votre CPU à partir d'un fichier
- index (Page 539) : page d'introduction pour entrer dans les pages Web standard

Par exemple, si vous entrez "http://ww.xx.yy.zz/communication.html", le navigateur affichera la page sur la communication.

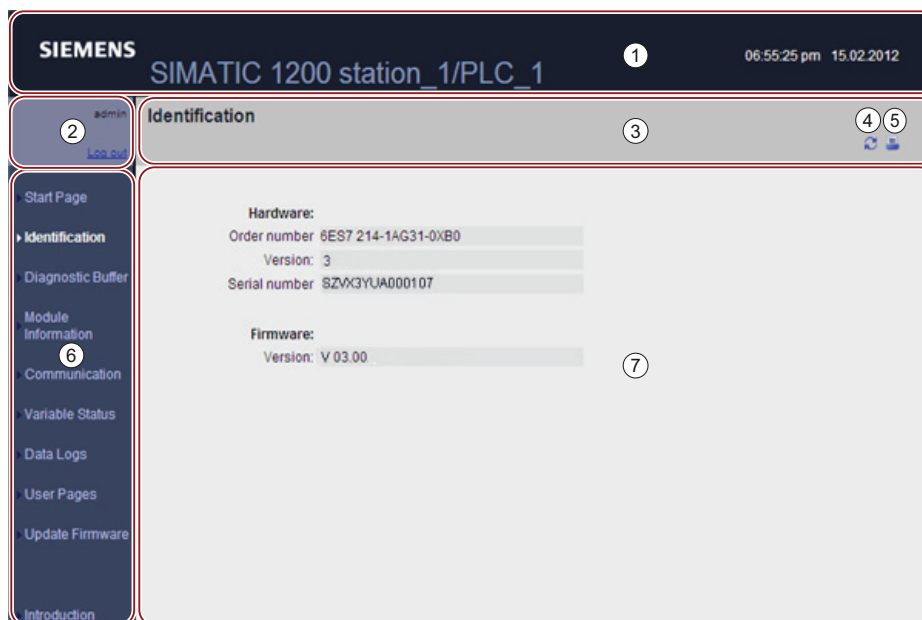
### Accès sécurisé

Vous pouvez utiliser https:// au lieu de http:// pour établir un accès sécurisé aux pages Web standard. Lorsque vous vous connectez au S7-1200 avec https://, le site Web crypte la session avec un certificat numérique. Les données sont transmises de manière sécurisée et ne sont accessibles à personne d'autre. Vous recevrez typiquement un avertissement de sécurité que vous pouvez acquitter avec "Oui" pour atteindre les pages Web standard. Pour éviter cet avertissement de sécurité à chaque accès sécurisé, vous pouvez importer le certificat de sécurité Siemens dans votre navigateur Web (Page 596).



## 11.2.2 Disposition des pages Web standard

Les pages Web standard ont une disposition commune avec des liens de navigation et des contrôles de page comme illustré ci-dessous :



- ① En-tête du serveur Web
- ② Ouverture ou fermeture de session
- ③ En-tête de page Web standard avec le nom de la page que vous regardez. Cet exemple montre la page d'identification de la CPU. Certaines pages Web standard, telles que les informations sur les modules, affichent également ici un chemin de navigation si plusieurs écrans de ce type sont accessibles.
- ④ Bouton d'actualisation : pour les pages à actualisation automatique, il active ou désactive la fonction d'actualisation automatique ; pour les pages sans actualisation automatique, il provoque l'actualisation avec des données actuelles.
- ⑤ Bouton d'impression : prépare et affiche une version imprimable des informations disponibles dans la page affichée.
- ⑥ Zone de navigation pour passer à une autre page
- ⑦ Zone de contenu de la page Web standard que vous consultez. Cet exemple montre la page d'identification de la CPU.

### Remarque

#### Impression de pages Web standard

Sachez, lors de l'impression du contenu de pages Web standard, que le contenu imprimé peut parfois différer de la page affichée. Ainsi, une copie imprimée de la page de la mémoire tampon de diagnostic peut contenir de nouvelles entrées de diagnostic qui ne figurent pas dans l'affichage de la page de la mémoire tampon de diagnostic. Si l'actualisation automatique n'est pas activée, l'affichage de la page montre les événements de diagnostic existant au moment où la page a été affichée la première fois et la copie papier contient les événements de diagnostic présents au moment où la fonction d'impression a été exécutée.

## Ouverture de session

Aucune ouverture de session n'est exigée pour afficher les données dans les pages Web standard. Mais vous devez ouvrir une session en tant qu'administrateur (utilisateur "admin") pour pouvoir exécuter certaines actions, comme changer l'état de fonctionnement de l'automate, écrire des valeurs en mémoire ou mettre à jour le firmware de la CPU.



The image shows a small rectangular login form with a light blue background. It contains two input fields: the first is labeled 'Name' and the second is labeled 'Password'. Below these fields is a blue button with the text 'Log in' in white.

Le cadre d'ouverture de session se situe près du coin supérieur gauche de chaque page.

Procédez comme suit pour ouvrir une session en tant qu'administrateur :

1. Entrez "admin" dans le champ Nom.
2. Dans le champ Mot de passe, entrez le mot de passe de la CPU s'il a été défini ; sinon, appuyez sur la touche Entrée.

Vous avez maintenant ouvert une session en tant qu'utilisateur "admin".

### ATTENTION

L'accès de personnes non autorisées à la CPU ou l'affectation de valeurs invalides aux variables API peut perturber le fonctionnement du processus, ce qui peut entraîner la mort, des blessures graves et des dommages matériels importants.

Comme la validation du serveur Web permet à des utilisateurs ayant les droits "admin" de changer l'état de fonctionnement, d'écrire dans des données de l'API et d'actualiser le firmware, Siemens vous recommande de mettre en oeuvre la politique de sécurité suivante :

- N'autorisez l'accès au serveur Web qu'avec le protocole HTTPS.
- Protégez la CPU (Page 175) avec un mot de passe fort. Les mots de passe forts contiennent au moins huit caractères, lettres, chiffres et caractères spéciaux mélangés, ne sont pas des mots du dictionnaire et ne sont pas des noms ou identifiants pouvant être déduits de vos données personnelles. Ne divulguez pas le mot de passe et changez-en fréquemment.
- Effectuez des recherches d'erreur et des vérifications de plage sur vos variables dans la logique de programme, car les utilisateurs des pages Web ont la possibilité de donner des valeurs invalides aux variables API.

Si des erreurs se produisent à l'ouverture de session, retournez à la page Introduction (Page 539) et téléchargez le certificat de sécurité Siemens (Page 596). Vous pourrez alors ouvrir une session sans rencontrer d'erreurs.

## Fermeture de session



Pour fermer la session de l'utilisateur "admin", cliquez sur le lien de fermeture de session "Log out" dans n'importe quelle page.

Vous pouvez continuer à afficher les pages Web standard sans session ouverte, mais vous ne pouvez pas exécuter les actions réservées à l'utilisateur "admin". Chacune des descriptions de page Web standard définit, le cas échéant, les actions nécessitant une ouverture de session en tant qu'administrateur.

### 11.2.3 Introduction

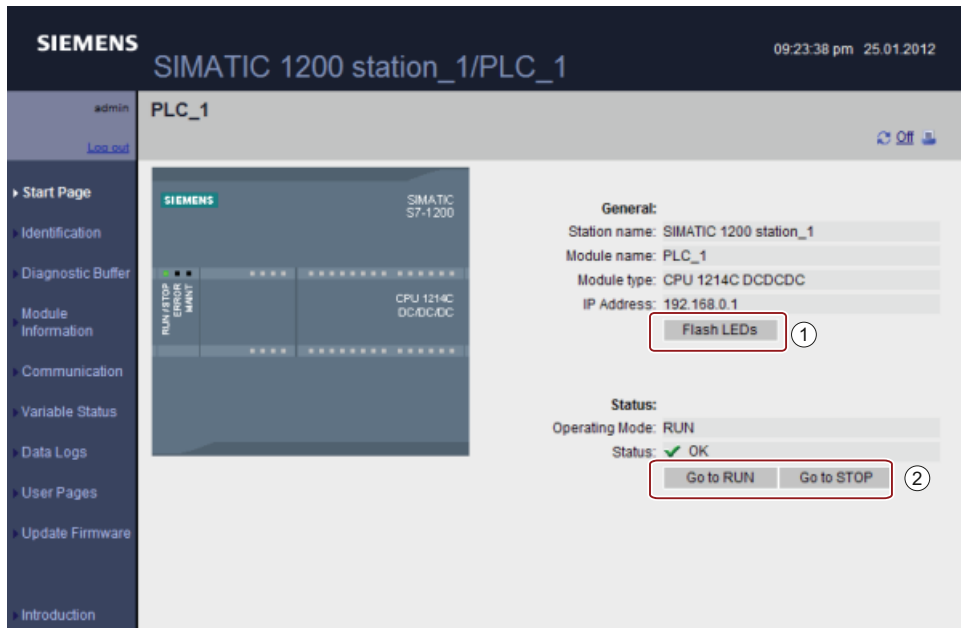
La page Introduction est l'écran d'accueil pour entrer dans les pages Web standard du S7-1200.



Dans cette page, cliquez sur "Entrer" pour accéder aux pages Web standard du S7-1200. En haut de l'écran se trouvent des liens vers des sites Web utiles de Siemens, ainsi qu'un lien pour télécharger le certificat de sécurité Siemens (Page 596).

### 11.2.4 Page de démarrage

La page de démarrage (Start Page) affiche une représentation de la CPU à laquelle vous êtes connecté et liste les informations générales concernant cette CPU. Si vous ouvrez une session en tant qu'utilisateur "admin", vous pouvez y changer l'état de fonctionnement de la CPU et faire clignoter les DEL.

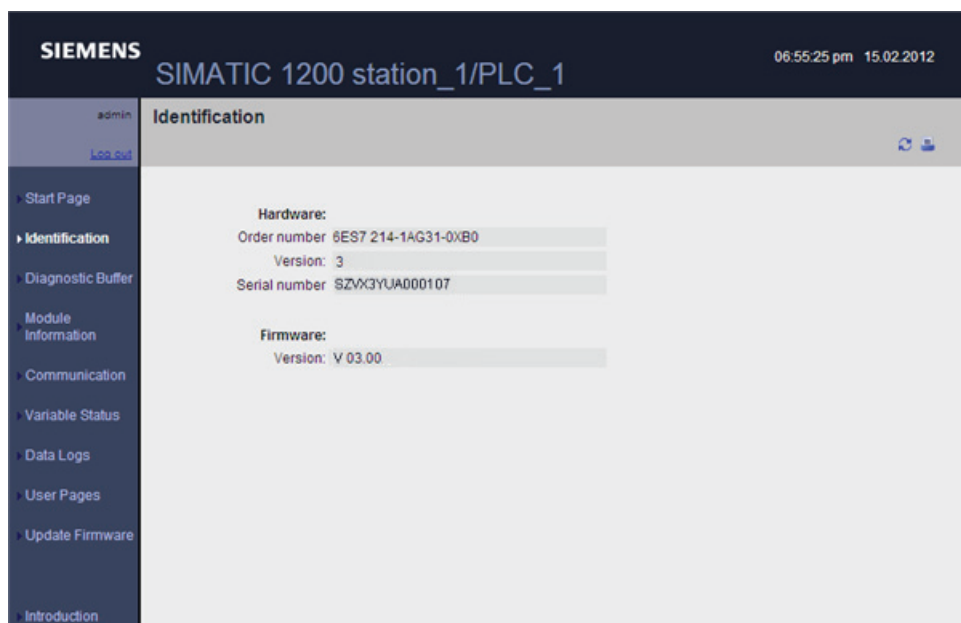


① et ② Les boutons pour faire clignoter les DEL et changer l'état de fonctionnement n'apparaissent sur la page de démarrage que si vous avez ouvert une session en tant qu'utilisateur "admin".

### 11.2.5 Identification

La page d'identification (Identification ) affiche des caractéristiques identifiant la CPU :

- Numéro de série
- Numéros de référence
- Information de version

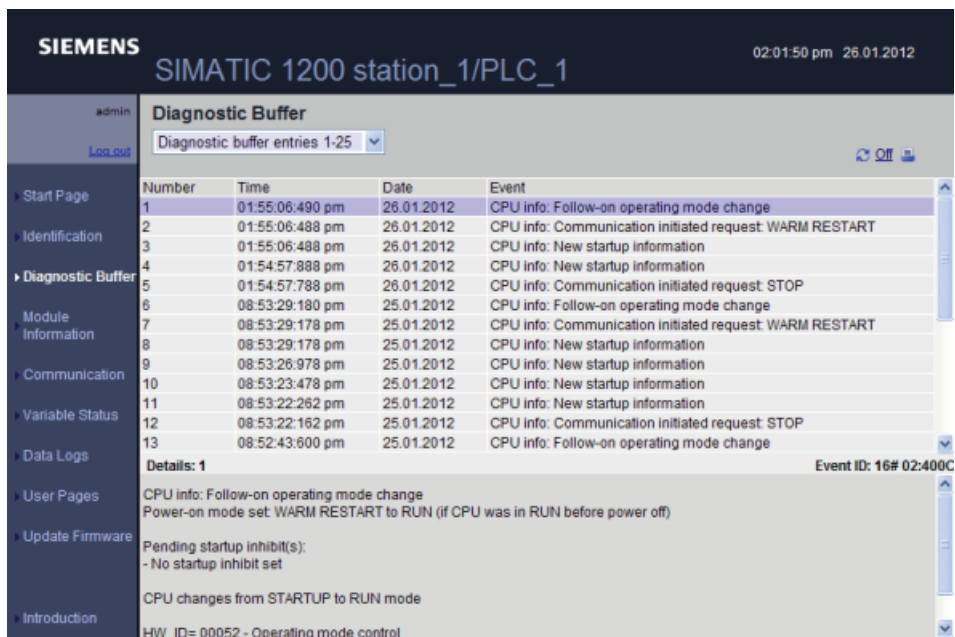


La page d'identification ne change pas en cas d'ouverture de session avec "admin".

### 11.2.6 Mémoire tampon de diagnostic

La page de mémoire tampon de diagnostic (Diagnostic Buffer) affiche des événements de diagnostic. Vous pouvez choisir dans le sélecteur la plage d'entrées de mémoire tampon de diagnostic à afficher, soit 1 à 25 soit 26 à 50. La partie supérieure de la page affiche les entrées correspondantes avec l'horodatage CPU auquel l'événement s'est produit. Les heures sont des heures système provenant de l'horloge temps réel (Page 93) de la CPU.

Dans la partie supérieure de la page, vous pouvez sélectionner n'importe quelle entrée individuelle pour afficher des informations détaillées sur cette entrée dans la partie inférieure de la page.



La page de mémoire tampon de diagnostic ne change pas en cas d'ouverture de session avec "admin".

### 11.2.7 Informations sur les modules

La page d'information sur les modules (Module Information) fournit des informations sur tous les modules dans le châssis local. La partie supérieure de l'écran affiche un récapitulatif des modules et la partie inférieure montre l'état et l'identification du module sélectionné.

### Affichage de l'état

**SIEMENS** SIMATIC 1200 station\_1/PLC\_1 03:49:55 pm 26.01.2012

admin **Module information** Slot  Filter

Log out SIMATIC 1200 station\_1 - SIMATIC 1200 station\_1

Slot	Status	Name	Order number	I address	Q address	Comment
1	✓	<a href="#">PLC_1</a> <a href="#">Details</a>	6ES7 214-1AG31-0XB0	---	---	
2	✓	<a href="#">DQ16 x 24VDC_1</a> <a href="#">Details</a>	6ES7 222-1BH30-0XB0	---	8	
3	✓	<a href="#">DI16/DQ16 x 24VDC_1</a> <a href="#">Details</a>	6ES7 223-1BL30-0XB0	12	12	

Start Page Identification Diagnostic Buffer **Module information** Communication Variable Status Data Logs User Pages Update Firmware Introduction

**Status** Identification

The module is parameterized, and it exchanges data with the CPU. The module did not detect any parameter assignment errors, channel errors, or hardware related errors.

### Affichage de l'identification

**SIEMENS** SIMATIC 1200 station\_1/PLC\_1 03:51:25 pm 26.01.2012

admin **Module information** Slot  Filter

Log out SIMATIC 1200 station\_1 - SIMATIC 1200 station\_1

Slot	Status	Name	Order number	I address	Q address	Comment
1	✓	<a href="#">PLC_1</a> <a href="#">Details</a>	6ES7 214-1AG31-0XB0	---	---	
2	✓	<a href="#">DQ16 x 24VDC_1</a> <a href="#">Details</a>	6ES7 222-1BH30-0XB0	---	8	
3	✓	<a href="#">DI16/DQ16 x 24VDC_1</a> <a href="#">Details</a>	6ES7 223-1BL30-0XB0	12	12	

Start Page Identification Diagnostic Buffer **Module information** Communication Variable Status Data Logs User Pages Update Firmware Introduction

**Status** Identification

Hardware Version: 1  
 Firmware Version: R.3.0.0  
 Serial Number: SZVX3YUA000107

### Zoom avant

Vous pouvez sélectionner un lien dans la partie supérieure pour faire un zoom avant sur les informations concernant ce module particulier. Les modules munis de sous-modules ont des liens pour chaque sous-module. Le type d'informations affichées varie selon le module sélectionné. Ainsi, la page d'information sur les modules affiche initialement le nom de la station SIMATIC 1200, un indicateur d'état et un commentaire. Si vous faites un zoom avant sur la CPU, les informations de module affichent le nom des entrées et sorties TOR et analogiques fournies par ce modèle de CPU (par exemple, "DI14/DO10", "AI2"), des informations d'adressage pour les E/S, des indicateurs d'état, des numéros d'emplacement et des commentaires.

Slot	Status	Name	Order number	I address	Q address	Comment
1.16	✓	HSC_1	<a href="#">Details</a> 6ES7 214-1AG31-0XB0	1000	---	
1.17	✓	HSC_2	<a href="#">Details</a> 6ES7 214-1AG31-0XB0	1004	---	
1.18	✓	HSC_3	<a href="#">Details</a> 6ES7 214-1AG31-0XB0	1008	---	
1.19	✓	HSC_4	<a href="#">Details</a> 6ES7 214-1AG31-0XB0	1012	---	
1.20	✓	HSC_5	<a href="#">Details</a> 6ES7 214-1AG31-0XB0	1016	---	
1.21	✓	HSC_6	<a href="#">Details</a> 6ES7 214-1AG31-0XB0	1020	---	
1.2	✓	AI2_1	<a href="#">Details</a> 6ES7 214-1AG31-0XB0	64	---	
1.1	✓	DI14/DO10_1	<a href="#">Details</a> 6ES7 214-1AG31-0XB0	0	0	
1.32	✓	Pulse_1	<a href="#">Details</a> 6ES7 214-1AG31-0XB0	---	1000	
1.33	✓	Pulse_2	<a href="#">Details</a> 6ES7 214-1AG31-0XB0	---	1002	

La page d'information sur les modules montre le chemin que vous avez suivi au fur et à mesure que vous zoomez. Vous pouvez cliquer sur n'importe quel lien dans ce chemin pour revenir à un niveau supérieur.

**Module information**  
[SIMATIC 1200 station 1 - SIMATIC 1200 station 1 - PLC\\_1](#)

### Tri des champs

Lorsque la liste affiche plusieurs modules, vous pouvez cliquer sur l'entête de colonne d'un champ pour effectuer un tri croissant ou décroissant selon ce champ.

Name ▼	
AI2_1	<a href="#">Details</a>
DI14/DO10_1	<a href="#">Details</a>
HSC_1	<a href="#">Details</a>
HSC_2	<a href="#">Details</a>
HSC_3	<a href="#">Details</a>
HSC_4	<a href="#">Details</a>
HSC_5	<a href="#">Details</a>
HSC_6	<a href="#">Details</a>
Pulse_1	<a href="#">Details</a>
Pulse_2	<a href="#">Details</a>

### Filtrage des informations sur les modules

Vous pouvez filtrer n'importe quel champ dans la liste d'informations sur les modules. Dans la liste déroulante, sélectionnez le nom du champ pour lequel vous voulez filtrer les données. Entrez un texte dans la boîte de texte associée et cliquez sur le lien Filtre. La liste est actualisée et vous montre les modules correspondant à votre critère de filtrage.

### Informations d'état

L'onglet Etat (Status) dans la partie inférieure de la page d'informations sur les modules affiche une description de l'état en cours du module sélectionné dans la partie supérieure.



## Identification

L'onglet Identification affiche le numéro de série et les numéros de version du module sélectionné.

La page d'information sur les modules ne change pas en cas d'ouverture de session avec "admin".

## 11.2.8 Communication

La page Communication affiche les paramètres de la CPU connectée, ainsi que des statistiques de communication. L'onglet Paramètres (Parameter) montre l'adresse MAC de la CPU, l'adresse et les réglages IP de la CPU, ainsi que ses propriétés physiques. L'onglet Statistiques (Statistics) montre les statistiques d'émission et de réception.

### Communication : affichage des paramètres



Communication : affichage des statistiques

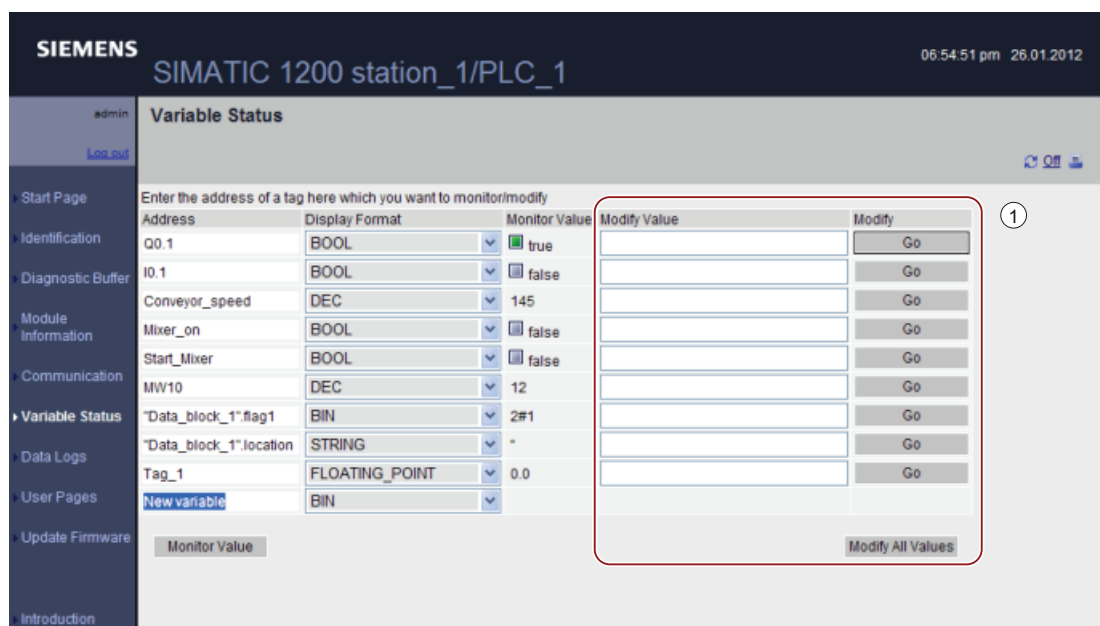


La page de communication ne change pas en cas d'ouverture de session avec "admin".

11.2.9 Etat des variables

La page Etat des variables (Variable Status) vous permet de visualiser n'importe quelle donnée d' E/S ou de mémoire dans votre CPU. Vous pouvez entrer une adresse directe (comme I0.0), un nom de variable API ou une variable d'un bloc de données spécifique. Pour les variables de bloc de données, vous devez mettre le nom du bloc de données entre guillemets. Vous pouvez sélectionner un format d'affichage des données pour chaque valeur à visualiser. Vous pouvez continuer à entrer et à spécifier toutes les valeurs que vous voulez dans les limites de la page. Les valeurs à visualiser s'affichent automatiquement et sont actualisées par défaut à moins que vous ne cliquiez sur le bouton "Off" en haut à droite de la page. Lorsque l'actualisation est désactivée, vous pouvez cliquer sur "On" pour réactiver l'actualisation automatique.

Si vous avez ouvert une session avec "admin", vous pouvez également forcer les valeurs de données. Entrez les valeurs de forçage dans le champ "Modify Value" (Forcer valeur) approprié. Cliquez sur le bouton "Go" à côté d'une valeur pour écrire cette valeur dans la CPU. Vous pouvez également entrer plusieurs valeurs et cliquer sur "Modify All Values" (Forcer toutes les valeurs) pour écrire toutes les valeurs dans la CPU.



① La fonctionnalité "Forcer valeur" n'est visible et accessible que si vous avez ouvert une session en tant qu'utilisateur "admin".

La page d'état des variables ne conserve pas vos entrées lorsque vous y revenez après l'avoir quittée. Vous pouvez marquer la page d'un signet et revenir au signet pour voir les mêmes entrées. Si vous ne marquez pas la page d'un signet, vous devez ressaisir les variables.

### Remarque

Tenez compte des particularités suivantes lors de l'utilisation de la page d'état des variables :

- La page d'état des variables ne vous permet pas de modifier une chaîne dépassant 198 caractères.
- Si la notation exponentielle est utilisée pour entrer une valeur pour un type de données Real ou LReal dans la page d'état des variables :
  - Pour entrer la valeur d'un réel (Real ou LReal) avec un exposant positif (tel +3.402823e+25), utilisez l'un des formats suivants :
    - +3.402823e25
    - +3.402823e+25
  - Pour entrer la valeur d'un réel (Real ou LReal) avec un exposant négatif (tel +3.402823e-25), entrez la valeur comme suit :
    - +3.402823e-25
  - Veillez à ce que la mantisse de la valeur réelle en notation exponentielle comprenne un séparateur décimal. En l'absence de séparateur décimal, vous obtiendrez une valeur entière inattendue. Ainsi, entrez -1.0e8 et non -1e8.
- La page d'état des variables ne prend en charge que 15 chiffres pour une valeur LReal (quelle que soit la position du séparateur décimal). La saisie de plus de 15 chiffres entraîne une erreur d'arrondi.

Limitations concernant la page d'état des variables :

- Le nombre maximum d'entrées de variable par page est de 50.
- Le nombre maximum de caractères pour l'URL correspondant à la page d'état des variables est de 2083. Vous pouvez voir l'URL représentant la page d'état des variables en cours dans la barre d'adresse de votre navigateur.
- Pour le format d'affichage des caractères, la page affiche des valeurs hexadécimales si les valeurs en cours de la CPU ne sont pas des caractères ASCII valides tels qu'interprétés par le navigateur.

---

**Remarque**

Si un nom de variable comprend des caractères spéciaux de sorte qu'il est refusé comme entrée dans la page d'état des variables, vous pouvez l'écrire entre guillemets. Dans la plupart des cas, la page d'état des variables reconnaîtra alors le nom de variable.

---

### 11.2.10 Journaux

La page Journaux (Data Logs) vous permet de voir ou de télécharger un nombre donné d'entrées de journal. Si vous avez ouvert une session en tant qu'utilisateur "admin", vous pouvez également effacer ces entrées après les avoir téléchargées ou vous pouvez les supprimer. Le serveur Web télécharge les journaux dans votre PC en format CSV (valeurs séparées par des virgules).

La page Journaux affiche jusqu'à 40 fichiers journaux. S'il en existe plus de 40, la page Journaux affiche les 40 premiers fichiers journaux créés.

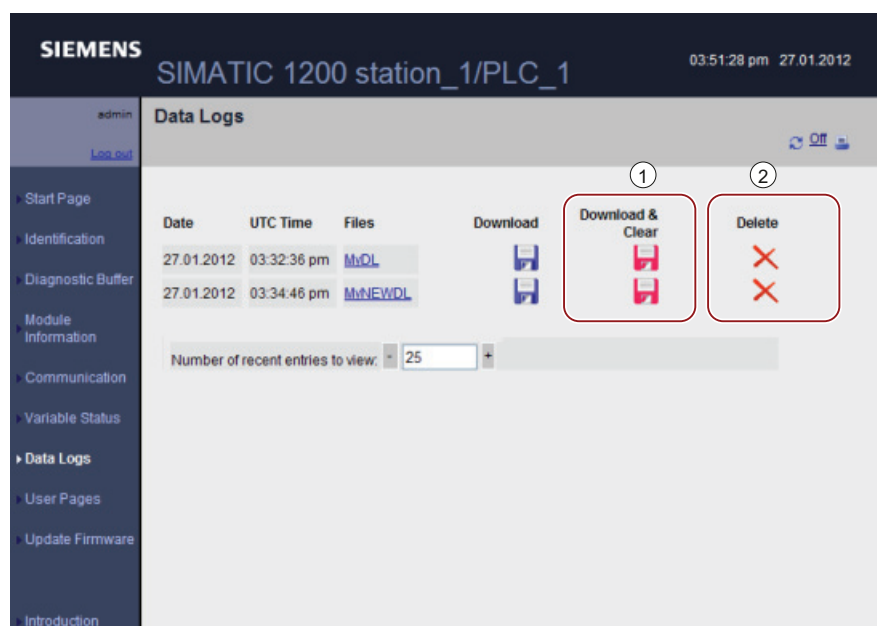
---

**Remarque**

**L'horodatage pour les journaux est indiqué en heure système et non en heure locale.**

La CPU écrit les horodatages pour les entrées de journaux en heure système (Page 93) et la page standard "Journaux" du serveur Web affiche les horodatages pour les journaux en heure système.

---



- ① L'option "Download & Clear" (télécharger et effacer) n'est pas disponible si vous n'avez pas ouvert de session en tant qu'utilisateur "admin".
- ② L'option "Delete" (supprimer) n'est pas disponible si vous n'avez pas ouvert de session en tant qu'utilisateur "admin".

### Remarque

Le fichier journal est en format CSV anglais/américain. Pour l'ouvrir dans Excel sur des systèmes non anglais/américains, vous devez l'y importer avec des paramètres spécifiques (Page 597).

### Entrées récentes : téléchargement d'un nombre donné d'entrées récentes

Définissez le nombre d'entrées récentes à télécharger, puis cliquez sur le nom du journal pour déclencher le téléchargement du nombre d'entrées indiqué. Dans le fichier .csv de sortie, les entrées sont classées par ordre d'entrée décroissant. Windows vous demandera si vous voulez ouvrir ou enregistrer le fichier journal.

Le nombre d'entrées récentes à visualiser est par défaut de 25 entrées. Vous pouvez modifier cette valeur dans le champ "Number of recent entries to view" en saisissant un nombre ou en utilisant le bouton + ou - pour incrémenter ou décrémenter la valeur.

### Remarque

Le paramètre "Records" de l'instruction DataLogCreate (Page 337) définit le nombre maximum d'entrées par fichier journal.

### Téléchargement d'un fichier journal contenant toutes les entrées

Pour télécharger un fichier journal entier, cliquez sur l'icône de téléchargement correspondant à un fichier journal spécifique. Windows vous demandera si vous voulez ouvrir ou enregistrer le fichier journal.

Le fichier .csv de sortie contient toutes les entrées classées par ordre d'entrée croissant, à moins que le journal ne soit plein et que d'anciennes entrées (de numéro plus petit) aient été écrasées par des entrées ultérieures (de numéro plus grand).

### Téléchargement et effacement d'un fichier journal

Vous devez avoir ouvert une session en tant qu'utilisateur "admin" pour télécharger un fichier journal puis effacer toutes les entrées. Vous pouvez alors cliquer sur l'icône "Download & Clear" correspondant à un fichier journal spécifique. Windows vous demandera si vous voulez ouvrir ou enregistrer le fichier journal.

Une fois le téléchargement achevé, une nouvelle ligne "//END" est insérée après l'entrée d'en-tête du fichier journal enregistré dans l'API. Cela efface de fait le journal pour les futurs traitements API internes mais, lors de téléchargements ultérieurs de ce fichier, de nouvelles entrées seront insérées au-dessus de la première ligne "//END".

---

#### Remarque

##### Marque "//END" du fichier journal .csv

La marque "//END" de fin de fichier .csv est uniquement utilisée pour les ((entrées max.) -1) premières entrées afin de repérer la fin logique du fichier. Le fichier peut contenir, après la fin logique, des données pouvant être interprétées comme des entrées supplémentaires par Excel. Vous devez rechercher le premier "//END" et l'effacer ainsi que toutes les entrées qui le suivent. S'il n'y a pas de marque de fin logique, vous pouvez classer les lignes de données à l'aide du numéro d'entrée.

---

### Suppression d'un fichier journal

Vous devez avoir ouvert une session en tant qu'utilisateur "admin" pour supprimer un fichier journal. Vous pouvez alors cliquer sur l'icône Delete correspondant à un fichier journal spécifique. Le serveur Web supprime alors le contenu du fichier.

### Informations supplémentaires

Reportez-vous au chapitre Consignation de données (Page 335) pour plus d'informations sur la programmation avec les instructions de journal.

### 11.2.11 Mise à jour du firmware

La page Mise à jour du firmware permet à un utilisateur "admin" de mettre à jour le firmware de la CPU à partir d'un fichier.

---

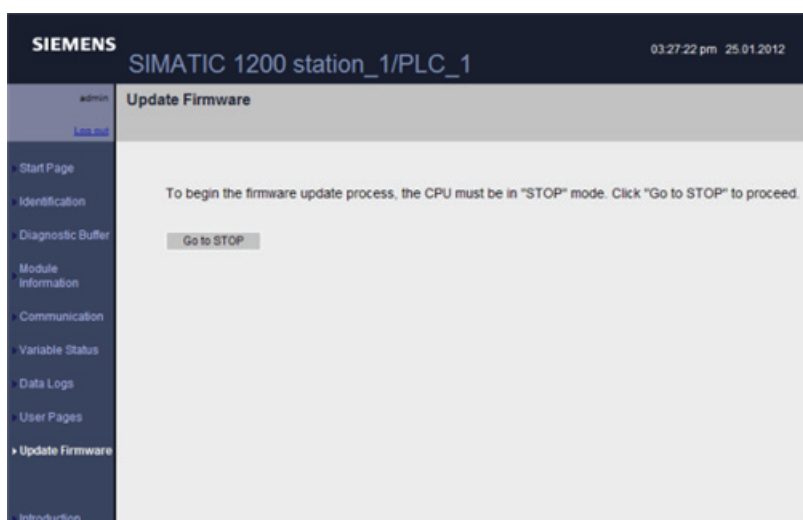
#### Remarque

La fonction Mise à jour du firmware ne permet de mettre à jour que des CPU S7-1200 de version 3.0 ou plus.

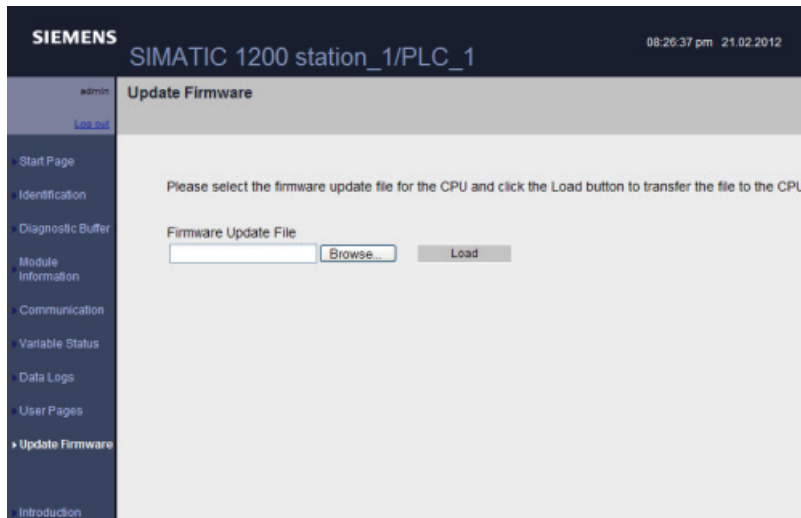
Le serveur Web utilise le protocole "https" pour effectuer l'actualisation du firmware.

---

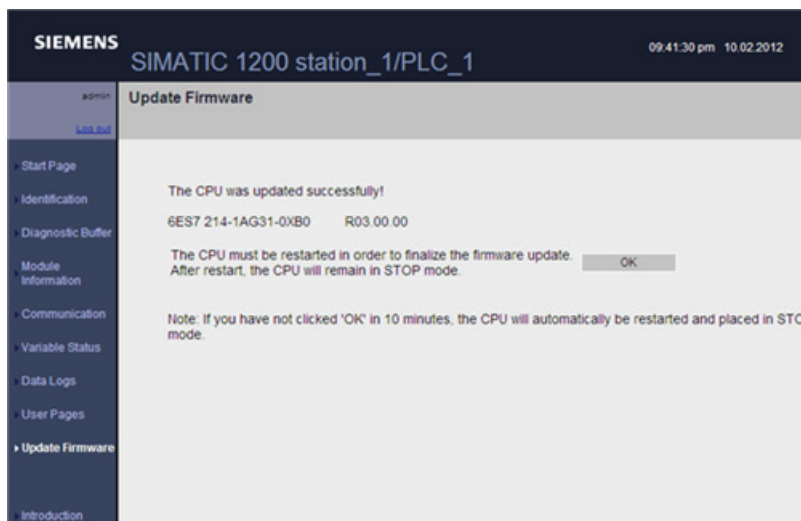
La CPU doit être à l'état ARRET lorsque vous actualisez le firmware. Si ce n'est pas le cas, le serveur Web vous demande de la faire passer à l'état ARRET.



Lorsque la CPU est à l'état ARRET, vous pouvez naviguer et sélectionner le fichier à partir duquel charger la mise à jour de la version du firmware. Les mises à jour du firmware sont disponibles sur le site Web d'assistance client (<http://support.automation.siemens.com>). Une fois la mise à jour de firmware appropriée téléchargée du site Web sur votre ordinateur, vous pouvez naviguer jusqu'à ce fichier sur votre ordinateur et le sélectionner pour effectuer la mise à jour.

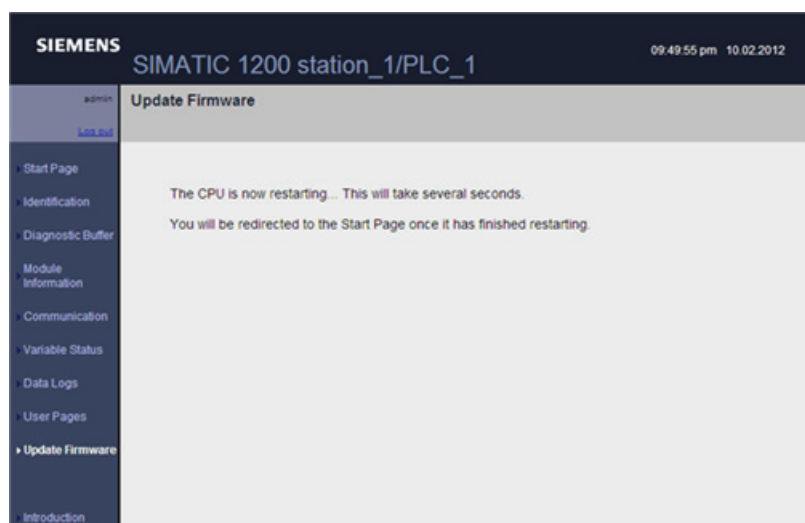


Pendant la mise à jour, la page Mise à jour du firmware affiche un message signalant que la mise à jour est en cours. Une fois la mise à jour achevée, la page Mise à jour du firmware affiche le numéro de référence et le numéro de version du firmware mis à jour et vous demande de redémarrer la CPU.





Si vous ne réagissez pas dans les dix minutes, la CPU redémarre automatiquement.

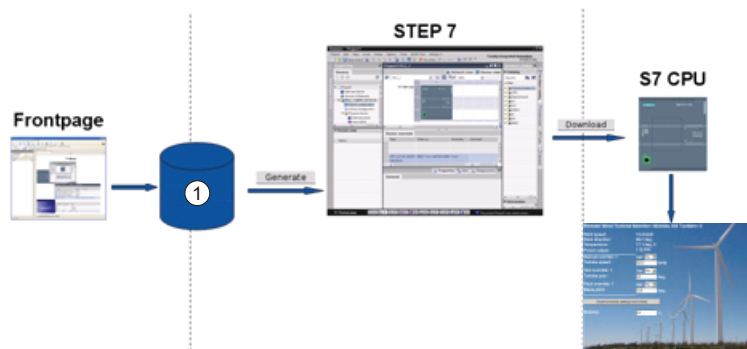


## 11.3 Pages Web personnalisées

Le serveur Web du S7-1200 vous offre la possibilité de créer vos propres pages HTML propres à une application qui incorporent les données provenant de l'API. Vous créez ces pages à l'aide de l'éditeur HTML de votre choix et vous les téléchargez dans la CPU où elles sont accessibles dans le menu des pages Web standard. Ce processus implique plusieurs tâches :

- Créer des pages HTML avec un éditeur HTML, par exemple Microsoft Frontpage (Page 554)
- Inclure des commandes AWP dans des commentaires HTML dans le code HTML (Page 555) : Les commandes AWP constituent un jeu fixe de commandes que Siemens fournit pour accéder aux informations de la CPU.
- Configurer STEP 7 pour lire et traiter les pages HTML (Page 569)
- Générer les blocs à partir des pages HTML (Page 569)
- Programmer STEP 7 pour commander l'utilisation des pages HTML (Page 571)
- Compiler et charger les blocs dans la CPU (Page 573)
- Accéder aux pages Web personnalisées depuis votre PC (Page 573)

Ce processus est illustré ci-dessous :



① Fichiers HTML avec commandes AWP intégrées

### 11.3.1 Création de pages HTML

Vous pouvez utiliser le progiciel de votre choix pour créer vos propres pages HTML à utiliser avec le serveur Web. Assurez-vous que le code HTML est conforme aux standards HTML du W3C (World Wide Web Consortium). STEP 7 ne procède à aucune vérification de votre syntaxe HTML.

Vous pouvez utiliser un progiciel qui permet la conception en mode WYSIWYG ou mise en page, mais vous devez pouvoir éditer votre code HTML en HTML pur. La plupart des outils de création Web fournissent ce type d'édition ; sinon, vous pouvez toujours utiliser un éditeur de texte simple pour éditer le code HTML. Insérez la ligne suivante dans votre page HTML pour définir le jeu de caractères de la page à UTF-8 :

```
<meta http-equiv="content-type" content="text/html; charset=utf-8">
```

Veillez également à enregistrer le fichier en codage de caractères UTF-8 depuis l'éditeur.

Vous utilisez STEP 7 pour compiler tout le contenu de vos pages HTML en blocs de données STEP 7. Ces blocs de données consistent en un bloc de données de commande qui pilote l'affichage des pages Web et en un ou plusieurs blocs de données de fragment qui contiennent les pages Web compilées. Soyez conscient que les jeux importants de pages HTML, en particulier celles avec de nombreuses images, nécessitent une quantité significative d'espace en mémoire de chargement (Page 573) pour les DB de fragment. Si la mémoire de chargement interne de votre CPU n'est pas suffisante pour vos pages Web personnalisées, utilisez une carte mémoire (Page 115) pour fournir une mémoire de chargement externe.

Pour que votre code HTML utilise des données dans le S7-1200, vous incluez des commandes AWP (Page 555) sous forme de commentaires HTML. Lorsque vous avez terminé, sauvegardez vos pages HTML sur votre PC et notez le chemin du dossier où vous les avez sauvegardées.

## Actualisation de pages Web personnalisées

Les pages Web personnalisées ne sont pas actualisées automatiquement. C'est à vous de choisir si votre programme HTML doit actualiser la page ou non. Le rafraîchissement périodique des pages qui affichent des données de l'API permet d'avoir des données toujours actuelles. Dans les pages HTML servant de formulaires pour la saisie de données, l'actualisation peut interférer avec la saisie des données par l'utilisateur. Si vous voulez que votre page entière soit actualisée automatiquement, vous pouvez ajouter cette ligne dans votre en-tête HTML, "10" étant le nombre de secondes entre les rafraîchissements :

```
<meta http-equiv="Refresh" content="10">
```

Vous pouvez aussi utiliser des JavaScripts ou d'autres techniques HTML pour gérer l'actualisation de la page ou des données. Consultez la documentation HTML et JavaScript à ce sujet.

### 11.3.2 Commandes AWP prises en charge par le serveur Web du S7-1200

Le serveur Web S7-1200 fournit des commandes AWP que vous intégrez dans vos pages Web personnalisées sous forme de commentaires HTML aux fins suivantes :

- Lecture de variables (Page 556)
- Ecriture de variables (Page 557)
- Lecture de variables spéciales (Page 559)
- Ecriture de variables spéciales (Page 561)
- Définition de types Enum (Page 563)
- Affectation de variables aux types Enum (Page 564)
- Création de blocs de données de fragment (Page 565)

#### Syntaxe générale

Mis à part la commande de lecture d'une variable, les commandes AWP ont la syntaxe suivante :

```
<!-- AWP_ <nom et paramètres de la commande> -->
```

Vous utilisez les commandes AWP conjointement avec des commandes de formulaire HTML typiques pour écrire des variables dans la CPU.

Les descriptions des commandes AWP dans les pages suivantes utilisent les conventions suivantes :

- Les éléments entre crochets [ ] sont facultatifs.
- Les éléments entre chevrons < > sont des valeurs de paramètres à préciser.
- Les apostrophes et guillemets font partie intégrante de la commande. Ils doivent être présents comme indiqués.
- Les caractères spéciaux dans les noms de variables ou de blocs de données doivent être précédés d'un caractère d'échappement ou être entourés d'apostrophes ou de guillemets (Page 567) selon l'usage.

Utilisez un éditeur de texte ou le mode d'édition HTML pour insérer des commandes AWP dans vos pages.

## Récapitulatif des commandes AWP

Les détails d'utilisation de chaque commande AWP sont précisés dans les paragraphes suivants, mais voici un bref récapitulatif des commandes :

### Lecture de variables

```
:=<NomVar>:
```

### Écriture de variables

```
<!-- AWP_In_Variable Name='<NomVar1>' [Use='<NomVar2>'] ... -->
```

Cette commande AWP ne fait que déclarer que la variable dans la clause Name est inscriptible. C'est votre code HTML qui exécute des écritures dans la variable en indiquant son nom dans des instructions HTML <input>, <select> ou autres à l'intérieur d'un formulaire HTML.

### Lecture de variables spéciales

```
<!-- AWP_Out_Variable Name='<Type>:<Nom>' [Use='<NomVar>'] -->
```

### Écriture de variables spéciales

```
<!-- AWP_In_Variable Name='<Type>:<Nom>' [Use='<NomVar>'] -->
```

### Définition de types Enum

```
<!-- AWP_Enum_Def Name='<nom du type  
Enum>' Values='<valeur>, <valeur>, ... ' -->
```

### Référencement de types Enum

```
<!-- AWP_Enum_Ref Name='<NomVar>' Enum='<TypeEnum>' -->
```

### Création de fragments

```
<!-- AWP_Start_Fragment Name='<Nom>' [Type=<Type>] [ID=<id>] -->
```

### Importation de fragments

```
<!-- AWP_Import_Fragment Name='<Nom>' -->
```

### 11.3.2.1 Lecture de variables

Les pages Web personnalisées peuvent lire des variables (API) dans la CPU.

#### Syntaxe

```
:=<NomVar>:
```

## Paramètres

<NomVar>	Variable à lire qui peut être un nom de variable API dans votre programme STEP 7, une variable de bloc de données, une E/S ou un memento adressable. N'utilisez pas de guillemets autour du nom de variable pour les mémentos, les adresses d'E/S et les noms d'alias (Page 567). Pour les variables API, utilisez des guillemets de part et d'autre du nom de variable. Pour les variables de bloc de données, vous devez mettre uniquement le nom du bloc entre guillemets. Le nom de la variable se trouve en dehors des guillemets. Notez que vous indiquez le nom du bloc et non son numéro.
----------	---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

## Exemples

```
:= "Vitesse_convoyeur" := "Mon_bloc_données".memento1:
:= I0.0:
:= MW100:
```

### Exemple de lecture d'une variable avec alias

```
<!-- AWP_Out_Variable Name='memento1'
Use='Mon_bloc_données'.memento1' -->
:= memento1:
```

---

#### Remarque

La définition d'alias pour les variables API et les variables de blocs de données est décrite à la rubrique Utilisation d'un alias pour une référence de variable (Page 562).

---

Si un nom de variable ou de bloc de données comprend des caractères spéciaux, vous devez en plus utiliser des apostrophes ou guillemets ou des caractères d'échappement comme décrit à la rubrique Gestion des noms de variables contenant des caractères spéciaux (Page 567).

### 11.3.2.2 Ecriture de variables

Les pages personnalisées peuvent écrire des données dans la CPU. Pour ce faire, vous utilisez une commande AWP afin d'identifier une variable dans la CPU dans laquelle il est possible d'écrire depuis la page HTML. Il faut indiquer la variable par le nom de variable API ou par le nom de variable de bloc de données. Vous pouvez déclarer plusieurs noms de variables dans une instruction. Utilisez les commandes HTTP POST standard pour écrire les données dans la CPU.

Un usage typique est de concevoir un formulaire dans votre page HTML avec des champs de saisie de texte ou des choix de liste de sélection qui correspondent aux variables CPU inscriptibles. Comme avec toutes les pages personnalisées, vous générez alors les blocs dans STEP 7 de manière à les inclure dans votre programme STEP 7. Lorsqu'un utilisateur "admin" accède par la suite à cette page et entre des données dans les champs d'entrée ou sélectionne un choix dans une liste de sélection, le serveur Web convertit l'entrée au type de données approprié pour la variable et écrit la valeur dans la variable dans la CPU. Notez que la clause de nom pour les champs d'entrée HTML et les listes de sélection HTML utilise une syntaxe typique de la clause de nom de la commande AWP\_In\_Variable. Typiquement, vous écrivez le nom entre apostrophes et, si vous référencez un bloc de données, vous écrivez le nom du bloc de données entre guillemets.

Reportez-vous à la documentation HTML pour plus de détails sur la gestion des formulaires.

### Syntaxe

```
<!-- AWP_In_Variable Name='<Varname1>' [Use='<Varname2>'] ... -->
```

### Paramètres

<NomVar1>	<p>Si aucune clause d'utilisation (Use) n'est fournie, "NomVar1" est la variable dans laquelle il faut écrire. Il peut s'agir d'un nom de variable API dans votre programme STEP 7 ou d'une variable d'un bloc de données spécifique.</p> <p>En présence d'une clause Use, NomVar1 est un nom de remplacement pour la variable référencée dans &lt;NomVar2&gt; (Page 562). Il s'agit d'un nom local à l'intérieur de la page HTML.</p>
<NomVar2>	<p>En présence d'une clause Use, "NomVar2" est la variable dans laquelle il faut écrire. Il peut s'agir d'un nom de variable API dans votre programme STEP 7 ou d'une variable d'un bloc de données spécifique.</p>

Pour les clauses Name et Use, le nom complet doit être entre apostrophes. A l'intérieur des apostrophes, utilisez des guillemets autour d'un nom de variable API et d'un nom de bloc de données. Le nom du bloc de données est entre guillemets mais pas le nom de la variable de bloc de données. Notez que, pour les variables de blocs de données, vous utilisez le nom du bloc et non son numéro.

### Exemples d'utilisation d'un champ de saisie HTML

```
<!-- AWP_In_Variable Name=' "Target_Level" ' -->
<form method="post">
<p>Input Target Level: <input name=' "Target_Level" ' type="text" />
</p>
</form>

<!-- AWP_In_Variable Name=' "Data_block_1".Braking' -->
<form method="post">
<p>Braking: <input name=' "Data_block_1".Braking' type="text" />
%</p>
</form>

<!-- AWP_In_Variable Name=' "Braking" ' Use=' "Data_block_1".Braking' -
->
```

```
<form method="post">
<p>Braking: <input name="Braking" type="text" /> %</p>
</form>
```

### Exemple d'utilisation d'une liste de sélection HTML

```
<!-- AWP_In_Variable Name="Data_block_1".ManualOverrideEnable'-->
<form method="post">
<select name="Data_block_1".ManualOverrideEnable'>
<option value="Data_block_1".ManualOverrideEnable:> </option>
<option value=1>Yes</option>
<option value=0>No</option>
</select><input type="submit" value="Submit setting" /></form>
```

---

#### Remarque

Seul un utilisateur "admin" peut écrire des données dans la CPU. Les commandes ne sont pas prises en compte si l'utilisateur n'a pas ouvert de session en tant qu'administrateur.

---

Si un nom de variable ou de bloc de données comprend des caractères spéciaux, vous devez en plus utiliser des apostrophes ou guillemets ou des caractères d'échappement comme décrit à la rubrique "Gestion des noms de variables contenant des caractères spéciaux (Page 567)".

### 11.3.2.3 Lecture de variables spéciales

Le serveur Web offre la possibilité de lire des valeurs dans l'API afin de les stocker dans des variables spéciales dans l'en-tête de réponse HTTP. Vous pourriez, par exemple, vouloir lire un nom de chemin dans une variable API pour réorienter l'URL vers un autre emplacement à l'aide de la variable spéciale HEADER:Location.

#### Syntaxe

```
<!-- AWP_Out_Variable Name='<Type>:<Name>' [Use='<Varname>'] -->
```

## Paramètres

<Type>	Type de la variable spéciale. Ce peut être : HEADER COOKIE_VALUE COOKIE_EXPIRES
<Nom>	Vous trouverez dans la documentation HTTP la liste de tous les noms de variables HEADER. En voici quelques exemples : Status : code de réponse Location : chemin pour un réacheminement Retry-After : durée pendant laquelle le service risque d'être indisponible pour le client demandeur Pour les types COOKIE_VALUE et COOKIE_EXPIRES, <Nom> est le nom du cookie spécifique. COOKIE_VALUE:nom : valeur du cookie nommé COOKIE_EXPIRES:nom : durée d'expiration en secondes du cookie nommé La clause Name doit être écrite entre apostrophes ou entre guillemets. Si aucune clause Use n'est indiquée, le nom de variable spéciale correspond à un nom de variable API. Ecrivez la clause Name entière entre apostrophes et la variable API entre guillemets. Le nom de variable spéciale et le nom de variable API doivent correspondre exactement.
<NomVar>	Nom de la variable API ou de bloc de données pour la variable dans laquelle lire NomVar doit être écrit entre apostrophes. A l'intérieur des apostrophes, utilisez des guillemets autour d'un nom de variable API ou d'un nom de bloc de données. Le nom du bloc de données est entre guillemets mais pas le nom de la variable de bloc de données. Notez que, pour les variables de blocs de données, vous utilisez le nom du bloc et non son numéro.

## Exemples

```
<!-- AWP_Out_Variable Name="HEADER:Status" -->
```

Dans cet exemple, la variable spéciale HTTP "HEADER:Status" reçoit la valeur de la variable API "HEADER:Status". Le nom dans la table de variables API doit correspondre exactement au nom de la variable spéciale si aucune clause Use n'est indiquée.

```
<!-- AWP_Out_Variable Name='HEADER:Status' Use="Etat" -->
```

Dans cet exemple, la variable spéciale HTTP "HEADER:Status" reçoit la valeur de la variable API "Etat".

Si un nom de variable ou de bloc de données comprend des caractères spéciaux, vous devez en plus utiliser des apostrophes ou guillemets ou des caractères d'échappement comme décrit à la rubrique Gestion des noms de variables contenant des caractères spéciaux (Page 567).



### 11.3.2.4 Ecriture de variables spéciales

Le serveur Web offre la possibilité d'écrire dans la CPU des valeurs provenant de variables spéciales dans l'en-tête de requête HTTP. Vous pouvez, par exemple, stocker dans STEP 7 des informations sur le cookie associé à une page Web personnalisée, sur l'utilisateur qui accède à une page ou sur des informations d'en-tête. Le serveur Web permet d'accéder à des variables spéciales spécifiques que vous pouvez écrire dans la CPU si vous avez ouvert une session en tant qu'administrateur (admin).

#### Syntaxe

```
<!-- AWP_In_Variable Name='<Type>:<Nom>' [Use='<NomVar>'] -->
```

#### Paramètres

<Type>	Type de la variable spéciale. Ce peut être : HEADER SERVER COOKIE_VALUE
<Nom>	Variable spécifique dans les types définis ci-dessus, comme illustré dans ces exemples : HEADER:Accept : types de contenu qui sont acceptables HEADER:User-Agent : informations sur l'agent utilisateur à l'origine de la requête SERVER:current_user_id : ID de l'utilisateur en cours ; 0 si aucun utilisateur n'a ouvert de session SERVER:current_user_name : nom de l'utilisateur en cours COOKIE_VALUE:<name> : valeur du cookie nommé Ecrivez la clause Name entre apostrophes. Si aucune clause Use n'est indiquée, le nom de variable spéciale correspond à un nom de variable API. Ecrivez la clause Name entière entre apostrophes et la variable API entre guillemets. Le nom de variable spéciale et le nom de variable API doivent correspondre exactement. Vous trouverez dans la documentation HTTP la liste de tous les noms de variables HEADER.
<NomVar>	Nom de la variable dans votre programme STEP 7 dans laquelle vous voulez écrire la variable spéciale. Ce peut être un nom de variable API ou une variable de bloc de données. NomVar doit être écrit entre apostrophes. A l'intérieur des apostrophes, utilisez des guillemets autour d'un nom de variable API ou d'un nom de bloc de données. Le nom du bloc de données est entre guillemets mais pas le nom de la variable de bloc de données. Notez que, pour les variables de blocs de données, vous utilisez le nom du bloc et non son numéro.

#### Exemples

```
<!-- AWP_In_Variable Name='"SERVER:current_user_id"' -->
```

Dans cet exemple, la page Web écrit la valeur de la variable spéciale HTTP "SERVER:current\_user\_id" dans la variable API de nom "SERVER:current\_user\_id".

```
<!-- AWP_In_Variable Name=SERVER:current_user_id'
Use='mon_IDutilisateur' -->
```

Dans cet exemple, la page Web écrit la valeur de la variable spéciale HTTP "SERVER:current\_user\_id" dans la variable API de nom "mon\_IDutilisateur".

---

**Remarque**

Seul un utilisateur "admin" peut écrire des données dans la CPU. Les commandes ne sont pas prises en compte si l'utilisateur n'a pas ouvert de session en tant qu'administrateur.

---

Si un nom de variable ou de bloc de données comprend des caractères spéciaux, vous devez en plus utiliser des apostrophes ou guillemets ou des caractères d'échappement comme décrit à la rubrique "Gestion des noms de variables contenant des caractères spéciaux (Page 567)".

**11.3.2.5 Utilisation d'un alias pour une référence de variable**

Vous pouvez utiliser dans votre page Web personnalisée un alias pour une variable In\_Variable ou Out\_Variable. Par exemple, vous pouvez utiliser dans votre page HTML un nom symbolique différent de celui utilisé dans la CPU ou établir une correspondance entre une variable dans la CPU et une variable spéciale. La clause AWP Use offre cette possibilité.

**Syntaxe**

```
<-- AWP_In_Variable Name='<NomVar1>' Use='<NomVar2>' -->
<-- AWP_Out_Variable Name='<NomVar1>' Use='<NomVar2>' -->
```

**Paramètres**

<NomVar1>	Nom de l'alias ou de la variable spéciale NomVar1 doit être écrit entre apostrophes ou entre guillemets.
<NomVar2>	Nom de la variable API à laquelle vous voulez affecter un alias. Il peut s'agir d'une variable API, d'une variable de bloc de données ou d'une variable spéciale. NomVar2 doit être écrit entre apostrophes. A l'intérieur des apostrophes, utilisez des guillemets autour d'un nom de variable API, de variable spéciale ou de bloc de données. Le nom du bloc de données est entre guillemets mais pas le nom de la variable de bloc de données. Notez que, pour les variables de blocs de données, vous utilisez le nom du bloc et non son numéro.

**Exemples**

```
<-- AWP_In_Variable Name='SERVER:current_user_id'
Use='Data_Block_10.utilisateur_serveur' -->
```

Dans cet exemple, la variable spéciale SERVER:current\_user\_id est écrite dans la variable "utilisateur\_serveur" du bloc de données "Data\_Block\_10".

```
<-- AWP_Out_Variable Name='Poids'
Use='Data_Block_10.Données_cuve.Poids' -->
```

Dans cet exemple, la valeur dans l'élément de structure de bloc de données Data\_Block\_10.Données\_cuve.Poids peut être plus simplement référencé par son alias "Poids" dans tout le reste de la page Web personnalisée.

```
<!-- AWP_Out_Variable Name='Poids' Use='"Poids_cuve_lait_cru"' -->
```

Dans cet exemple, la valeur dans la variable API "Poids\_cuve\_lait\_cru" peut être plus simplement référencée par son alias "Poids" dans tout le reste de la page Web personnalisée.

Si un nom de variable ou de bloc de données comprend des caractères spéciaux, vous devez en plus utiliser des apostrophes ou guillemets ou des caractères d'échappement comme décrit à la rubrique Gestion des noms de variables contenant des caractères spéciaux (Page 567).

### 11.3.2.6 Définition de types Enum

Vous pouvez définir des types Enum dans vos pages personnalisées et affecter les éléments dans une commande AWP.

#### Syntaxe

```
<!-- AWP_Enum_Def Name='<nom du type Enum>' Values='<valeur>,
<valeur>,... ' -->
```

#### Paramètres

<nom du type Enum>	Nom du type énumération entre apostrophes ou entre guillemets
<valeur>	<constante>:<nom> La constante indique la valeur numérique pour l'affectation du type Enum. Le nombre total n'est pas borné. Le nom est la valeur affectée à l'élément Enum.

Notez que la chaîne entière d'affectations de valeur Enum est placée entre apostrophes, chaque affectation individuelle d'un élément du type Enum étant placée entre guillemets. La portée d'une définition de type Enum est globale pour les pages Web personnalisées. Si vous avez configuré vos pages Web personnalisées dans des dossiers de langue (Page 586), la définition du type Enum est globale pour toutes les pages dans le dossier de langue.

#### Exemple

```
<!-- AWP_Enum_Def Name='AlarmEnum' Values='0:"Pas d'alarmes",
1:"Cuve pleine", 2:"Tank is empty"' -->
```

### 11.3.2.7 Référencage de variables CPU avec un type Enum

Vous pouvez affecter une variable dans la CPU à un type Enum. Cette variable peut être utilisée à un autre endroit de votre page Web personnalisée dans une opération de lecture (Page 556) ou d'écriture (Page 557). Dans une opération de lecture, le serveur Web remplace la valeur numérique lue dans la CPU par la valeur textuelle Enum correspondante. Dans une opération d'écriture, le serveur Web remplace la valeur textuelle par la valeur entière de l'énumération correspondant au texte avant d'écrire la valeur dans la CPU.

#### Syntaxe

```
<!-- AWP_Enum_Ref Name='<NomVar>' Enum="<TypeEnum">" -->
```

#### Paramètres

<NomVar>	Nom de la variable API ou de la variable de bloc de données à associer au type Enum ou nom d'alias pour une variable API (Page 562) le cas échéant. NomVar doit être écrit entre apostrophes. A l'intérieur des apostrophes, utilisez des guillemets autour d'un nom de variable API ou d'un nom de bloc de données. Notez que, pour les variables de blocs de données, vous utilisez le nom du bloc et non son numéro. Le nom du bloc de données est entre guillemets mais pas le nom de la variable de bloc de données.
<EnumTypeEnum>	Nom du type énumération entre apostrophes ou entre guillemets

La portée d'une référence au type Enum est le fragment en cours.

#### Exemple de déclaration

```
<!-- AWP_Enum_Ref Name=' "Alarme" ' Enum="EnumAlarme" -->
```

#### Exemple d'utilisation dans une lecture de variables

```
<!-- AWP_Enum_Def Name='EnumAlarme' Values='0:"Pas d'alarmes",
1:"Cuve pleine", 2:"Cuve vide"' -->
<!-- AWP_Enum_Ref Name=' "Alarme" ' Enum="EnumAlarme" -->
...
<p>La valeur en cours de "Alarme" est := "Alarme":</p>
```

Si la valeur de "Alarme" dans la CPU est 2, la page HTML affiche 'La valeur en cours de "Alarme" est Cuve vide' car la définition du type Enum (Page 563) affecte la chaîne littérale "Cuve vide" à la valeur numérique 2.

#### Exemple d'utilisation dans une écriture de variables

```
<!-- AWP_Enum_Def Name='EnumAlarme' Values='0:"Pas d'alarmes",
1:"Cuve pleine", 2:"Cuve vide"' -->
<!-- AWP_In_Variable Name=' "Alarme" ' -->
<!-- AWP_Enum_Ref Name=' "Alarme" ' Enum='EnumAlarme' -->
...
<form method="POST">
<p><input type="hidden" name="Alarme" value="Cuve pleine" /></p>
<p><input type="submit" value='Activer Cuve pleine' /></p>
```

```
</form>
```

Comme la définition du type Enum (Page 563) affecte "Cuve pleine" à la valeur numérique 1, la valeur 1 est écrite dans la variable API nommée "Alarm" dans la CPU.

Notez que la clause Name dans la déclaration AWP\_In\_Variable doit correspondre exactement à la clause Name dans la déclaration AWP\_Enum\_Ref .

### Exemple d'utilisation dans une écriture de variables avec utilisation d'un alias

```
<!-- AWP_Enum_Def Name='EnumAlarme' Values='0:"Pas d'alarmes",
1:"Cuve pleine", 2:"Cuve vide"' -->
<!-- AWP_In_Variable Name='"Alarme"'
Use='"Bloc_données_4".Moteur1.Alarme'-->
<!-- AWP_Enum_Ref Name='"Alarme"' Enum='EnumAlarme' -->
...
<form method="POST">
<p><input type="hidden" name='"Alarme"' value="Cuve pleine" /></p>
<p><input type="submit" value='Activer Cuve pleine' /><p>
</form>
```

Comme la définition du type Enum (Page 563) affecte "Cuve pleine" à la valeur numérique 1, la valeur 1 est écrite dans l'alias "Alarme" qui correspond à la variable API nommée "Moteur1.Alarme" dans le bloc de données "Data\_Block\_4" dans la CPU.

Si un nom de variable ou de bloc de données comprend des caractères spéciaux, vous devez en plus utiliser des apostrophes ou guillemets ou des caractères d'échappement comme décrit à la rubrique Gestion des noms de variables contenant des caractères spéciaux (Page 567).

### 11.3.2.8 Création de fragments

STEP 7 convertit et sauvegarde les pages Web personnalisées sous la forme d'un DB de commande et de DB de fragment lorsque vous cliquez sur "Générer blocs" dans les propriétés CPU pour le serveur Web. Vous pouvez configurer des fragments spécifiques pour des pages spécifiques ou pour des parties de pages spécifiques. Vous pouvez identifier ces fragments par un nom et un numéro à l'aide de la commande AWP "AWP\_Start\_Fragment". Tout ce qu'il y a dans la page après la commande AWP\_Start\_Fragment appartient à ce fragment jusqu'à ce qu'une autre commande AWP\_Start\_Fragment soit émise ou jusqu'à ce que la fin de fichier soit atteinte.

### Syntaxe

```
<!-- AWP_Start_Fragment Name='<Nom>' [Type=<Type>] [ID=<id>] -->
```

## Paramètres

<Nom>	<p>Chaîne de texte : nom du DB de fragment</p> <p>Les noms de fragment doivent commencer par une lettre ou un trait de soulignement et contenir des lettres, des chiffres et des traits de soulignement. Le nom de fragment est une expression normale de la forme : [a-zA-Z_][a-zA-Z_0-9]*</p>
<Type>	<p>Manuel ou automatique</p> <p>Manuel : Le programme STEP 7 doit demander ce fragment et peut réagir en conséquence. L'exploitation du fragment doit être gérée avec STEP 7 et les variables du DB de commande.</p> <p>Automatique : Le serveur Web traite le fragment automatiquement.</p> <p>Si vous n'indiquez pas le paramètre "type", "automatique" est pris par défaut.</p>
<id>	<p>Numéro d'identification entier. Si vous n'indiquez pas le paramètre "id", le serveur Web affecte un numéro par défaut. Donnez un numéro d'ID faible aux fragments manuels. C'est au moyen de l'ID que le programme STEP 7 gère un fragment manuel.</p>
<Mode>	<p>Visible ou masqué</p> <p>Visible : Le contenu du fragment s'affichera sur la page Web personnalisée.</p> <p>Masqué : Le contenu du fragment ne s'affichera pas sur la page Web personnalisée.</p> <p>Si vous n'indiquez pas le paramètre "mode", "visible" est pris par défaut.</p>

## Fragments manuels

Si vous créez un fragment manuel pour une page Web personnalisée ou une partie d'une telle page, votre programme STEP 7 doit gérer le moment où le fragment est envoyé. Le programme STEP 7 doit définir les paramètres appropriés dans le DB de commande pour une page personnalisée en mode manuel, puis appeler l'instruction WWW avec le DB de commande modifié. Consultez le paragraphe Commande avancée de pages Web personnalisées (Page 590) pour plus d'informations sur la structure du DB de commande et la manipulation de pages et de fragments individuels.

### 11.3.2.9 Importation de fragments

Vous pouvez créer un fragment avec un nom à partir d'une partie de votre code HTML, puis importer ce fragment à un autre endroit dans votre jeu de pages Web personnalisées. Considérons, par exemple, un jeu de pages Web personnalisées qui comprend une page de démarrage et plusieurs autres pages HTML accessibles à l'aide de liens sur cette page de démarrage. Supposons que chacune de ces pages distinctes doive afficher le logo de l'entreprise. Vous pouvez réaliser cette opération en créant un fragment (Page 565) qui charge l'image du logo de l'entreprise. Chaque page HTML individuelle peut alors importer ce fragment pour afficher le logo de l'entreprise. Utilisez pour ce faire une commande AWP\_Import\_Fragment. Le code HTML pour ce fragment existe uniquement dans un fragment, mais vous pouvez importer ce DB de fragment autant de fois que nécessaire dans autant de pages Web que vous le désirez.

## Syntaxe

```
<!-- AWP_Import_Fragment Name='<Nom>' -->
```

## Paramètres

<Nom>	Chaîne de texte : nom du DB de fragment à importer
-------	----------------------------------------------------

## Exemple

Extrait de code HTML qui crée un fragment pour afficher une image :

```
<!-- AWP_Start_Fragment Name='Mon_logo_dentreprise' --><p></p>
```

Extrait de code HTML d'un autre fichier .html qui importe le fragment affichant l'image du logo :

```
<!-- AWP_Import_Fragment Name='Mon_logo_dentreprise' -->
```

Les deux fichiers .html (celui qui crée le fragment et celui qui l'importe) sont dans la structure de dossiers que vous définissez lorsque vous configurez les pages personnalisées dans STEP 7 (Page 569).

### 11.3.2.10 Combinaison de définitions

Lorsque vous déclarez des variables à utiliser dans vos pages Web personnalisées, vous pouvez combiner une déclaration de variable et un alias pour la variable (Page 562). Vous pouvez aussi déclarer plusieurs variables In\_Variables et plusieurs variables Out\_Variables dans une instruction.

## Exemples

```
<!-- AWP_In_Variable Name='"Niveau"', Name='"Poids"', Name='"Temp"' -->  
-->  
<--! AWP_Out_Variable Name='HEADER:Status', Use='"Etat"',  
Name='HEADER:Location', Use="Emplacement",  
Name='COOKIE_VALUE:name', Use="mon_cookie" -->  
<!-- AWP_In_Variable Name='Alarme' Use='"Data_block_10".Alarme' -->
```

### 11.3.2.11 Gestion des noms de variables contenant des caractères spéciaux

Lorsque vous indiquez des noms de variables dans des pages Web personnalisées, vous devez être particulièrement prudent si ces noms contiennent des caractères ayant une signification particulière.

## Lecture de variables

Vous utilisez la syntaxe suivante pour lire une variable (Page 556) :  
:=<NomVar>:

Les règles suivantes s'appliquent à la lecture de variables :

- Pour les noms de variables provenant de la table de variables API, placez le nom de variable entre guillemets.
- Pour les noms de variables qui sont des variables de bloc de données, placez le nom du bloc de données entre guillemets. La variable se trouve en dehors des guillemets.
- Pour les noms de variables qui sont des adresses d'E/S directes, des adresses de memento ou des noms d'alias, n'utilisez pas de guillemets ou apostrophes autour de la variable lue.
- Pour les noms de variables ou de variables de blocs de données contenant une barre oblique inversée, faites précéder la barre oblique inversée d'une autre barre oblique inversée.
- Si un nom de variable ou de variable de bloc de données contient un point, un signe inférieur à, un signe supérieur à ou une perluète, définissez pour la variable lue un alias sans caractères spéciaux et lisez la variable à l'aide de cet alias. Faites précéder les deux-points de noms de variables dans une clause Use d'une barre oblique inversée.

Tableau 11- 1 Exemples de lecture de variables

Nom du bloc de données	Nom de la variable	Commande de lecture
Sans objet	ABC:DEF	<code>&lt;!--AWP_Out_Variable Name='variable_speciale' Use ="ABC:DEF" --&gt; :=variable_speciale:</code>
Sans objet	T\	<code>:= "T\\" :</code>
Sans objet	A \B 'C :D	<code>&lt;!--AWP_Out_Variable Name='autre_variable_speciale' Use="A \\B \ 'C :D" ' --&gt; :=autre_variable_speciale:</code>
Sans objet	a<b	<code>&lt;!--AWP_Out_Variable Name='a_inférieur_à_b' Use="a&lt;b" ' --&gt; :=a inférieur à b:</code>
Data_block_1	Tag_1	<code>:= "Data_block_1".Tag_1:</code>
Data_block_1	ABC:DEF	<code>&lt;!-- AWP_Out_Variable Name='variable_speciale' Use="Data_block_1".ABC\ :DEF' --&gt; :=variable_speciale:</code>
DB A' B C D\$ E	Tag	<code>:= "DB A' B C D\$ E".Tag:</code>
DB:DB	Tag:Tag	<code>&lt;!--AWP_Out_Variable Name='ma_variable' Use ="DB:DB".Tag\ :Tag' --&gt; :=ma_variable:</code>



## Clauses Name et Use

Les commandes AWP AWP\_In\_Variable, AWP\_Out\_Variable, AWP\_Enum\_Def, AWP\_Enum\_Ref, AWP\_Start\_Fragment et AWP\_Import\_Fragment comportent des clauses Name. Les commandes de formulaire HTML telles que <input> et <select> ont également des clauses Name. AWP\_In\_Variable et AWP\_Out\_Variable peuvent en outre comporter des clauses Use. Quelle que soit la commande, la syntaxe des clauses Name et Use est identique en ce qui concerne la gestion des caractères spéciaux :

- Le texte que vous fournissez pour une clause Name ou Use doit être entre apostrophes. Si le nom concerné est une variable API ou un nom de bloc de données, utilisez des apostrophes pour la clause entière.
- A l'intérieur de la clause Name ou Use, les noms de blocs de données et les noms de variables API doivent être entre guillemets.
- Si un nom de variable ou de bloc de données comprend une apostrophe ou une barre oblique inversée, faites précéder ce caractère d'une barre oblique inversée. La barre oblique inversée est le caractère d'échappement dans le compilateur de commandes AWP.

Tableau 11- 2 Exemples de clauses Name

Nom du bloc de données	Nom de la variable	Exemples de clauses Name
Sans objet	ABC'DEF	Name=" 'ABC\ 'DEF" '
Sans objet	A \B 'C :D	Name=" 'A \\B \ 'C :D" '
Data_block_1	Tag_1	Name=" 'Data_block_1".Tag_1'
Data_block_1	ABC'DEF	Name=" 'Data_block_1".ABC\ 'DEF'
Data_block_1	A \B 'C :D	Name=" 'Data_block_1".A \\B \ 'C :D'
DB A' B C D\$ E	Tag	Name=" 'DB A\ ' B C D\$ E".Tag'

Les clauses Use utilisent les mêmes conventions que les clauses Name.

### Remarque

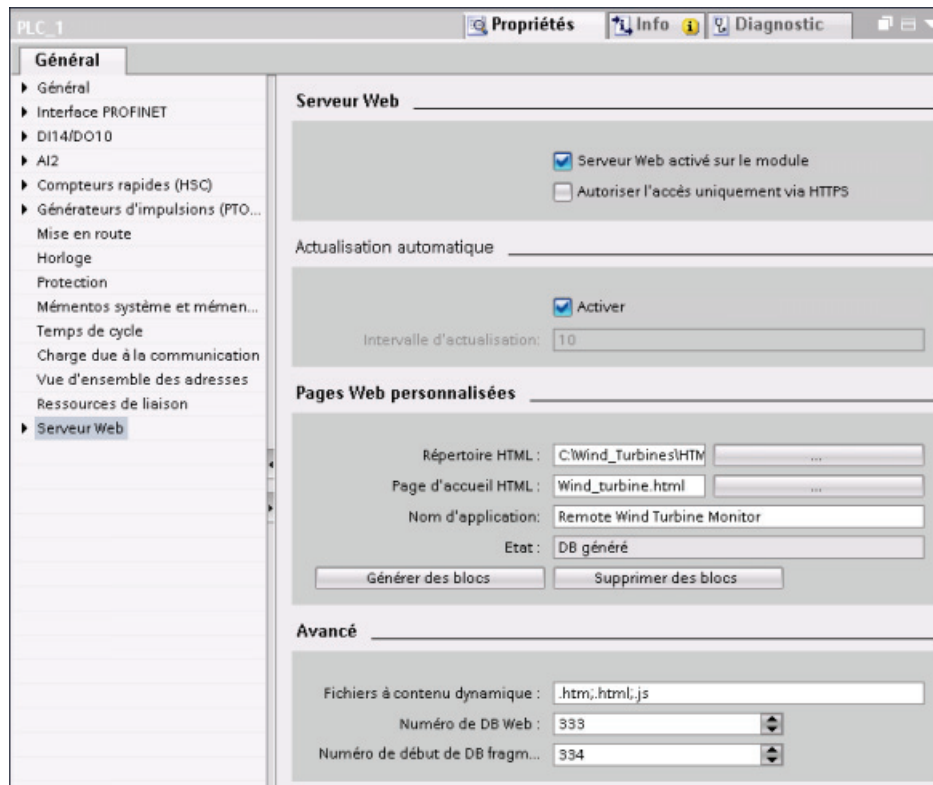
Quels que soient les caractères que vous utilisez dans votre page HTML, définissez le jeu de caractères de la page HTML à UTF-8 et sauvegardez-la à partir de l'éditeur avec le codage de caractères UTF-8.

### 11.3.3 Configuration de l'utilisation de pages Web personnalisées

Procédez comme suit pour configurer des pages Web personnalisées dans STEP 7 :

1. Sélectionnez la CPU dans la vue Configuration des appareils.

2. Affichez les propriétés "Serveur Web" dans la fenêtre d'inspection pour la CPU.



3. Si elle n'est pas déjà cochée, cochez la case "Activer le serveur Web".
4. Cochez la case "Autoriser l'accès uniquement via HTTPS" pour garantir que le serveur Web crypte les communications et pour augmenter la sécurité de votre CPU accessible par le Web.
5. Entrez le nom du dossier sur votre PC où vous avez sauvegardé la page HTML par défaut (page d'accueil) ou naviguez jusqu'à ce dossier.
6. Entrez le nom de la page par défaut.
7. Indiquez un nom pour votre application (facultatif). Le nom de l'application est utilisé pour former des sous-catégories ou des groupes de pages Web. Lorsqu'un nom d'application existe, l'URL apparaît dans le format : `http://ww.xx.yy.zz/awp/<nom application>/<nom page>.html`.
8. Indiquez les extensions de nom de fichier dans lesquelles rechercher la présence de commandes AWP. Par défaut, STEP 7 analyse les fichiers d'extension .htm, .html et .js. Si vous avez des extensions de fichier supplémentaires, ajoutez-les.
9. Conservez la valeur par défaut pour le numéro du DB Web ou entrez un numéro de votre choix. Il s'agit du numéro du DB de commande pilotant l'affichage des pages Web.
10. Conservez la valeur par défaut pour le numéro de début des DB de fragment ou entrez un numéro de votre choix. Il s'agit du premier des DB de fragment qui contiennent les pages Web.

### Génération de blocs de programme

Lorsque vous cliquez sur le bouton "Générer blocs", STEP 7 génère des blocs de données à partir des pages HTML dans le répertoire source HTML que vous avez indiqué, ainsi qu'un bloc de données de commande pour la gestion de vos pages Web. Vous pouvez définir les attributs selon les besoins de votre application (Page 571). STEP 7 génère aussi un jeu de blocs de données de fragment pour contenir la représentation de toutes vos pages HTML. Lorsque vous générez les blocs de données, STEP 7 actualise les propriétés afin qu'elles affichent le numéro du bloc de données de commande et le numéro du premier des blocs de données de fragment. Une fois les blocs de données générés, vos pages Web personnalisées font partie de votre programme STEP 7. Les blocs correspondant à ces pages apparaissent dans le dossier de serveur Web qui se trouve dans le dossier Blocs système sous Blocs de programme dans le navigateur du projet.

### Suppression de blocs de programme

Pour supprimer les blocs de données précédemment générés, cliquez sur le bouton "Supprimer blocs". STEP 7 supprime de votre projet le bloc de données de commande et tous les blocs de données de fragment correspondant aux pages Web personnalisées.


## 11.3.4 Programmation de l'instruction WWW pour les pages Web personnalisées

Votre programme utilisateur STEP 7 doit contenir et exécuter l'instruction WWW afin que les pages Web personnalisées soient accessibles à partir des pages Web standard. Le bloc de données de commande est le paramètre d'entrée de l'instruction WWW et indique le contenu des pages tel que représenté dans les blocs de données de fragment, ainsi que des informations d'état et de commande. STEP 7 crée le bloc de données de commande lorsque vous cliquez sur le bouton "Générer blocs" dans la configuration des pages Web personnalisées (Page 569).

### Programmation de l'instruction WWW

Le programme STEP 7 doit exécuter l'instruction WWW pour rendre les pages Web personnalisées accessibles à partir des pages Web standard. Mais vous voudrez peut-être que les pages Web personnalisées ne soient disponibles que dans certaines circonstances, en fonction des exigences de votre application et de vos préférences. Dans ce cas, la logique de votre programme peut gérer le moment où l'instruction WWW est appelée.

Tableau 11- 3 Instruction WWW

CONT/LOG	SCL	Description
	<pre>ret_val := WWW(     ctrl_db:=_uint_in_);</pre>	Permet l'accès aux pages Web personnalisées à partir des pages Web standard.

Vous devez indiquer le paramètre d'entrée bloc de données de commande (CTRL\_DB) qui correspond au numéro de DB entier du DB de commande. Vous trouverez ce numéro de DB de commande (numéro du DB Web) dans les propriétés du serveur Web de la CPU après avoir créé les blocs pour les pages Web personnalisées. Entrez le numéro de DB entier comme paramètre CTRL\_DB de l'instruction WWW. La valeur en retour RET\_VAL contient le résultat de la fonction. Notez que l'instruction WWW s'exécute de manière asynchrone et que la sortie RET\_VAL peut avoir une valeur initiale égale à 0 même si une erreur peut se produire par la suite. Le programme peut vérifier l'état du DB de commande pour s'assurer que l'application a été lancée avec succès ou contrôler RET\_VAL à l'aide d'un appel ultérieur de WWW.

Tableau 11- 4 Valeur en retour

RET_VAL	Description
0	Pas d'erreur
16#00yx	<p>x : La requête représentée par le bit respectif est en état d'attente :</p> <p>x=1 : requête 0</p> <p>x=2 : requête 1</p> <p>x=4 : requête 2</p> <p>x=8 : requête 3</p> <p>Une opération OU logique peut être effectuée sur les valeurs de x, le résultat représentant alors les états d'attente de plusieurs requêtes. Par exemple, x égal à 6 signifie que les requêtes 1 et 2 sont en attente.</p> <p>y : 0: pas d'erreur ; 1 : il y a une erreur et "last_error" a été mis à 1 dans le DB de commande</p>
16#803a	Le DB de commande n'est pas chargé.
16#8081	Le DB de commande a un type, un format ou une version incorrecte.
16#80C1	Il n'y a pas de ressources disponibles pour initialiser l'application Web.

### Utilisation du DB de commande

STEP 7 crée le bloc de données de commande lorsque vous cliquez sur "Générer blocs" et affiche le numéro du DB de commande dans les propriétés des pages Web personnalisées. Vous pouvez également trouver le DB de commande dans le dossier Blocs de programme du navigateur de projet.

Typiquement, votre programme STEP 7 utilise le DB de commande directement tel qu'il a été créé par l'opération "Générer blocs" sans manipulation supplémentaire. Toutefois, le programme utilisateur STEP 7 peut définir des commandes globales dans le DB de commande afin de désactiver le serveur Web ou le réactiver par la suite. En outre, le programme utilisateur STEP 7 doit gérer le comportement des pages personnalisées que vous créez comme DB de fragment manuels (Page 569) à l'aide d'une table de requête dans le DB de commande. Pour plus d'informations sur ces tâches avancées, reportez-vous à la rubrique Commande avancée de pages Web personnalisées (Page 590).

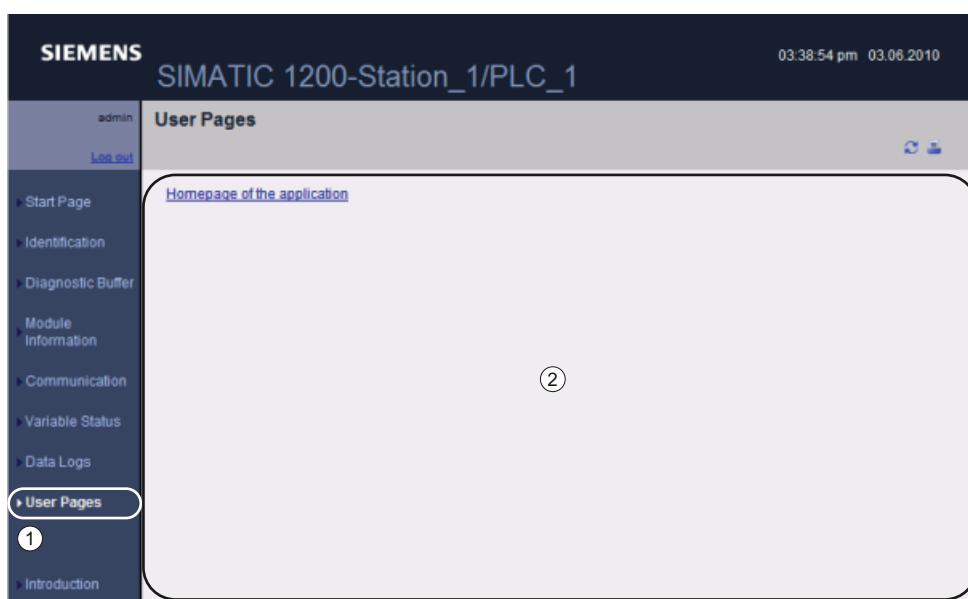
### 11.3.5 Chargement des blocs de programme dans la CPU

Une fois que vous avez généré les blocs pour les pages Web personnalisées, ils font partie de votre programme STEP 7 tout comme tous les autres blocs de programme. Vous procédez de la manière usuelle pour les charger dans la CPU. Notez que vous pouvez charger les blocs de programme des pages Web personnalisées uniquement lorsque la CPU est à l'état ARRET.

### 11.3.6 Accès aux pages Web personnalisées depuis le PC

Vous accédez à vos pages Web personnalisées à partir des pages Web standard (Page 535). Les pages Web standard comportent un lien pour les "Pages utilisateur" dans le menu de gauche où les liens vers les autres pages sont affichés. Lorsque vous cliquez sur le lien "Pages utilisateur", votre navigateur Web va à la page permettant d'accéder à votre page par défaut. Dans la zone au contenu personnalisé, la navigation se fait conformément à la manière dont vous avez conçu les différentes pages.

Le serveur Web affiche le contenu de la page personnalisée à l'intérieur du cadre de l'affichage de pages Web standard. Ainsi, le cadre d'en-tête en haut et les cadres d'ouverture de session et de navigation à gauche restent en positions fixes.



- ① Lien vers la page par défaut des pages Web personnalisées
- ② Zone de contenu des pages Web personnalisées

### 11.3.7 Contraintes spécifiques aux pages Web personnalisées

Les contraintes pour les pages Web standard (Page 593) s'appliquent également aux pages Web personnalisées. Il faut en outre tenir compte de quelques points spécifiques pour les pages Web personnalisées.

### Espace en mémoire de chargement

Lorsque vous cliquez sur "Générer blocs", vos pages Web personnalisées deviennent des blocs de données qui ont besoin d'espace en mémoire de chargement. Si une carte mémoire est installée, vous disposez de la capacité de votre carte mémoire comme espace de mémoire de chargement externe pour les pages Web personnalisées.

Si vous n'avez pas installé de carte mémoire, ces blocs prennent de l'espace en mémoire de chargement interne qui est limitée selon votre modèle de CPU.

Vous pouvez vérifier la quantité de mémoire de chargement utilisée et disponible à l'aide des outils en ligne et de diagnostic dans STEP 7. Vous pouvez également consulter les propriétés des blocs individuels que STEP 7 génère à partir de vos pages Web personnalisées et voir la mémoire de chargement qu'ils consomment.

---

#### Remarque

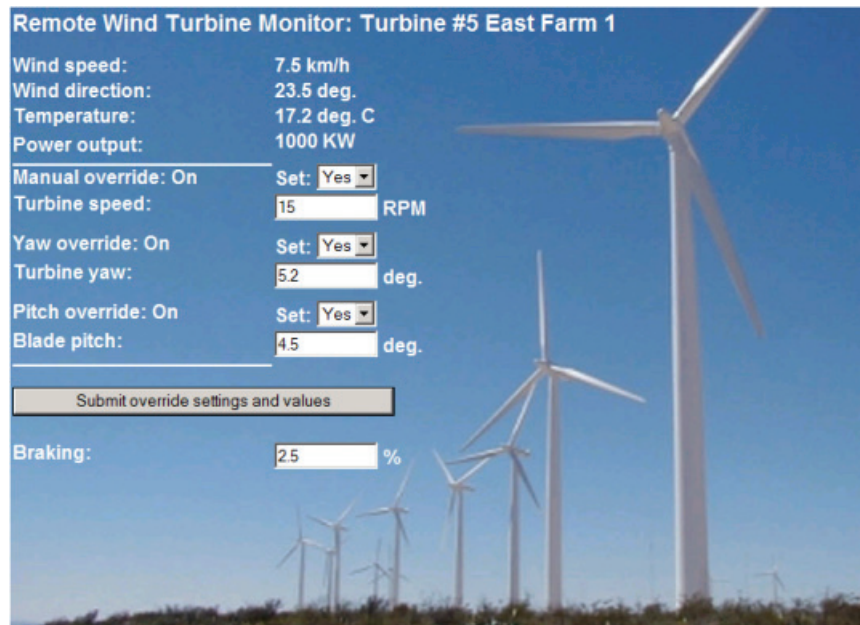
Si vous devez réduire l'espace nécessaire pour vos pages Web personnalisées, utilisez moins d'images si c'est possible.

---

### 11.3.8 Exemple de page Web personnalisée

#### 11.3.8.1 Page Web pour le contrôle-commande d'une éolienne

Prenons, comme exemple de page Web personnalisée, une page Web utilisée pour le contrôle-commande à distance d'une éolienne :



---

### Remarque

Cet exemple de page est en anglais, mais vous pouvez bien sûr utiliser n'importe quelle langue lorsque vous concevez vos propres pages HTML.

---

## Description

Dans cette application, chaque éolienne d'une ferme d'éoliennes est équipée d'un S7-1200 pour la commande de l'éolienne. Dans le programme STEP 7, chaque éolienne dispose d'un bloc de données avec des données spécifiques concernant cette éolienne et sa position.

La page Web personnalisée fournit un accès à distance à l'éolienne à partir d'un PC. Un utilisateur peut se connecter aux pages Web standard de la CPU d'une éolienne particulière et accéder à la page Web personnalisée de contrôle d'éolienne à distance pour visualiser les données de cette éolienne. Un utilisateur admin peut également faire passer l'éolienne en mode manuel et piloter les variables pour la vitesse, le lacet et le pas à partir de la page Web. Il peut en outre définir une valeur de freinage, que l'éolienne soit en mode manuel ou automatique.

Le programme STEP 7 doit surveiller les valeurs booléennes de substitution du mode automatique et, si elles sont à 1, utiliser les valeurs de vitesse d'éolienne, lacet et pas entrées par l'utilisateur. Si ce n'est pas le cas, le programme ne tient pas compte de ces valeurs.

## Fichiers utilisés

Cet exemple de page Web personnalisée consiste en trois fichiers :

- **Wind\_turbine.html** : Il s'agit de la page HTML qui implémente l'écran montré plus haut à l'aide de commandes AWP pour accéder aux données de l'automate.
- **Wind\_turbine.css** : Il s'agit de la feuille de style en cascade qui contient les règles de formatage de la page HTML. L'utilisation d'une feuille de style en cascade est facultative, mais elle peut simplifier la conception des pages Web.
- **Wind\_turbine.jpg** : Il s'agit de l'image de fond utilisée par la page HTML. L'utilisation d'images dans des pages Web personnalisées est, bien sûr, optionnelle et requiert en outre de l'espace supplémentaire dans la CPU.

Ces fichiers ne sont pas fournis avec votre installation, mais sont décrits comme exemple.

## Mise en oeuvre

La page HTML utilise des commandes AWP pour lire des valeurs dans l'API (Page 556) pour les champs d'affichage et pour écrire des valeurs dans l'API (Page 557) correspondant à des données saisies par l'utilisateur. Cette page utilise également des commandes AWP pour la définition (Page 563) et la référence (Page 564) de types Enum pour la gestion des réglages ON/OFF.

La première partie de la page affiche une ligne d'en-tête qui comprend le numéro de l'éolienne et sa position.

**Remote Wind Turbine Monitor: Turbine #5 East Farm 1**

La partie suivante de la page affiche les conditions atmosphériques au niveau de l'éolienne. Ces champs reçoivent leurs valeurs d'entrées/sorties qui sont situées sur le site de l'éolienne et qui fournissent la vitesse et la direction du vent ainsi que la température en cours.

Wind speed:	7.5 km/h
Wind direction:	23.5 deg.
Temperature:	17.2 deg. C

La page affiche ensuite la puissance utile de l'éolienne telle qu'elle est lue dans le S7-1200.

Power output:	1000 KW
---------------	---------

La partie suivante permet la commande manuelle de l'éolienne qui se substitue à la commande automatique normale par le S7-1200. Elle comprend :

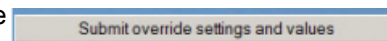
- Manual override (Manuel prioritaire) : active le mode manuel de la turbine. Le programme utilisateur STEP 7 requiert que le mode manuel soit sur vrai pour utiliser l'un des réglages manuels pour la vitesse d'éolienne, le lacet ou le pas quelconque.
- Yaw override (Lacet manuel prioritaire) : active la priorité manuelle du réglage du lacet et un réglage manuel pour le lacet. Le programme utilisateur STEP 7 n'applique le réglage du lacet que si Manuel prioritaire et Lacet manuel prioritaire sont tous deux vrais.
- Pitch override (Pas manuel prioritaire) : active la priorité manuelle pour le pas des pales. Le programme utilisateur STEP 7 n'applique le réglage du pas des pales que si Manuel prioritaire et Pas manuel prioritaire sont tous deux vrais.

Manual override: On	Set: Yes
Turbine speed:	15 RPM

Yaw override: On	Set: Yes
Turbine yaw:	52 deg.

Pitch override: On	Set: Yes
Blade pitch:	4.5 deg.

La page HTML comprend un bouton Soumettre pour transmettre les réglages de priorité à l'automate.



Le champ d'entrée de freinage utilisateur fournit une valeur manuelle pour un pourcentage de freinage. Le programme utilisateur STEP 7 accepte la valeur de freinage même si le mode manuel n'est pas activé.

Braking:	2.5 %
----------	-------

En outre, la page HTML utilise une commande AWP pour écrire la variable spéciale (Page 561) contenant l'identification de l'utilisateur qui accède à la page dans une variable de la table de variables API.



### 11.3.8.2 Lecture et affichage des données de l'automate

La page HTML de contrôle d'éolienne à distance utilise de nombreuses commandes AWP pour lire des données dans l'automate (Page 556) et les afficher sur la page. Considérons, par exemple, le code HTML pour l'affichage de la puissance utile, comme illustré dans cette partie de l'exemple de page Web :



#### Exemple de code HTML

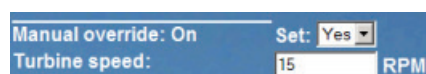
L'extrait suivant de la page HTML de contrôle d'éolienne à distance affiche le texte "Power Output:" dans la cellule gauche d'une ligne de tableau, lit la variable pour la puissance utile et l'affiche dans la cellule de droite du tableau conjointement avec le texte pour les unités, KW.

La commande AWP := "Data\_block\_1".PowerOutput: exécute l'opération de lecture. Notez que les blocs de données sont référencés par leur nom et pas par leur numéro ("Data\_block\_1" et non "DB1").

```
<tr style="height:2%;">
<td>
<p>Power output:</p>
</td>
<td>
<p style="margin-bottom:5px;"> := "Data_block_1".PowerOutput: KW</p>
</td>
</tr>
```

### 11.3.8.3 Utilisation d'un type Enum

%La page HTML de contrôle d'éolienne à distance ("Remote Wind Turbine Monitor") utilise des types Enum pour trois instances dans lesquelles la page HTML affiche "ON" ou "OFF" pour une valeur booléenne et pour lesquelles l'utilisateur définit une valeur booléenne. Le type Enum pour "ON" correspond à une valeur de 1 et le type Enum pour "OFF" correspond à une valeur de 0. Considérons, par exemple, le code HTML pour la lecture et l'écriture du paramètre d'activation du mode manuel dans la valeur "Data\_block\_1".ManualOverrideEnable à l'aide d'un type Enum :



#### Exemple de code HTML

Les extraits suivants de la page HTML de contrôle d'éolienne à distance montrent comment déclarer un type Enum appelé "OverrideStatus" et ayant les valeurs "Off" et "On" pour 0 et 1 et comment définir une référence au type Enum OverrideStatus pour la variable booléenne ManualOverrideEnable dans le bloc de données "Data\_block\_1". Notez qu'il doit y avoir pour chaque déclaration AWP\_Enum\_Ref une déclaration AWP\_In\_Variable correspondante pour la variable de bloc de données ou la variable API si la page HTML écrit dans la variable par le biais d'un type Enum.

```
<!-- AWP_In_Variable Name='Data_block_1'.ManualOverrideEnable' -->
<!-- AWP_Enum_Def Name="OverrideStatus" Values='0:"Off",1:"On"' -->
<!-- AWP_Enum_Ref Name='Data_block_1'.ManualOverrideEnable'
Enum="OverrideStatus" -->
```

La page HTML comprend un champ d'affichage dans une cellule de tableau pour l'état en cours de ManualOverrideEnable. Elle n'utilise qu'une commande normale de lecture de variable mais grâce à l'utilisation du type Enum préalablement déclaré et référencé, la page affiche "Off" ou "On" au lieu de 0 ou 1.

```
<td style="width:24%; border-top-style: Solid; border-top-width:
2px; border-top-color: #ffffff;">
<p>Manual override: :=Data_block_1.ManualOverrideEnable:</p>
</td>
```

La page HTML inclut une liste de sélection déroulante pour que l'utilisateur puisse changer la valeur de ManualOverrideEnable. La liste de sélection utilise les textes "Yes" et "No" à afficher dans les listes de sélection. Grâce à l'utilisation du type Enum, "Yes" est corrélé à la valeur "On" du type Enum et "No" est corrélé à la valeur "Off". La sélection vide laisse la valeur de ManualOverrideEnable telle quelle.

```
<select name='Data_block_1.ManualOverrideEnable'>
<option value=':Data_block_1.ManualOverrideEnable:'> </option>
<option value="On">Yes</option>
<option selected value="Off">No</option>
</select>
```

La liste de sélection est incluse dans un formulaire sur la page HTML. Lorsque l'utilisateur clique sur le bouton Soumettre, la page transmet le formulaire, ce qui inscrit la valeur "1" dans la variable booléenne ManualOverrideEnable du bloc Data\_block\_1 si l'utilisateur a sélectionné "Yes" ou la valeur "0" si l'utilisateur a sélectionné "No".

#### 11.3.8.4 Ecriture d'une entrée utilisateur dans l'automate

La page HTML de contrôle d'éolienne à distance comprend plusieurs commandes AWP pour l'écriture de données dans le contrôleur (Page 557). La page HTML déclare des variables AWP\_In\_Variables pour les variables booléennes de sorte qu'un utilisateur administrateur peut placer l'éolienne en mode manuel et activer la substitution de valeurs manuelles pour la vitesse de l'éolienne, le lacet et/ou le pas des pales. La page utilise également des variables AWP\_In\_Variables pour permettre à un utilisateur admin de régler par la suite des valeurs à virgule flottante pour la vitesse, le lacet et le pas de l'éolienne et pour le pourcentage de freinage. La page utilise une commande d'envoi de formulaire HTTP pour écrire les variables AWP\_In\_Variables dans l'automate.

Considérons, par exemple, le code HTML pour le réglage manuel de la valeur de freinage :



## Exemple de code HTML

L'extrait suivant de la page HTML de contrôle d'éolienne à distance déclare d'abord pour "Data\_block\_1" une variable AWP\_In\_Variable qui permet à la page HTML d'écrire dans n'importe quelle variable du bloc de données "Data\_block\_1". La page affiche le texte "Braking:" dans la cellule gauche d'une ligne de tableau. La cellule droite de la ligne de tableau correspond au champ qui accepte l'entrée utilisateur pour la variable "Braking" de "Data\_block\_1". La valeur entrée par l'utilisateur se situe dans un formulaire HTML qui utilise la méthode HTML "POST" pour envoyer les données textuelles entrées à la CPU. La page lit alors la valeur de freinage en cours dans l'automate et l'affiche dans le champ d'entrée de données.

Un utilisateur admin peut alors utiliser cette page pour écrire une valeur de freinage dans le bloc de données de la CPU qui commande le freinage.

```
<!-- AWP_In_Variable Name='Data_block_1' -->
...
<tr style="vertical-align: top; height: 2%;">
<td style="width: 22%;"><p>Braking:</p></td>
<td>
<form method="POST">
<p><input name='Data_block_1'.Braking' size="10" type="text"> %</p>
</form>
</td>
</tr>
```

---

### Remarque

Notez que, si une page personnalisée comporte un champ d'entrée de données pour une variable de bloc de données inscriptible, l'utilisateur doit écrire la chaîne entre apostrophes lorsqu'il l'entre dans le champ.

---

### Remarque

Notez que si vous déclarez l'ensemble d'un bloc de données dans une déclaration AWP\_In\_Variable (par exemple, <!-- AWP\_In\_Variable Name=""Data\_block\_1"" -->), il est possible d'écrire dans chaque variable de ce bloc de données à partir de la page Web personnalisée. Utilisez cette méthode lorsque vous voulez que toutes les variables d'un bloc de données soient inscriptibles. Sinon, si vous voulez qu'il ne soit possible d'écrire que dans certaines variables spécifiques du bloc de données à partir de la page Web personnalisée, déclarez spécifiquement les variables concernées à l'aide d'une déclaration telle que <!-- AWP\_In\_Variable Name=""Data\_block\_1".Braking' -->.

---

### 11.3.8.5 Ecriture d'une variable spéciale

La page Web de contrôle d'éolienne à distance écrit la variable spéciale SERVER:current\_user\_id dans une variable API dans la CPU. Dans ce cas, la valeur de la variable API contient l'ID utilisateur de la personne qui accède à la page Web de contrôle d'éolienne à distance. Actuellement, l'utilisateur admin a l'ID utilisateur 1, de sorte que la variable API a la valeur 1.

La variable spéciale est écrite dans l'API par la page Web et ne nécessite pas d'interface utilisateur.

Exemple de code HTML

```
<!-- AWP_In_Variable Name="SERVER:current_user_id" Use="User_ID"-->
```

11.3.8.6 Référence : listage HTML de la page Web de contrôle d'éolienne à distance

Wind\_turbine.html

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd"><!--
Ce programme de test simule une page Web pour le contrôle-commande
d'une éolienne.
Variables API et variables de blocs de données requises dans STEP 7
:
```

```
Variable API :
User_ID: Int
```

```
Blocs de données :
Data_block_1
```

```
Variables dans Data_Block_1 :
Location: String
TurbineNumber: Int
WindSpeed: Real
WindDirection: Real
Temperature: Real
PowerOutput: Real
ManualOverrideEnable: Bool
TurbineSpeed: Real
YawOverride: Bool
Yaw: Real
PitchOverride: Bool
Pitch: Real
Braking: Real
```

La page Web personnalisée affiche les valeurs actuelles pour les données API et fournit une liste de sélection pour définir les trois valeurs booléennes à l'aide d'une affectation de type d'énumération. Le bouton "Soumettre" permet de transmettre les valeurs booléennes sélectionnées ainsi que les champs de saisie de données pour la vitesse (TurbineSpeed), le lacet (Yaw) et le pas (Pitch). La valeur pour le freinage peut être définie sans utiliser le bouton "Soumettre".

Aucun programme STEP 7 réel n'est nécessaire pour utiliser cette page. Théoriquement, le programme STEP 7 agirait uniquement sur les valeurs de vitesse, lacet et pas si les valeurs booléennes associées étaient à 1. La seule exigence de STEP 7 est d'appeler l'instruction WWW avec les numéros des blocs de données générés pour cette page.

```
-->
<!-- AWP_In_Variable Name="'Data_block_1'" -->
<!-- AWP_In_Variable Name="'Data_block_1'.ManualOverrideEnable' -->
<!-- AWP_In_Variable Name="'Data_block_1'.PitchOverride' -->
<!-- AWP_In_Variable Name="'Data_block_1'.YawOverride' -->
```

```

<!-- AWP_In_Variable Name="SERVER:current_user_id" Use="User_ID"-->
<!-- AWP_Enum_Def Name="OverrideStatus" Values='0:"Off",1:"On"' -->
<!-- AWP_Enum_Ref Name=' "Data_block_1".ManualOverrideEnable '
Enum="OverrideStatus" -->
<!-- AWP_Enum_Ref Name=' "Data_block_1".PitchOverride '
Enum="OverrideStatus" -->
<!-- AWP_Enum_Ref Name=' "Data_block_1".YawOverride '
Enum="OverrideStatus" -->

<html>
<head>
<meta http-equiv="content-type" content="text/html; charset=utf-
8"><link rel="stylesheet" href="Wind_turbine.css">
<title>Contrôle d'éolienne à distance</title>
</head>
<body>
<table style="background-image: url('./Wind_turbine.jpg'); width:
904px; height: 534px;" cellpadding="0" cellspacing="2"><tr
style="height: 2%;"><td colspan="2">
<h2>Remote Wind Turbine Monitor: Turbine
#:="Data_block_1".TurbineNumber: :="Data_block_1".Location:</h2>
</td>

<tr style="height: 2%;"><td style="width: 24%;"><p>Wind
speed:</p></td>
<td><p> :="Data_block_1".WindSpeed: km/h</p></td>
</tr>

<tr style="height: 2%;">
<td style="width: 24%;"><p>Wind direction:</p></td>
<td><p> :="Data_block_1".WindDirection: deg.</p></td>
</tr>

<tr style="height: 2%;"><td style="width:
24%;"><p>Temperature:</p></td>
<td><p> :="Data_block_1".Temperature: deg. C</p></td>
</tr>

<tr style="height: 2%;">
<td style="width: 24%;"><p>Power output:</p></td>
<td><p style="margin-bottom:5px;"> :="Data_block_1".PowerOutput:
KW</p>
</td>
</tr>

<form method="POST" action="">
<tr style="height: 2%;" >
<td style="width=24%; border-top-style: Solid; border-top-width:
2px; border-top-color: #ffffff;">
<p>Manual override: :="Data_block_1".ManualOverrideEnable:</p>
</td>
<td class="Text">Set:

<select name=' "Data_block_1".ManualOverrideEnable '>
<option value=' :="Data_block_1".ManualOverrideEnable: '> </option>

```

```

<option value="On">Yes</option>
<option value="Off">No</option>
</select>

</td>
</tr>

<tr style="vertical-align: top; height: 2%;"><td style="width:
24%;"><p>Turbine speed:</p></td>
<td>
<p style="margin-bottom:5px;"><input
name=' "Data_block_1".TurbineSpeed' size="10"
value=' :="Data_block_1".TurbineSpeed:' type="text"> RPM</p>
</td>
</tr>

<tr style="vertical-align: top; height: 2%;">
<td style="width: 24%;">
<p>Yaw override: :="Data_block_1".YawOverride: </p>
</td>
<td class="Text">Set:

<select name=' "Data_block_1".YawOverride'>
<option value=' :="Data_block_1".YawOverride:'> </option>
<option value="On">Yes</option>
<option value="Off">No</option>
</select>

</td>
</tr>

<tr style="vertical-align: top; height: 2%;">
<td style="width: 24%;">
<p>Turbine yaw:</p>
</td>
<td>
<p style="margin-bottom:5px;"><input name=' "Data_block_1".Yaw'
size="10" value=' :="Data_block_1".Yaw:' type="text"> deg.</p>
</td>
</tr>

<tr style="vertical-align: top; height: 2%;">
<td style="width: 24%;">
<p>Pitch override: :="Data_block_1".PitchOverride: </p>
</td>
<td class="Text">Set:

<select name=' "Data_block_1".PitchOverride'>
<option value=' :="Data_block_1".PitchOverride:'> </option>
<option value="On">Yes</option>
<option value="Off">No</option>
</select>

</td>
</tr>

```

```

<tr style="vertical-align: top; height: 2%;">
<td style="width=24%; border-bottom-style: Solid; border-bottom-
width: 2px; border-bottom-color: #ffffff;">
<p>Blade pitch:</p>
</td>
<td>
<p style="margin-bottom:5px;"><input name="'Data_block_1'.Pitch'
size="10" value=':="'Data_block_1'.Pitch:' type="text"> deg.</p>
</td>

</tr>
<tr style="height: 2%;">
<td colspan="2"><br>
<input type="submit" value="Submit override settings and values">
</td>
</tr>
</form>

<tr style="vertical-align: top; height: 2%;">
<td style="width: 24%;"><p>Braking:</p></td>
<td>
<form method="POST" action="">
<p> <input name="'Data_block_1'.Braking' size="10"
value=':="'Data_block_1'.Braking:' type="text"> %</p>
</form>
</td>
</tr>
<tr><td></td></tr>

</table>
</body>
</html>

```

Wind\_turbine.css

```

H2 {
font-family: Arial;
font-weight: bold;
font-size: 14.0pt;
color: #FFFFFF;
margin-top: 0px;
margin-bottom: 10px;
}
P {
font-family: Arial;
font-weight: bold;
color: #FFFFFF;
font-size: 12.0pt;
margin-top: 0px;
margin-bottom: 0px;
}
TD.Text {
font-family: Arial;
font-weight: bold;

```

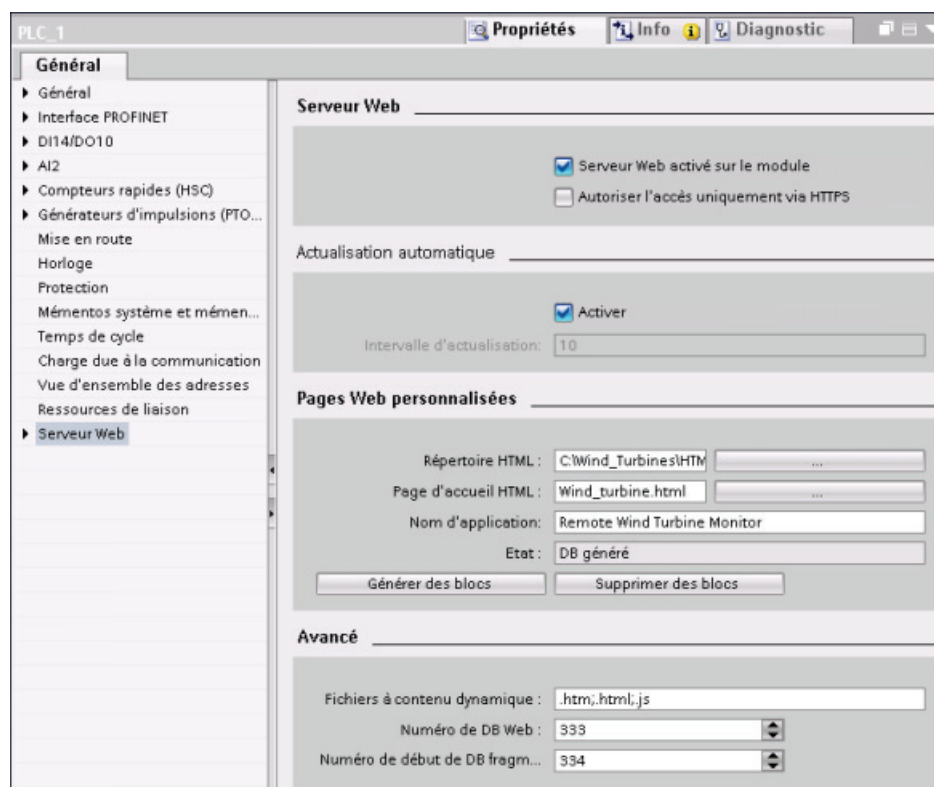
```
color: #FFFFFF;  
font-size: 12.0pt;  
margin-top: 0px;  
margin-bottom: 0px;  
}
```

#### 11.3.8.7 Configuration de l'exemple de page Web dans STEP 7

Pour inclure la page HTML de contrôle d'éolienne à distance en tant que page Web personnalisée pour le S7-1200, vous configurez les données concernant la page HTML dans STEP 7 et vous créez des blocs de données à partir de la page HTML.



Accédez aux propriétés CPU du S7-1200 qui commande l'éolienne et entrez les informations de configuration dans les propriétés de pages Web personnalisées du serveur Web :



### Champs de configuration

- HTML directory (répertoire HTML) : Ce champ indique le nom de chemin complet du dossier dans lequel se trouve la page par défaut (ou page d'accueil) sur l'ordinateur. Le bouton "..." vous permet de naviguer jusqu'au dossier dont vous avez besoin.
- Default HTML page (page HTML par défaut) : Ce champ indique le nom de fichier de la page par défaut ou page d'accueil de l'application HTML. Le bouton "..." vous permet de sélectionner le fichier dont vous avez besoin. Dans cet exemple, WindTurbine.html est la page HTML par défaut. L'exemple de contrôle d'éolienne à distance est constitué d'une seule page mais, dans d'autres applications personnalisées, la page par défaut peut appeler d'autres pages au moyen de liens sur la page par défaut. Dans le code HTML, la page par défaut doit référencer les autres pages par rapport au dossier source HTML.
- Application name (nom de l'application) : Ce champ optionnel contient le nom que le navigateur Web reporte dans le champ d'adresse quand il affiche la page. Il s'agit de "Remote Wind Turbine Monitor" dans cet exemple, mais ce pourrait être n'importe quel nom.

Aucun autre champ ne nécessite de configuration.

## Etapes finales

Pour utiliser la page de contrôle d'éolienne à distance telle que configurée, générez les blocs, programmez l'instruction WWW (Page 571) avec, comme paramètre d'entrée, le numéro du DB de commande généré, chargez les blocs de programme dans la CPU et faites passer la CPU à l'état MARCHE.

Plus tard, lorsqu'un opérateur accédera aux pages Web standard du S7-1200 qui commande l'éolienne, la page de contrôle d'éolienne à distance sera accessible via le lien "Pages utilisateur" dans la barre de navigation. Cette page offre maintenant les moyens pour réaliser un contrôle-commande de l'éolienne.

### 11.3.9 Configuration de pages Web personnalisées dans plusieurs langues








Le serveur Web vous offre la possibilité de configurer des pages Web personnalisées dans les langues suivantes :

- allemand (de)
- anglais (en)
- espagnol (es)
- français (fr)
- italien (it)
- chinois (zh)

Pour ce faire, vous configurez vos pages HTML dans une structure de dossiers (Page 586) qui correspond aux différentes langues et vous définissez un cookie spécifique nommé "siemens\_automation\_language" depuis vos pages (Page 587). Le serveur Web réagit à ce cookie et commute sur la page par défaut dans le dossier de langue correspondant.

#### 11.3.9.1 Création de la structure de dossiers

Pour fournir des pages Web personnalisées dans plusieurs langues, vous créez une structure de dossiers sous votre répertoire HTML. Les noms de dossier à deux lettres sont spécifiques et doivent être attribués comme suit :

 html	
 de	de : allemand
 en	en : anglais
 es	es : espagnol
 fr	fr : français
 it	it : italien
 zh	zh : chinois

Vous pouvez inclure au même niveau tout autre dossier dont vos pages ont besoin, par exemple des dossiers pour les images ou les scripts.

Vous pouvez créer n'importe quel sous-ensemble des dossiers de langue. Vous n'êtes pas obligé d'inclure la totalité des six langues. A l'intérieur des dossiers de langue, vous créez et programmez vos pages HTML dans la langue appropriée.

### 11.3.9.2 Programmation du changement de langue

Le serveur Web commute entre les différentes langues grâce à l'utilisation d'un cookie appelé "siemens\_automation\_language". Il s'agit d'un cookie défini et activé dans les pages HTML, que le serveur Web interprète pour afficher une page dans la langue appropriée provenant du dossier de langue de même nom. La page HTML doit inclure du code JavaScript pour définir ce cookie à l'un des identificateurs de langue prédéfinis : "de", "en", "es", "fr", "it" ou "zh".

Par exemple, si la page HTML définit le cookie à "de", le serveur Web commute dans le dossier "de" et affiche la page ayant le nom de page HTML par défaut tel que défini dans la configuration STEP 7 (Page 589).

#### Exemple

L'exemple suivant utilise une page HTML par défaut nommée "langswitch.html" dans chaque dossier de langue. Le répertoire HTML contient également un dossier nommé "script". Le dossier script comprend un fichier JavaScript de nom "lang.js". Chaque page langswitch.html utilise ce JavaScript pour définir le cookie de langue, "siemens\_automation\_language".

#### HTML pour "langswitch.html" dans le dossier "en"

L'en-tête de la page HTML définit la langue à "anglais", le jeu de caractères à UTF-8, ainsi que le chemin allant au fichier JavaScript lang.js.

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN">
<html>
<head>
<meta http-equiv="Content-Language" content="en">
<meta http-equiv="Content-Type" content="text/html; charset=utf-8">
<title>Language switching english page</title>
<script type="text/javascript" src="script/lang.js" ></script>
```

Le corps du fichier utilise une liste de sélection permettant à l'utilisateur de choisir entre l'allemand et l'anglais. L'anglais ("en") est la langue sélectionnée par défaut. Lorsque l'utilisateur change de langue, la page appelle la fonction JavaScript DoLocalLanguageChange() avec la valeur de l'option sélectionnée.

```
<!-- Language Selection -->
<table>
  <tr>
    <td align="right" valign="top" nowrap>
      <!-- change language immediately on selection change -->
      <select name="Language"
        onchange="DoLocalLanguageChange(this)"
        size="1">
        <option value="de" >German</option>
        <option value="en" selected >English</option>
      </select>
    </td>
  </tr>
</table><!-- Language Selection End-->
```

## HTML pour "langswitch.html" dans le dossier "de"

L'en-tête pour la page langswitch.html allemande est identique à celui de l'anglais, si ce n'est que la langue définie est l'allemand.

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN">
<html>
<head>
<meta http-equiv="Content-Language" content="de"><meta http-
equiv="Content-Type" content="text/html; charset=utf-8">
<title>Sprachumschaltung Deutsche Seite</title>
<script type="text/javascript" src="script/lang.js" ></script>
</head>
```

Le code HTML dans la page allemande est identique à celui de la page anglaise, si ce n'est que la langue sélectionnée par défaut est l'allemand ("de").

```
<!-- Language Selection -->
<table>
  <tr>
    <td align="right" valign="top" nowrap>
      <!-- change language immediately on change of the selection -
->
      <select name="Language"
        onchange="DoLocalLanguageChange(this)"
        <size="1">
          <option value="de" selected >Deutsch</option>
          <option value="en" >Englisch</option>
        </select>
      </td>
    </tr>
  </table><!-- Language Selection End-->
```

## JavaScript "lang.js" dans le dossier "script"

La fonction "DoLocalLanguageChange()" se trouve dans le fichier lang.js. Cette fonction appelle la fonction "SetLangCookie()", puis recharge la fenêtre qui affiche la page HTML.

La fonction "SetLangCookie()" construit une affectation qui affecte la valeur de la liste sélectionnée au cookie "siemens\_automation\_language" du document. Elle définit également le chemin vers l'application de sorte que la page commutée, et non la page demandeuse, reçoit la valeur du cookie.

En option, la page peut définir une valeur d'expiration pour le cookie dans la section mise en commentaire.

```

function DoLocalLanguageChange(oSelect) {
    SetLangCookie(oSelect.value);
    top.window.location.reload();
}
function SetLangCookie(value) {
    var strval = "siemens_automation_language=";
    // Ceci est le cookie par lequel le serveur Web
    // détecte la langue désirée
    // Ce nom est requis par le serveur Web.
    strval = strval + value;
    strval = strval + "; path=/ ";
    // Définissez le chemin de l'application sinon le chemin
    // serait pris pour la page ayant posé la requête
    // et cette page ne recevrait pas le cookie.
    /* OPTIONAL
    Utilisez une expiration si ce cookie doit vivre plus
longtemps
    que la session en cours du navigateur :
    var now = new Date();
    var endttime = new Date(now.getTime() + expiration);
    strval = strval + "; expires=" +
        endttime.toGMTString() + ";";
    */
    document.cookie = strval;
}

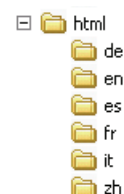
```

### 11.3.9.3 Configuration de STEP 7 pour qu'il utilise une structure de page multilingue

La procédure pour configurer des pages Web personnalisées multilingues est similaire à la procédure générale de configuration de pages Web personnalisées (Page 569). Toutefois, lorsque vous avez créé des dossiers pour les langues, vous définissez votre répertoire HTML comme étant le dossier qui contient les dossiers de langue individuels et non comme étant l'un des dossiers de langue.

Lorsque vous sélectionnez la page HTML par défaut, vous naviguez jusqu'au dossier de langue et vous y sélectionnez la page HTML qui doit servir de page d'accueil. Ensuite, lorsque vous générez les blocs et les chargez dans la CPU, le serveur Web affiche la page d'accueil dans le dossier de langue que vous avez configuré.

Par exemple, si la structure de dossiers présentée ici était dans C:\,



la définition du répertoire HTML serait C:\html et, si l'anglais devait être la page d'affichage initiale, vous navigueriez jusqu'à en\langswitch.html pour y définir la page HTML par défaut.

### 11.3.10 Commande avancée de pages Web personnalisées

Lorsque vous générez les blocs de données pour vos pages Web personnalisées, STEP 7 crée un DB de commande qu'il utilise pour gérer l'interaction avec les pages personnalisées et leur affichage. STEP 7 crée également un jeu de DB de fragment qui représentent les pages individuelles. Dans les circonstances normales, vous n'avez pas besoin de connaître la structure du DB de commande ni de savoir comment le manipuler.

Si vous voulez activer et désactiver une application Web ou manipuler des fragments manuels individuels, vous utilisez les variables du DB de commande et l'instruction WWW.

#### Structure du DB de commande

Le DB de commande est une structure de données étendue à laquelle vous pouvez accéder lors de la programmation de votre programme utilisateur STEP 7. Seules certaines des variables du bloc de données de commande sont décrites ici.

#### Structure Commandstate

"Commandstate" est une structure qui contient des commandes globales et des états globaux pour le serveur Web.

##### Commandes globales dans la structure "Commandstate"

Les commandes globales s'appliquent au serveur Web en général. Vous pouvez désactiver le serveur Web ou le redémarrer à partir des paramètres du DB de commande.

Variable du bloc	Type de données	Description
init	BOOL	Evaluer le DB de commande et initialiser l'application Web
deactivate	BOOL	Désactiver l'application Web

##### Etats globaux dans la structure Commandstate

Les états globaux s'appliquent au serveur Web en général et contiennent des informations d'état sur l'application Web.

Variable du bloc	Type de données	Description
initializing	BOOL	L'application Web lit le DB de commande.
error	BOOL	Initialisation de l'application Web impossible
deactivating	BOOL	L'application Web est en cours de désactivation.
deactivated	BOOL	L'application Web est désactivée.
initialized	BOOL	L'application Web est initialisée.

### Table de requête

La table de requête est un tableau de structures contenant des commandes et des états s'appliquant à des DB de fragment individuels. Si vous avez créé des fragments de type "manuel" avec la commande AWP\_Start\_Fragment (Page 565), le programme utilisateur STEP 7 doit gérer ces pages par le biais du DB de commande. Les états de requête sont en lecture seule et fournissent des informations sur le fragment en cours. Vous utilisez les commandes de requête pour commander le fragment en cours.

Variable du bloc	Type de données	Description
requesttab	ARRAY [ 1 .. 4 ] OF STRUCT	Tableau de structures pour la commande de DB de fragment individuels. Le serveur Web peut traiter jusqu'à quatre fragments à un moment donné. L'indice de tableau pour un fragment particulier est arbitraire lorsque le serveur Web traite plusieurs fragments ou des fragments provenant de plusieurs sessions de navigateur.

### Éléments de la structure requesttab

Variable du bloc	Type de données	Description
page_index	UINT	Numéro de la page Web en cours
fragment_index	UINT	Numéro du fragment en cours. Peut être défini à un fragment différent.
// Commandes de requête		
continue	BOOL	Activer la page/le fragment en cours pour l'émission et poursuivre avec le fragment suivant.
repeat	BOOL	Activer la page/le fragment en cours pour la ré-émission et poursuivre avec le même fragment.
abort	BOOL	Fermer la connexion http sans émission.
finish	BOOL	Envoyer ce fragment ; la page est complète, ne pas traiter de fragments supplémentaires.
// Etats de requête		Les états de requête sont en lecture seule.
idle	BOOL	Rien à faire, mais actif
waiting	BOOL	Le fragment attend d'être activé.
sending	BOOL	Le fragment émet.
aborting	BOOL	L'utilisateur a annulé la requête en cours.

### Fonctionnement

A chaque fois que votre programme modifie le DB de commande, il doit appeler l'instruction WWW avec, comme paramètre, le numéro du DB de commande modifié. Les commandes globales et les commandes de requête prennent effet lorsque le programme utilisateur STEP 7 exécute l'instruction WWW (Page 571).

Le programme utilisateur STEP 7 peut définir l'indice fragment\_index explicitement, le serveur Web traitant alors le fragment spécifié avec une commande de requête. Sinon, le serveur Web traite le fragment en cours pour la page en cours lorsque l'instruction WWW s'exécute.

Les techniques possibles d'utilisation de "fragment\_index" incluent :

- Traiter le fragment en cours : Laissez "fragment\_index" inchangé et activez la commande "continue".
- Sauter le fragment en cours : Définissez "fragment\_index" à 0 et activez la commande "continue".
- Remplacer le fragment en cours par un autre fragment : Définissez "fragment\_index" au nouvel ID de fragment et activez la commande "continue".

Pour vérifier les états globaux ou les états de requête qui auraient pu changer, le programme utilisateur STEP 7 doit appeler l'instruction WWW pour évaluer les valeurs en cours de ces états. Une façon de faire typique serait d'appeler l'instruction WWW périodiquement jusqu'à ce qu'un état spécifique apparaisse.

---

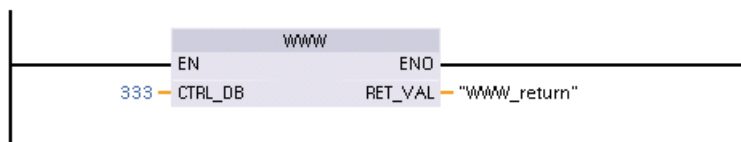
#### Remarque

Si le programme utilisateur STEP 7 active plus d'une commande de requête, l'instruction WWW n'en traite qu'une dans cet ordre : abort, finish, repeat, continue. L'instruction WWW efface toutes les commandes de requête après le traitement.

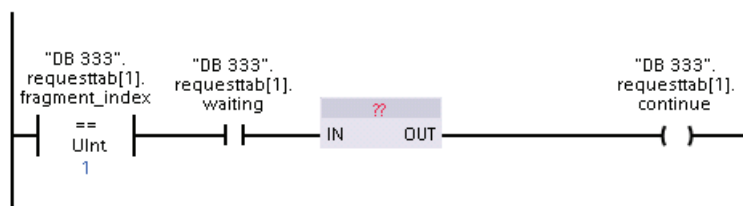
---

### Exemples

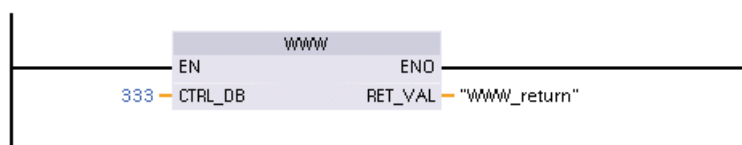
L'exemple suivant montre un programme utilisateur STEP 7 qui vérifie si un fragment d'ID égal à 1 est en état d'attente, après un appel antérieur de l'instruction WWW. Il peut également attendre que d'autres conditions spécifiques de l'application apparaissent. Puis il exécute tout traitement nécessaire pour le fragment, par exemple définir des variables du bloc de données, effectuer des calculs ou d'autres tâches spécifiques de l'application. Ensuite, il met à 1 le memento de "continue" afin que le serveur Web exécute ce fragment.







Lorsque le programme appelle l'instruction WWW avec ce DB de commande modifié, la page Web personnalisée avec ce fragment peut être affichée dans le navigateur Web.



Notez qu'il s'agit d'un exemple simplifié ; le fragment à contrôler pourrait se trouver dans n'importe laquelle des quatre structures requesttab du tableau.

## 11.4 Contraintes

Les considérations informatiques suivantes peuvent influencer sur votre utilisation du serveur Web :

- En règle générale, vous devez utiliser l'adresse IP de la CPU pour accéder aux pages Web standard ou aux pages Web personnalisées. Si votre navigateur Web n'autorise pas la connexion directe à une adresse IP, contactez votre administrateur informatique. Si votre réglementation locale prend en charge le service DNS, vous pouvez vous connecter à l'adresse IP via une entrée DNS désignant cette adresse.
- Les pare-feu, les réglages de proxy et autres restrictions spécifiques du site peuvent également restreindre l'accès à la CPU. Contactez votre administrateur informatique pour résoudre ces problèmes.
- Les pages Web standard utilisent des JavaScripts et des cookies. Si les JavaScripts ou les cookies sont désactivés dans votre navigateur Web, activez-les. Si vous ne pouvez pas les activer, cela limitera certaines fonctionnalités (Page 594). L'utilisation de JavaScripts et de cookies dans les pages Web personnalisées est facultative. S'ils sont utilisés, vous devez les activer dans votre navigateur.
- Le protocole SSL (Secure Sockets Layer) est pris en charge par le serveur Web. Vous pouvez accéder aux pages Web standard et aux pages Web personnalisées avec une URL `http://ww.xx.yy.zz` ou `https://ww.xx.yy.zz`, "ww.xx.yy.zz" représentant l'adresse IP de la CPU.
- Siemens fournit un certificat de sécurité pour sécuriser l'accès au serveur Web. Vous pouvez télécharger et importer ce certificat dans les options Internet de votre navigateur Web (Page 596) à partir de la page Web standard d'introduction (Page 539). Si vous décidez de ne pas importer ce certificat, vous recevrez une invitation à vérifier la sécurité à chaque fois que vous accéderez au serveur Web avec `https://`.

## Nombre de connexions

Le serveur Web accepte 30 connexions HTTP actives au maximum. Ces 30 connexions peuvent être utilisées de diverses façons en fonction du navigateur Web que vous utilisez et du nombre d'objets différents par page (fichiers .css, images, fichiers .html supplémentaires). Certaines connexions persistent pendant l'affichage de la page alors que d'autres sont libérées après la connexion initiale.

Si, par exemple, vous utilisez Mozilla Firefox 8 qui prend en charge six connexions persistantes au maximum, vous pouvez utiliser cinq instances de navigateur ou d'onglet de navigateur avant que le serveur Web ne se mette à couper des connexions. Si une page n'utilise pas les six connexions, vous pouvez avoir des instances de navigateur ou d'onglet de navigateur supplémentaires.

N'oubliez pas que les performances des pages peuvent être affectées par le nombre de connexions actives.

### 11.4.1 Fonctionnalités limitées lorsque JavaScript est désactivé

Les pages Web standard sont implémentées à l'aide de HTML, de JavaScripts et de cookies. Si votre site limite l'utilisation des JavaScripts et des cookies, activez-les afin que les pages fonctionnent correctement. Si vous ne pouvez pas activer les JavaScripts pour votre navigateur Web, les fonctionnalités commandées par les JavaScripts ne peuvent pas s'exécuter.

## Généralités

Les pages ne sont pas actualisées dynamiquement. Vous devez rafraîchir manuellement la page avec le bouton d'actualisation (Page 537) pour afficher les nouvelles données.

## Page Mémoire tampon de diagnostic

- Affichage des détails des événements : Avec JavaScript, vous sélectionnez une ligne dans la mémoire tampon de diagnostic pour voir les détails dans la partie inférieure. Sans JavaScript, vous devez cliquer sur l'hyperlien de la zone d'événements d'une mémoire tampon de diagnostic pour voir les données de l'événement dans la partie inférieure.
- Modification de la plage des entrées de mémoire tampon de diagnostic à visualiser : Avec JavaScript, vous utilisez la liste déroulante située en haut pour sélectionner la plage des entrées de mémoire tampon de diagnostic à visualiser et la page est automatiquement actualisée. Sans JavaScript, vous utilisez la liste déroulante en haut pour sélectionner la plage des entrées de mémoire tampon de diagnostic à visualiser, mais vous devez ensuite cliquer sur le lien "Go" pour mettre à jour la page de mémoire tampon de diagnostic avec la plage que vous avez sélectionnée dans la liste déroulante.

Notez que "Go" et les hyperliens de la zone d'événements ne sont visibles que lorsque JavaScript n'est pas activé. Ils ne sont pas nécessaires et sont donc absents lorsque JavaScript est activé.

---

#### Remarque

Le navigateur Opera V11.0 ne supporte pas le bouton "Go" ou des entrées de diagnostic avec hyperlien. Avec Opera V11.0, vous pouvez accéder aux détails sur les événements ou modifier la plage si vous avez désactivé JavaScript.

---

#### Page Informations sur les modules

- Vous ne pouvez pas filtrer les données.
- Vous ne pouvez pas trier les champs.

#### Page Etat des variables

- Après avoir entré chaque variable, vous devez pointer manuellement dans la colonne "Nouvelle variable" pour entrer une nouvelle variable.
- La sélection d'un format d'affichage ne modifie pas automatiquement l'affichage de la valeur de données au format sélectionné. Vous devez cliquer sur le bouton "Monitor value" pour rafraîchir l'affichage avec le nouveau format.

#### Page des journaux

- Vous ne pouvez pas cliquer sur un nom de fichier sous "Recent entries" pour ouvrir ou enregistrer un fichier journal. Vous pouvez, cependant, utiliser l'icône de téléchargement pour la même fonctionnalité.
- La page Journaux ne se réactualise pas.
- Les boutons "+" et "-" pour augmenter et réduire le nombre d'entrées récentes n'ont aucun effet.
- La saisie directe d'une valeur dans le nombre d'entrées récentes ne modifie pas le nombre d'entrées. Si vous tentez d'entrer une valeur dans ce champ de Mozilla Firefox, l'écran devient blanc. Vous devez resélectionner "Data Logs" dans le volet de navigation pour restaurer l'affichage des Data Logs. Le champ affichant le nombre d'entrées récentes reste inchangé.

Notez que vous pouvez quitter la page Journaux et y revenir pour voir les 25 entrées les plus récentes.

### 11.4.2 Fonctions restreintes lorsque les cookies ne sont pas autorisés

Si votre navigateur Web n'autorise pas les cookies, vous ne pouvez pas vous connecter avec le nom d'utilisateur "admin".

### 11.4.3 Importation du certificat de sécurité Siemens

Vous pouvez importer le certificat de sécurité Siemens dans vos options Internet de manière à ne pas recevoir de vérification de sécurité à chaque fois que vous entrez `https://ww.xx.yy.zz` dans votre navigateur Web, "`ww.xx.yy.zz`" étant l'adresse IP de la CPU. Vous n'avez pas besoin de télécharger et d'installer le certificat si vous utilisez une URL `http://` et non une URL `https://`.

#### Téléchargement du certificat

Utilisez le lien de téléchargement du certificat dans la page Introduction (Page 539) pour télécharger le certificat de sécurité Siemens sur votre PC. La procédure varie selon le navigateur Web que vous utilisez.

[download certificate](#)

#### Importation du certificat dans Internet Explorer

1. Cliquez sur le lien "download certificate" dans la page Introduction. Une boîte de dialogue "Téléchargement de fichier - Avertissement de sécurité" s'affiche.
2. Cliquez sur "Ouvrir" dans cette boîte de dialogue pour ouvrir le fichier. Une boîte de dialogue "Certificate" s'ouvre.
3. Dans cette boîte de dialogue, cliquez sur le bouton "Install Certificate" pour lancer l'assistant d'importation de certificat.
4. Suivez les instructions de l'assistant d'importation de certificat afin d'importer le certificat en laissant le système d'exploitation choisir automatiquement l'emplacement de stockage du certificat.

#### Importation du certificat dans Mozilla Firefox

1. Cliquez sur le lien "download certificate" dans la page Intro. Une boîte de dialogue "Ouverture de MiniWebCA\_Cer.crt" s'affiche.
2. Cliquez sur "Enregistrer le fichier" dans la boîte de dialogue "Ouverture de MiniWebCA\_Cer.crt". Une boîte de dialogue "Téléchargements" apparaît.
3. Dans la boîte de dialogue "Téléchargements", double-cliquez sur "MiniWebCA\_Cer.crt". Si vous avez tenté le téléchargement plus d'une fois, plusieurs copies sont montrées. Il suffit que vous double-cliquiez sur l'une quelconque des entrées "MiniWebCA\_Cer.crt".
4. Cliquez sur "OK" si un message vous demande d'ouvrir un fichier exécutable.
5. Cliquez sur "Ouvrir" dans la boîte de dialogue "Ouverture de fichier - Avertissement de sécurité" si elle apparaît. Une boîte de dialogue "Certificate" s'ouvre.
6. Dans cette boîte de dialogue, cliquez sur le bouton "Install Certificate".
7. Suivez les instructions de l'assistant d'importation de certificat afin d'importer le certificat en laissant le système d'exploitation choisir automatiquement l'emplacement de stockage du certificat.
8. Si la boîte de dialogue "Avertissement de sécurité" apparaît, cliquez sur "Oui" pour confirmer l'installation du certificat.

## Autres navigateurs

Suivez les conventions de votre navigateur Web pour importer et installer le certificat Siemens.

Une fois le certificat de sécurité Siemens "S7-Controller Family" installé dans les options Internet pour le contenu de votre navigateur Web, il n'y aura plus de demande de vérification par un message de sécurité lorsque vous accéderez au serveur Web via `https://ww.xx.yy.zz`.

### 11.4.4 Importation de journaux en format CSV dans des versions non anglaises/américaines de Microsoft Excel

Les fichiers journaux sont en format CSV (valeurs séparées par des virgules). Vous pouvez ouvrir ces fichiers directement dans Excel à partir de la page Journaux lorsque votre système utilise la version anglaise ou américaine d'Excel. Toutefois, ce format n'est pas très représenté dans d'autres pays, car les virgules sont utilisées dans la notation numérique.

Pour les versions d'Excel autres que la version anglaise/américaine, procédez comme suit pour ouvrir un fichier journal que vous avez enregistré :

1. Ouvrez Excel et créez un classeur vide.
2. Dans le menu "Données > Données externes", sélectionnez la commande "Importer des données".
3. Naviguez jusqu'au fichier journal que vous voulez ouvrir et sélectionnez-le. L'Assistant Importation de texte est lancé.
4. Dans l'Assistant Importation de texte, changez l'option par défaut pour "Type de données d'origine" de "Largeur fixe" à "Délimité".
5. Cliquez sur le bouton Suivant.
6. Dans la boîte de dialogue Etape 2, cochez la case "Virgule" pour changer le type de séparateur de "Tabulation" à "Virgule".
7. Cliquez sur le bouton Suivant.
8. Dans la boîte de dialogue Etape 3, vous pouvez facultativement changer le format des données de MJA (mois/jour/année) à un autre format.
9. Exécutez les étapes restantes de l'Assistant Importation de texte pour importer le fichier.



## Processeur de communication

### 12.1 Utilisation des interfaces de communication série

Deux modules de communication (CM) et un Communication Board (CB) fournissent l'interface pour les communications PtP :

- CM 1241 RS232 (Page 875)
- CM 1241 RS422/485 (Page 877)
- CB 1241 RS485 (Page 873)

Vous pouvez connecter jusqu'à trois CM (de tout type) plus une CB pour un total de quatre interfaces de communication. Installez le CM à gauche de la CPU ou d'un autre CM. Installez la CB sur le devant de la CPU. Reportez-vous au chapitre "Installation" du Manuel Système S7-1200 (Page 57) pour des instructions détaillées sur l'installation et la désinstallation de modules.

Les interfaces de communication série ont les caractéristiques suivantes :

- Disposent d'un port isolé
- Prennent en charge les protocoles Point-à-Point
- Sont configurés et programmés par des instructions et des fonctions bibliothèque étendues
- Affichent l'activité de transmission et réception au moyen de DEL
- Affiche une DEL de diagnostic (CM uniquement)
- Sont alimentés par la CPU : connexion à une alimentation externe inutile

Reportez-vous aux caractéristiques techniques pour les interfaces de communication (Page 864).

#### DEL de signalisation

Les modules de communication comportent trois DEL de signalisation :

- DEL de diagnostic (DIAG) : Cette DEL clignote en rouge jusqu'à ce que la CPU y accède. Une fois la CPU sous tension, elle recherche les CM et y accède. La DEL de diagnostic commence à clignoter en vert. Cela signifie que la CPU a accédé au CM, mais ne lui a pas encore fourni la configuration. La CPU charge la configuration dans les CM configurés lorsque le programme est chargé dans la CPU. Après un chargement dans la CPU, la DEL de diagnostic du module de communication devrait émettre une lumière verte continue.
- DEL de transmission (Tx) : La DEL de transmission s'allume lorsque des données sont émises depuis le port de communication.
- DEL de réception (Rx) : Cette DEL s'allume lorsque le port de communication reçoit des données.

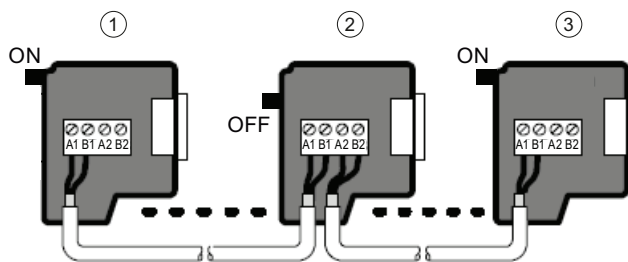
Le Communication Board présente des DEL de transmission (TxD) et de réception (RxD). Il n'y a pas de DEL de diagnostic.

## 12.2 Polarisation et terminaison d'un connecteur de réseau RS485

Siemens propose un connecteur de réseau RS485 (Page 888) vous permettant de connecter aisément plusieurs appareils à un réseau RS485. Ce connecteur comporte deux jeux de bornes vous permettant de connecter les câbles de réseau entrants et sortants. Ce connecteur présente également des commutateurs pour la polarisation et la terminaison sélectives du réseau.

### Remarque

La terminaison et la polarisation ne concernent que les deux extrémités du réseau RS485. Il n'y a pas de terminaison ni de polarisation des appareils situés entre les deux appareils d'extrémité. Blindage de câble nu : 12 mm environ doivent être en contact avec les guides métalliques de tous les emplacements.

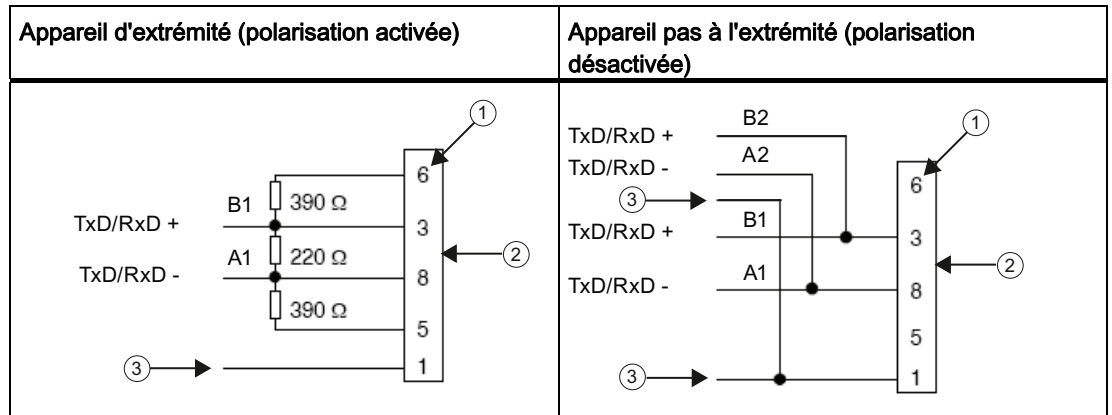


- ① Position du commutateur = ON : terminaison et polarisation
- ② Position du commutateur = OFF : ni terminaison ni polarisation
- ③ Position du commutateur = ON : terminaison et polarisation



12.2 Polarisation et terminaison d'un connecteur de réseau RS485

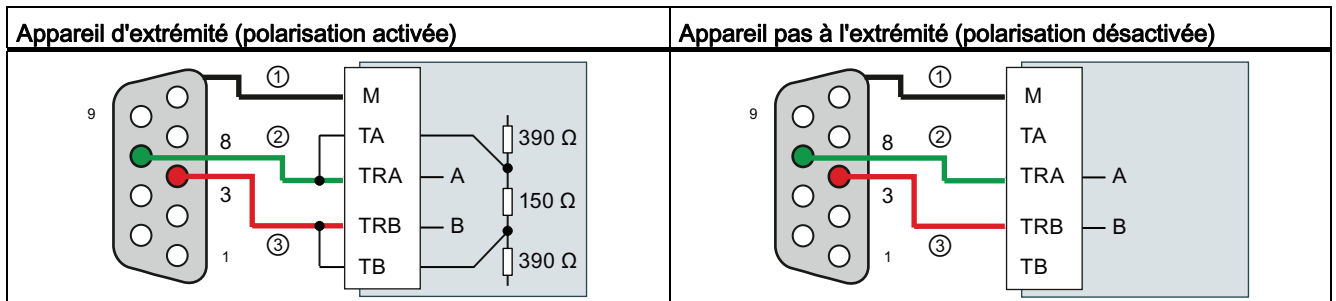
Tableau 12- 1 Terminaison et polarisation pour le connecteur RS485



- ① Numéro de broche
- ② Connecteur de réseau
- ③ Blindage du câble

Le CB 1241 comporte des résistances internes pour la terminaison et la polarisation du réseau. Pour activer la terminaison et la polarisation de la liaison, raccordez TRA à TA et TRB à TB pour inclure les résistances internes dans le circuit. Le CB 1241 n'a pas de connecteur à 9 broches. Le tableau suivant montre les connexions à un connecteur à 9 broches sur le partenaire de communication.

Tableau 12- 2 Terminaison et polarisation pour le CB 1241

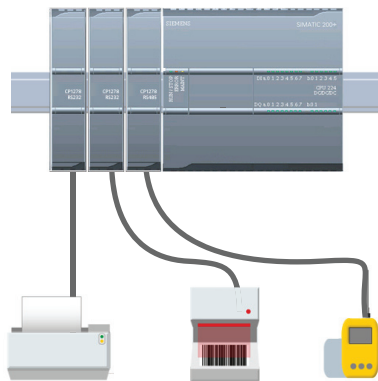


- ① Connectez M au blindage du câble
- ② A = TxD/RxD - (fil vert / broche 8)
- ③ B = TxD/RxD + (fil rouge / broche 3)

## 12.3 Communication point à point (PtP)

La CPU prend en charge la communication point à point (PtP) suivante pour les protocoles série en mode caractères. La communication point à point offre une liberté et une souplesse maximales, mais requiert une implémentation importante dans le programme utilisateur.

- PtP (Page 602)
- USS (Page 646)
- Modbus (Page 663)



La communication point à point offre de nombreuses possibilités :

- Possibilité d'envoyer des informations directement à un périphérique externe tel qu'une imprimante
- Possibilité de recevoir des informations d'autres appareils, tels que lecteurs de codes à barres, lecteurs RFID, systèmes de caméra ou de vision tiers et nombreux autres types d'appareils
- Possibilité d'échanger des informations, avec émission et réception de données, avec d'autres appareils tels que des appareils GPS, des systèmes de caméra ou de vision tiers, des modems radio et bien d'autres

La communication point à point est une communication série qui utilise des UART standard pour prendre en charge des débits et parités variés. Les modules de communication RS232 et RS422/485 et le Communication Board RS485 fournissent les interfaces électriques pour l'exécution de la communication point à point.

## 12.3.1 Instructions point à point

### 12.3.1.1 Paramètres communs pour les instructions point à point

Tableau 12- 3 Paramètres d'entrée communs pour les instructions point à point

Paramètre	Description
REQ	<p>De nombreuses instructions point à point utilisent l'entrée REQ pour déclencher l'opération en présence d'une transition du niveau bas au niveau haut. L'entrée REQ doit être au niveau haut (VRAI) pour une exécution de l'instruction, mais REQ peut rester à VRAI aussi longtemps que désiré. L'instruction ne déclenche pas de nouvelle opération tant qu'elle n'a pas été exécutée avec l'entrée REQ à FAUX de manière à ce qu'elle puisse réinitialiser l'historique de l'entrée REQ. Cela est nécessaire pour que l'instruction puisse à nouveau détecter une transition du niveau bas au niveau haut afin de déclencher l'opération suivante.</p> <p>Lorsque vous insérez une instruction point à point dans votre programme, STEP 7 vous demande d'identifier le DB d'instance. Utilisez un DB unique pour chaque appel de l'instruction point à point. Cela garantit que chaque instruction gère correctement les entrées telles que REQ.</p>
PORT	<p>Une adresse de port est affectée pendant la configuration de l'appareil de communication. Après la configuration, un nom symbolique de port par défaut peut être sélectionné dans la liste déroulante d'aide pour les paramètres. La valeur de port CM ou CB affectée est la propriété de configuration d'appareil "identificateur matériel". Le nom symbolique du port est défini dans l'onglet "Constantes" de la table de variables API.</p>
Résolution en temps bit	<p>Plusieurs paramètres sont indiqués en nombre de temps bit à la vitesse de transmission configurée. Indiquer le paramètre en temps bit permet d'avoir un paramètre indépendant de la vitesse de transmission. Tous les paramètres en unités temps bit peuvent avoir une valeur maximale de 65535. Toutefois, la quantité de temps maximale que peut mesurer un CM ou un CB est de huit secondes.</p>

Les paramètres de sortie DONE, NDR, ERROR et STATUS des instructions point à point indiquent l'état d'achèvement de l'exécution des opérations point à point.

Tableau 12- 4 Paramètres de sortie DONE, NDR, ERROR et STATUS

Paramètre	Type de données	Valeur par défaut	Description
DONE	Bool	FAUX	A VRAI pour un cycle pour indiquer que la dernière demande s'est achevée sans erreur. FAUX sinon.
NDR	Bool	FAUX	A VRAI pour un cycle pour indiquer que la dernière action demandée s'est achevée sans erreur et que de nouvelles données ont été reçues. FAUX sinon.

12.3 Communication point à point (PtP)

Paramètre	Type de données	Valeur par défaut	Description
ERROR	Bool	FAUX	A VRAI pour un cycle pour indiquer que la dernière demande s'est achevée avec des erreurs, le code d'erreur correspondant étant contenu dans STATUS. FAUX sinon.
STATUS	Word	0	Etat du résultat : <ul style="list-style-type: none"> <li>• Si le bit DONE ou NDR est à 1, STATUS est mis à 0 ou reçoit un code informatif.</li> <li>• Si le bit ERROR est à 1, STATUS aura la valeur du code d'erreur.</li> <li>• Si aucun des bits ci-dessus n'est à 1, l'instruction renvoie des résultats d'état qui décrivent l'état en cours de la fonction.</li> </ul> STATUS conserve sa valeur pendant la durée de l'exécution de la fonction.

**Remarque**

Notez que DONE, NDR et ERROR sont à 1 pour un cycle uniquement. La logique de votre programme doit sauvegarder les valeurs temporaires de l'état de sortie dans des verrous de données de manière à ce que vous puissiez détecter des changements d'état lors des cycles suivants du programme.

Tableau 12- 5 Codes d'erreur communs

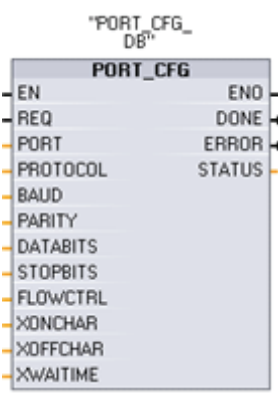
STATUS (W#16#....)	Description
0000	Pas d'erreur
7000	La fonction n'est pas occupée.
7001	La fonction est occupée avec le premier appel.
7002	La fonction est occupée avec les appels suivants (interrogations après le premier appel).
8x3A	Pointeur illicite dans le paramètre x
8070	Toute la mémoire d'instance interne est occupée, trop d'instructions concurrentes en cours.
8080	Numéro de port illicite
8081	Dépassement du délai d'attente, erreur de module ou autre erreur interne
8082	Echec du paramétrage car un paramétrage est en cours en arrière-plan.
8083	Débordement de la mémoire tampon : Le CM ou le CB a renvoyé un message reçu dont la longueur est supérieure à la longueur autorisée.
8090	Erreur interne : Longueur de message erronée, sous-module erroné ou message illicite Contactez l'assistance client.
8091	Erreur interne : Version erronée dans le message de paramétrage Contactez l'assistance client.
8092	Erreur interne : Longueur d'enregistrement erronée dans le message de paramétrage Contactez l'assistance client.

Tableau 12- 6 Classes d'erreurs communes

Description de la classe	Classes d'erreurs	Description
Configuration du port	80Ax	Utilisé pour définir les erreurs de configuration de port communes
Configuration d'émission	80Bx	Utilisé pour définir les erreurs de configuration d'émission communes
Configuration de réception	80Cx	Utilisé pour définir les erreurs de configuration de réception communes
Exécution de l'émission	80Dx	Utilisé pour définir les erreurs communes apparaissant à l'exécution de l'émission
Exécution de la réception	80Ex	Utilisé pour définir les erreurs communes apparaissant à l'exécution de la réception
Gestion des signaux	80Fx	Utilisé pour définir les erreurs communes associées à toute la gestion des signaux

### 12.3.1.2 Instruction PORT\_CFG

Tableau 12- 7 Instruction PORT\_CFG (Configuration de port)

CONT/LOG	SCL	Description
 <p>The diagram shows the PORT_CFG instruction with the following fields:</p> <ul style="list-style-type: none"> <li>EN (Enable)</li> <li>ENO (Enable Output)</li> <li>REQ (Request)</li> <li>DONE (Done)</li> <li>PORT (Port)</li> <li>ERROR (Error)</li> <li>PROTOCOL (Protocol)</li> <li>STATUS (Status)</li> <li>BAUD (Baud rate)</li> <li>PARITY (Parity)</li> <li>DATABITS (Data bits)</li> <li>STOPBITS (Stop bits)</li> <li>FLOWCTRL (Flow control)</li> <li>XONCHAR (XON character)</li> <li>XOFFCHAR (XOFF character)</li> <li>XWAITIME (Wait time)</li> </ul>	<pre>"PORT_CFG_DB" (   REQ:=_bool_in_,   PORT:=_uint_in_,   PROTOCOL:=_uint_in_,   BAUD:=_uint_in_,   PARITY:=_uint_in_,   DATABITS:=_uint_in_,   STOPBITS:=_uint_in_,   FLOWCTRL:=_uint_in_,   XONCHAR:=_char_in_,   XOFFCHAR:=_char_in_,   WAITTIME:=_uint_in_,   DONE=&gt;_bool_out_,   ERROR=&gt;_bool_out_,   STATUS=&gt;_word_out_);</pre>	<p>PORT_CFG vous permet de modifier les paramètres du port, tels que la vitesse de transmission, à partir de votre programme.</p> <p>Vous pouvez définir la configuration statique initiale du port dans les propriétés de la configuration des appareils ou simplement utiliser les valeurs par défaut. Vous pouvez exécuter l'instruction PORT_CFG dans votre programme pour modifier la configuration.</p>

1 STEP 7 crée automatiquement le DB lorsque vous insérez l'instruction.

Les modifications de configuration de PORT\_CFG ne sont pas stockées de façon permanente dans la CPU. Les paramètres configurés dans la configuration des appareils sont restaurés lorsque la CPU passe de l'état MARCHE à l'état ARRET et après une mise hors tension puis sous tension. Reportez-vous à Configuration des ports de communication (Page 622) et à Gestion du contrôle de flux (Page 624) pour plus d'informations.

12.3 Communication point à point (PtP)

Tableau 12- 8 Types de données pour les paramètres

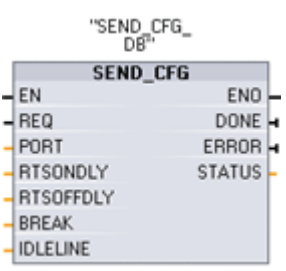
Paramètre et type		Type de données	Description
REQ	IN	Bool	Activation de la modification de configuration en cas de front montant de cette entrée (valeur par défaut : False)
PORT	IN	PORT	Une fois que vous avez installé et configuré un appareil de communication CM ou CB, l'identificateur de port apparaît dans la liste déroulante d'aide pour les paramètres disponible au niveau du connecteur PORT de la boîte. La valeur de port CM ou CB affectée est la propriété de configuration d'appareil "identificateur matériel". Le nom symbolique du port est défini dans l'onglet "Constantes système" de la table de variables API. (valeur par défaut : 0)
PROTOCOL	IN	UInt	0 : protocole de communication point à point (valeur par défaut) 1..n : définition future de protocoles spécifiques
BAUD	IN	UInt	Vitesse de transmission du port (valeur par défaut : 0): 1 = 300 bauds, 2 = 600 bauds, 3 = 1200 bauds, 4 = 2400 bauds, 5 = 4800 bauds, 6 = 9600 bauds, 7 = 19200 bauds, 8 = 38400 bauds, 9 = 57600 bauds, 10 = 76800 bauds, 11 = 115200 bauds
PARITY	IN	UInt	Parité du port (valeur par défaut : 0): 1 = Sans parité, 2 = Parité paire, 3 = Parité impaire, 4 = Parité marque, 5 = Parité d'espace
DATABITS	IN	UInt	Bits par caractère (valeur par défaut) : 1 = 8 bits de données, 2 = 7 bits de données
STOPBITS	IN	UInt	Bits d'arrêt (valeur par défaut : 0): 1 = 1 bit d'arrêt, 2 = 2 bits d'arrêt
FLOWCTRL	IN	UInt	Contrôle du flux (valeur par défaut : 0): 1 = Pas de contrôle de flux, 2 = XON/XOFF, 3 = Matériel RTS toujours activé, 4 = Matériel RTS commuté
XONCHAR	IN	Char	Indique le caractère qui est utilisé comme caractère XON. Ce sera typiquement le caractère DC1 (11H). Ce paramètre n'est évalué que si le contrôle de flux est activé. (valeur par défaut : 0)
XOFFCHAR	IN	Char	Indique le caractère qui est utilisé comme caractère XOFF. Ce sera typiquement le caractère DC3 (13H). Ce paramètre n'est évalué que si le contrôle de flux est activé. (valeur par défaut : 0)
XWAITIME	IN	UInt	Indique combien de temps attendre un caractère XON après avoir reçu un caractère XOFF ou combien de temps attendre le signal CTS après avoir activé RTS (0 à 65535 ms). Ce paramètre n'est évalué que si le contrôle de flux est activé. (valeur par défaut : 2000)
DONE	OUT	Bool	TRUE pour un cycle lorsque la dernière demande s'est achevée sans erreur.
ERROR	OUT	Bool	TRUE pour un cycle lorsque la dernière demande s'est achevée avec une erreur.
STATUS	OUT	Word	Code d'erreur d'exécution (valeur par défaut : 0)

Tableau 12- 9 Codes d'erreur

STATUS (W#16#....)	Description
80A0	Le protocole spécifique n'existe pas.
80A1	La vitesse de transmission spécifique n'existe pas.
80A2	La parité spécifique n'existe pas.
80A3	Le nombre spécifique de bits de données n'existe pas.
80A4	Le nombre spécifique de bits d'arrêt n'existe pas.
80A5	Le type spécifique de contrôle de flux n'existe pas.
80A6	Le temps d'attente est 0 et le contrôle de flux est activé.
80A7	XON et XOFF sont des valeurs illicites (par exemple, la même valeur).

### 12.3.1.3 Instruction SEND\_CFG

Tableau 12- 10 Instruction SEND\_CFG (Configuration d'émission)

CONT/LOG	SCL	Description
	<pre>"SEND_CFG_DB" (     REQ:=_bool_in_,     PORT:=_uint_in_,     RTSONDLY:=_uint_in_,     RTSOFFDLY:=_uint_in_,     BREAK:=_uint_in_,     IDLELINE:=_uint_in_,     DONE=&gt;_bool_out_,     ERROR=&gt;_bool_out_,     STATUS=&gt;_word_out_);</pre>	<p>SEND_CFG permet la configuration dynamique de paramètres d'émission série pour un port de communication point à point. Tout message mis en file d'attente à l'intérieur d'un CM ou d'un CB est rejeté lorsque SEND_CFG est exécuté.</p>

<sup>1</sup> STEP 7 crée automatiquement le DB lorsque vous insérez l'instruction.

Vous pouvez définir la configuration statique initiale du port dans les propriétés de la configuration des appareils ou simplement utiliser les valeurs par défaut. Vous pouvez exécuter l'instruction SEND\_CFG dans votre programme pour modifier la configuration.

Les modifications de configuration de SEND\_CFG ne sont pas stockées de façon permanente dans la CPU. Les paramètres configurés dans la configuration des appareils sont restaurés lorsque la CPU passe de l'état MARCHE à l'état ARRET et après une mise hors tension puis sous tension. Voir Configuration des paramètres d'émission (Page 625).

12.3 Communication point à point (PtP)

Tableau 12- 11 Types de données pour les paramètres

Paramètre et type		Type de données	Description
REQ	IN	Bool	Activation de la modification de configuration en cas de front montant de cette entrée (valeur par défaut : False)
PORT	IN	PORT	Une fois que vous avez installé et configuré un appareil de communication CM ou CB, l'identificateur de port apparaît dans la liste déroulante d'aide pour les paramètres disponible au niveau du connecteur PORT de la boîte. La valeur de port CM ou CB affectée est la propriété de configuration d'appareil "identificateur matériel". Le nom symbolique du port est défini dans l'onglet "Constantes système" de la table de variables API. (valeur par défaut : 0)
RTSONDLY	IN	UInt	Nombre de millisecondes à attendre après activation de RTS avant toute émission de données d'émission. Ce paramètre n'est valable que si le contrôle de flux matériel est activé. La plage valide va de 0 à 65535 ms. La valeur 0 désactive la fonction. (valeur par défaut : 0)
RTSOFFDLY	IN	UInt	Nombre de millisecondes à attendre avant de désactiver RTS après l'émission des données d'émission. Ce paramètre n'est valable que si le contrôle de flux matériel est activé. La plage valide va de 0 à 65535 ms. La valeur 0 désactive la fonction. (valeur par défaut : 0)
BREAK	IN	UInt	Ce paramètre indique qu'une pause du nombre indiqué de temps bit sera envoyée au début de chaque message. La valeur maximale est 65535 temps bit, avec un maximum de huit secondes. La valeur 0 désactive la fonction. (valeur par défaut : 12)
IDLELINE	IN	UInt	Ce paramètre indique que la ligne restera inactive pendant le nombre indiqué de temps bit avant le début de chaque message. La valeur maximale est 65535 temps bit, avec un maximum de huit secondes. La valeur 0 désactive la fonction. (valeur par défaut : 12)
DONE	OUT	Bool	TRUE pour un cycle lorsque la dernière demande s'est achevée sans erreur.
ERROR	OUT	Bool	TRUE pour un cycle lorsque la dernière demande s'est achevée avec une erreur.
STATUS	OUT	Word	Code d'erreur d'exécution (valeur par défaut : 0)

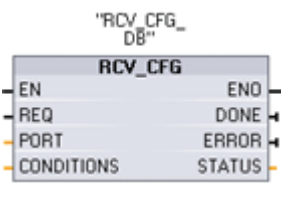
Tableau 12- 12 Codes d'erreur

STATUS (W#16#...)	Description
80B0	Configuration d'alarme d'émission non autorisée. Contactez l'assistance client.
80B1	Le temps de pause est supérieur à la valeur maximale autorisée.
80B2	Le temps d'inactivité est supérieur à la valeur maximale autorisée.



### 12.3.1.4 Instruction RCV\_CFG

Tableau 12- 13 Instruction RCV\_CFG (Configuration de réception)

CONT/LOG	SCL	Description
 <p>The diagram shows a block labeled "RCV_CFG" with a sub-label "DB". It has eight input/output ports: EN (input), REQ (input), PORT (input), CONDITIONS (input), END (output), DONE (output), ERROR (output), and STATUS (output).</p>	<pre>"RCV_CFG_DB" (     REQ:=_bool_in_,     PORT:=_uint_in_,     CONDITIONS:=_struct_in_,     DONE=&gt;_bool_out_,     ERROR=&gt;_bool_out_,     STATUS=&gt;_word_out_);</pre>	<p>RCV_CFG procède à la configuration dynamique de paramètres de réception série pour un port de communication point à point. Cette instruction configure les conditions qui signalent le début et la fin d'un message reçu. Tout message mis en file d'attente à l'intérieur d'un CM ou d'un CB est rejeté lorsque RCV_CFG est exécuté.</p>

<sup>1</sup> STEP 7 crée automatiquement le DB lorsque vous insérez l'instruction.

Vous pouvez définir la configuration statique initiale du port de communication dans les propriétés de la configuration des appareils ou simplement utiliser les valeurs par défaut. Vous pouvez exécuter l'instruction RCV\_CFG dans votre programme pour modifier la configuration.

Les modifications de configuration de RCV\_CFG ne sont pas stockées de façon permanente dans la CPU. Les paramètres configurés dans la configuration des appareils sont restaurés lorsque la CPU passe de l'état MARCHE à l'état ARRÊT et après une mise hors tension puis sous tension. Reportez-vous à Configuration des paramètres de réception (Page 625) pour plus d'informations.

Tableau 12- 14 Types de données pour les paramètres

Paramètre et type		Type de données	Description
REQ	IN	Bool	Activation de la modification de configuration en cas de front montant de cette entrée (valeur par défaut : False)
PORT	IN	PORT	Une fois que vous avez installé et configuré un appareil de communication CM ou CB, l'identificateur de port apparaît dans la liste déroulante d'aide pour les paramètres disponible au niveau du connecteur PORT de la boîte. La valeur de port CM ou CB affectée est la propriété de configuration d'appareil "identificateur matériel". Le nom symbolique du port est défini dans l'onglet "Constantes système" de la table de variables API. (valeur par défaut : 0)
CONDITIONS	IN	CONDITIONS	La structure de données Conditions indique les conditions de début et de fin de message comme décrit ci-dessous.
DONE	OUT	Bool	TRUE pour un cycle lorsque la dernière demande s'est achevée sans erreur.
ERROR	OUT	Bool	TRUE pour un cycle lorsque la dernière demande s'est achevée avec une erreur.
STATUS	OUT	Word	Code d'erreur d'exécution (valeur par défaut : 0)

### Conditions de début pour l'instruction RCV\_PTP

L'instruction RCV\_PTP utilise la configuration indiquée par l'instruction RCV\_CFG pour déterminer le début et la fin de messages de communication point à point. Le début d'un message est déterminé par les conditions de début. Le début d'un message peut être déterminé par une condition de début ou par une combinaison de conditions de début. Si plusieurs conditions de début sont spécifiées, toutes les conditions doivent être satisfaites avant que le message ne commence.

Reportez-vous à la rubrique "Configuration des paramètres de réception (Page 627)" pour une description des conditions de début de message.

### Structure pour le paramètre CONDITIONS, partie 1 (conditions de début)

Tableau 12- 15 Structure CONDITIONS pour les conditions de début

Paramètre et type		Type de données	Description
STARTCOND	IN	UInt	Définit la condition de début (valeur par défaut : 1) <ul style="list-style-type: none"> <li>• 01H : Caractère de début</li> <li>• 02H : Caractère quelconque</li> <li>• 04H : Pause</li> <li>• 08H : Ligne inactive</li> <li>• 10H : Séquence 1</li> <li>• 20H : Séquence 2</li> <li>• 40H : Séquence 3</li> <li>• 80H : Séquence 4</li> </ul>
IDLETIME	IN	UInt	Nombre de temps bit requis pour un délai de ligne inactive. (valeur par défaut : 40). Utilisé uniquement avec une condition de ligne inactive. 0 à 65535
STARTCHAR	IN	Byte	Caractère de début utilisé avec la condition caractère de début (valeur par défaut : B#16#2)
STRSEQ1CTL	IN	Byte	Contrôle ignorer/comparer pour chaque caractère de la séquence 1 : (valeur par défaut : B#16#0) Il s'agit des bits d'activation pour chaque caractère dans la séquence de début. <ul style="list-style-type: none"> <li>• 01H : Caractère 1</li> <li>• 02H : Caractère 2</li> <li>• 04H : Caractère 3</li> <li>• 08H : Caractère 4</li> <li>• 10H : Caractère 5</li> </ul> Désactiver le bit associé à un caractère signifie que n'importe quel caractère conviendra dans cette position de la séquence.
STRSEQ1	IN	Char[5]	Caractères de début de la séquence 1 (5 caractères). Valeur par défaut : 0
STRSEQ2CTL	IN	Byte	Contrôle ignorer/comparer pour chaque caractère de la séquence 2. Valeur par défaut : B#16#0

Paramètre et type		Type de données	Description
STRSEQ2	IN	Char[5]	Caractères de début de la séquence 2 (5 caractères). Valeur par défaut : 0
STRSEQ3CTL	IN	Byte	Contrôle ignorer/comparer pour chaque caractère de la séquence 3. Valeur par défaut : B#16#0
STRSEQ3	IN	Char[5]	Caractères de début de la séquence 3 (5 caractères). Valeur par défaut : 0
STRSEQ4CTL	IN	Byte	Contrôle ignorer/comparer pour chaque caractère de la séquence 4. Valeur par défaut : B#16#0
STRSEQ4	IN	Char[5]	Caractères de début de la séquence 4 (5 caractères). Valeur par défaut : 0

### Exemple

Considérons le message reçu suivant codé en hexadécimal : "68 10 aa 68 bb 10 aa 16" et les séquences de début configurées présentées dans le tableau ci-après. Les séquences de début commencent à être évaluées lorsque la réception du premier caractère 68H a abouti. Après réception du quatrième caractère (le second 68H), la condition de début 1 est satisfaite. Une fois les conditions de début satisfaites, l'évaluation des conditions de fin commence.

Le traitement de la séquence de début peut être interrompu en raison de diverses erreurs de parité, de trame ou de dépassement de délai inter-caractère. Si ces erreurs se produisent, aucun message n'est reçu puisque la condition de début n'était pas satisfaite.

Tableau 12- 16 Conditions de début

Condition de début	Premier caractère	Premier caractère +1	Premier caractère +2	Premier caractère +3	Premier caractère +4
1	68H	xx	xx	68H	xx
2	10H	aaH	xx	xx	xx
3	dcH	aaH	xx	xx	xx
4	e5H	xx	xx	xx	xx

### Conditions de fin pour l'instruction RCV\_PTP

La fin d'un message est déterminée par l'indication de conditions de fin. La fin d'un message est déterminée par la première occurrence d'une ou de plusieurs conditions de fin configurées. Le paragraphe "Conditions de fin de message" dans la rubrique "Configuration des paramètres de réception (Page 627)" décrit les conditions de fin que vous pouvez configurer dans l'instruction RCV\_CFG.

Vous pouvez configurer les conditions de fin soit dans les propriétés de l'interface de communication dans la configuration des appareils, soit dans l'instruction RCV\_CFG. Les paramètres de réception (conditions de début et conditions de fin) reprennent leur valeur de la configuration des appareils à chaque fois que la CPU passe de l'état ARRÊT à l'état MARCHÉ. Si le programme utilisateur STEP 7 exécute RCV\_CFG, les paramétrages sont modifiés pour être conformes aux conditions de RCV\_CFG.

**Structure pour le paramètre CONDITIONS, partie 2 (conditions de fin)**

Tableau 12- 17 Structure CONDITIONS pour les conditions de fin

Paramètre	Type de paramètre	Type de données	Description
ENDCOND	IN	UInt 0	Ce paramètre indique la condition de fin de message : <ul style="list-style-type: none"> <li>• 01H : Temps de réponse</li> <li>• 02H : Temps de message</li> <li>• 04H : Délai inter-caractères</li> <li>• 08H : Longueur maximale</li> <li>• 10H : N + Lg + M</li> <li>• 20H : Séquence</li> </ul>
MAXLEN	IN	UInt 1	Longueur de message maximale. Utilisé uniquement lorsque la condition de fin longueur maximale est sélectionnée. 1 à 1024 octets
N	IN	UInt 0	Position d'octet du champ de longueur à l'intérieur du message. Utilisé uniquement avec la condition de fin N + Lg + M. 1 à 1022 octets
LENGTHSIZE	IN	UInt 0	Taille du champ d'octet (1, 2 ou 4 octets). Utilisé uniquement avec la condition de fin N + Lg + M.
LENGTHM	IN	UInt 0	Indiquez le nombre de caractères suivant le champ de longueur qui ne sont pas inclus dans la valeur du champ de longueur. Utilisé uniquement avec la condition de fin N + Lg + M. 0 à 255 octets
RCVTIME	IN	UInt 200	Indiquez combien de temps attendre le premier caractère à recevoir. Il sera mis fin à l'opération de réception avec une erreur si aucune réception de caractère n'aboutit pendant le temps indiqué. Utilisé uniquement avec la condition temps de réponse. (0 à 65535 temps bit, avec un maximum de 8 secondes)  Ce paramètre n'est pas une condition de fin de message puisque l'évaluation s'achève lors de la réception du premier caractère d'une réponse. Ce n'est une condition de fin que dans la mesure où il met fin à une opération de réception car aucune réponse n'est reçue alors qu'une réponse est attendue. Vous devez sélectionner une condition de fin distincte.
MSGTIME	IN	UInt 200	Indiquez combien de temps attendre pour que le message entier soit complètement reçu une fois le premier caractère reçu. Ce paramètre est utilisé uniquement lorsque la condition dépassement du délai d'attente de message est sélectionnée. (0 à 65535 millisecondes)
CHARGAP	IN	UInt 12	Indiquez le nombre de temps bit entre caractères. Si le nombre de temps bit entre caractères dépasse la valeur indiquée, la condition de fin est satisfaite. Utilisé uniquement avec la condition délai inter-caractères. (0 à 65535 temps bit, avec un maximum de 8 secondes)

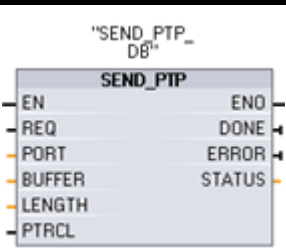
Paramètre	Type de paramètre	Type de données	Description
ENDSEQ1CTL	IN	Byte B#16#0	Contrôle ignorer/comparer pour chaque caractère de la séquence 1 : Il s'agit des bits d'activation pour chaque caractère dans la séquence de fin. Le caractère 1 est le bit 0, le caractère 2 le bit 1, ..., le caractère 5 le bit 4. Désactiver le bit associé à un caractère signifie que n'importe quel caractère conviendra dans cette position de la séquence.
ENDSEQ1	IN	Char[5] 0	Caractères de début de la séquence 1 (5 caractères)

Tableau 12- 18 Codes d'erreur

STATUS (W#16#....)	Description
80C0	Condition de début illicite sélectionnée
80C1	Condition de fin illicite sélectionnée, pas de condition de fin sélectionnée
80C2	Alarme de réception activée ce qui n'est pas possible
80C3	Condition de fin longueur maximale activée et longueur maximale est 0 ou > 1024.
80C4	Longueur calculée activée et N >= 1023
80C5	Longueur calculée activée et longueur différente de 1, 2 ou 4
80C6	Longueur calculée activée et M > 255
80C7	Longueur calculée activée et longueur calculée > 1024
80C8	Dépassement du délai d'attente de la réponse activé et délai d'attente de réponse est zéro.
80C9	Dépassement du délai inter-caractères activé et délai est 0.
80CA	Dépassement du délai de ligne inactive activé et délai est 0.
80CB	Séquence de fin activée mais tous les caractères sont indifférents.
80CC	Séquence de début (l'une quelconque parmi 4) activée mais tous les caractères sont indifférents.

### 12.3.1.5 Instruction SEND\_PTP

Tableau 12- 19 Instruction SEND\_PTP (Emission de données point à point)

CONT/LOG	SCL	Description
	<pre>"SEND_PTP_DB" (   REQ:=_bool_in_,   PORT:=_uint_in_,   BUFFER:=_variant_in_,   LENGTH:=_uint_in_,   PTRCL:=_bool_in_,   DONE=&gt;_bool_out_,   ERROR=&gt;_bool_out_,   STATUS=&gt;_word_out_);</pre>	<p>SEND_PTP déclenche l'émission des données et transfère la mémoire tampon affectée vers l'interface de communication. Le programme de la CPU se poursuit pendant que le CM ou le CB envoie les données à la vitesse de transmission définie. Une seule opération d'émission peut être en attente à un moment donné. Le CM ou le CB renvoie une erreur si une seconde instruction SEND_PTP est exécutée alors qu'il est déjà en train d'envoyer un message.</p>

<sup>1</sup> STEP 7 crée automatiquement le DB lorsque vous insérez l'instruction.

12.3 Communication point à point (PtP)

Tableau 12- 20 Types de données pour les paramètres

Paramètre et type		Type de données	Description
REQ	IN	Bool	Activation de l'émission demandée en cas de front montant de cette entrée de validation d'émission. Cela déclenche le transfert du contenu de la mémoire tampon vers l'interface de communication point à point. (valeur par défaut : False)
PORT	IN	PORT	Une fois que vous avez installé et configuré un appareil de communication CM ou CB, l'identificateur de port apparaît dans la liste déroulante d'aide pour les paramètres disponible au niveau du connecteur PORT de la boîte. La valeur de port CM ou CB affectée est la propriété de configuration d'appareil "identificateur matériel". Le nom symbolique du port est défini dans l'onglet "Constantes système" de la table de variables API. (valeur par défaut : 0)
BUFFER	IN	Variant	Ce paramètre pointe sur l'adresse de début de la mémoire tampon d'émission. (valeur par défaut : 0) <b>Remarque :</b> Les données booléennes et les tableaux booléens ne sont pas acceptés.
LENGTH	IN	UInt	Longueur de trame émise en octets (valeur par défaut : 0) Utilisez toujours une longueur de 0 lorsque vous envoyez une structure complexe.
PTRCL	IN	Bool	Ce paramètre sélectionne la mémoire tampon comme des protocoles point à point normaux ou spécifiques Siemens qui sont mis en oeuvre dans le CM ou le CB associé. (valeur par défaut : False) FALSE = fonctionnement point à point géré par le programme utilisateur. (seule option valable)
DONE	OUT	Bool	TRUE pour un cycle lorsque la dernière demande s'est achevée sans erreur.
ERROR	OUT	Bool	TRUE pour un cycle lorsque la dernière demande s'est achevée avec une erreur.
STATUS	OUT	Word	Code d'erreur d'exécution (valeur par défaut : 0)

Tant qu'une opération d'émission est en cours, les sorties DONE et ERROR sont à FALSE. Lorsqu'une opération d'émission est achevée, soit la sortie DONE soit la sortie ERROR est mise à TRUE pour montrer l'état de l'opération d'émission. Lorsque DONE ou ERROR est TRUE, la sortie STATUS est valide.

L'instruction renvoie un état de 16#7001 si l'interface de communication accepte les données d'émission. Les exécutions suivantes de SEND\_PTP renvoient 16#7002 si le CM ou le CB est toujours occupé à émettre. Une fois l'opération d'émission achevée, le CM ou le CB renvoie l'état 16#0000 pour l'opération d'émission (si aucune erreur ne s'est produite). Les exécutions suivantes de SEND\_PTP avec REQ au niveau bas renvoient l'état 16#7000 (pas occupé).

Les schémas suivants montrent la relation des valeurs de sortie à REQ. On part du principe que l'instruction est appelée périodiquement pour vérifier l'état du processus d'émission. Dans le schéma ci-après, on suppose que l'instruction est appelée à chaque cycle (représenté par les valeurs de STATUS).

REQ							
DONE							
ERROR							
STATUS	7000H	7001H	7002H	7002H	7002H	0000H	7000H

Le schéma suivant montre que les paramètres DONE et STATUS ne sont valables que pour un cycle si la ligne REQ présente une impulsion (pendant un cycle) pour déclencher l'opération d'émission.

REQ								
DONE								
ERROR								
STATUS	7000H	7001H	7002H	7002H	7002H	0000H	7000H	7000H

Le schéma suivant montre la relation des paramètres DONE, ERROR et STATUS en présence d'une erreur.

REQ								
DONE								
ERROR								
STATUS	7000H	7001H	7002H	7002H	7002H	80D1H	7000H	7000H

Les valeurs de DONE, ERROR et STATUS ne sont valables que jusqu'à l'exécution suivante de SENT\_PTP avec le même DB d'instance.

Tableau 12- 21 Codes d'erreur

STATUS (W#16#....)	Description
80D0	Nouvelle demande alors que l'émetteur est actif
80D1	Emission interrompue car pas de CTS pendant le temps d'attente
80D2	Emission interrompue car pas de DSR en provenance de l'équipement DCE
80D3	Emission interrompue en raison d'un débordement de la file d'attente (émission de plus de 1024 octets)
80D5	Signal de polarisation inverse (rupture de fil)
833A	Le DB pour le paramètre BUFFER n'existe pas.

### Interaction entre les paramètres LENGTH et BUFFER pour SEND\_PTP

La taille de données minimale pouvant être envoyée par l'instruction SEND\_PTP est un octet. Le paramètre BUFFER détermine la taille des données à envoyer. Vous ne pouvez pas utiliser le type de données Bool ou des tableaux de Bool pour le paramètre BUFFER.

12.3 Communication point à point (PtP)

Vous pouvez toujours définir le paramètre LENGTH à 0 pour vous assurer que SEND\_PTP envoie la structure de données entière représentée par le paramètre BUFFER. Si vous ne voulez envoyer qu'une partie des données dans le paramètre BUFFER, vous pouvez définir LENGTH comme suit :

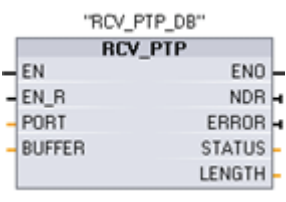
Tableau 12- 22 Paramètres LENGTH et BUFFER

LENGTH	BUFFER	Description
= 0	Non utilisé	L'ensemble des données tel que défini dans le paramètre BUFFER est envoyé. Vous n'avez pas besoin d'indiquer le nombre d'octets émis lorsque LENGTH = 0.
> 0	Type de données simple	La valeur LENGTH doit contenir le nombre d'octets de ce type de données. Pour une valeur Word par exemple, LENGTH doit être égal à deux. Pour une valeur Dword ou Real, LENGTH doit être égal à quatre. Sinon, rien n'est transféré et l'erreur 8088H est renvoyée.
	Structure	La valeur LENGTH peut contenir un nombre d'octets inférieur à la longueur totale en octets de la structure, auquel cas seuls les LENGTH premiers octets de la structure BUFFER sont émis. Mais comme l'organisation interne des octets d'une structure ne peut pas toujours être déterminée, vous risquez d'obtenir des résultats inattendus. Dans ce cas, donnez la valeur 0 au paramètre LENGTH pour envoyer la structure entière.
	Tableau	La valeur LENGTH doit contenir un nombre d'octets qui est inférieur à la longueur totale en octets du tableau et qui doit être un multiple du nombre d'octets de l'élément de données. Par exemple, le paramètre LENGTH doit être un multiple de deux pour un tableau de mots (Word) et un multiple de quatre pour un tableau de réels (Real). Lorsque LENGTH est précisé, le nombre d'éléments de tableau contenus dans LENGTH octets est transféré. Par exemple, si votre paramètre BUFFER contient un tableau de 15 doubles mots (Dword), soit 60 octets au total, et que vous indiquez une longueur LENGTH de 20, les cinq premiers doubles mots (Dword) du tableau seront transférés.  La valeur de LENGTH doit être un multiple du nombre d'octets de l'élément de données. Sinon, STATUS prend la valeur 8088H, ERROR prend la valeur 1 et rien n'est envoyé.
	String	Le paramètre LENGTH contient le nombre de caractères à envoyer. Seuls les caractères de la chaîne (String) sont envoyés. Les octets de longueur maximale et réelle de la chaîne (String) ne sont pas émis.



### 12.3.1.6 Instruction RCV\_PTP

Tableau 12- 23 Instruction RCV\_PTP (Réception point à point)

CONT/LOG	SCL	Description
	<pre>"RCV_PTP_DB" (     EN_R:=_bool_in_,     PORT:=_uint_in_,     BUFFER:=_variant_in_,     NDR=&gt;_bool_out_,     ERROR=&gt;_bool_out_,     STATUS=&gt;_word_out_,     LENGTH=&gt;_uint_out_);</pre>	<p>RCV_PTP vérifie la présence de messages reçus dans le CM ou le CB. Si un message est disponible, il est transféré du CM ou du CB à la CPU. Une valeur STATUS appropriée est renvoyée en cas d'erreur.</p>

<sup>1</sup> STEP 7 crée automatiquement le DB lorsque vous insérez l'instruction.

Tableau 12- 24 Types de données pour les paramètres

Paramètre et type		Type de données	Description
EN_R	IN	Bool	Lorsque cette entrée est à TRUE et qu'un message est disponible, ce dernier est transféré du CM ou du CB dans la mémoire tampon BUFFER. Lorsque EN_R est FALSE, la présence de messages est vérifiée dans le CM ou le CB et les sorties NDR, ERROR et STATUS sont actualisées, mais le message n'est pas transféré dans la mémoire tampon BUFFER. (valeur par défaut : 0)
PORT	IN	PORT	Une fois que vous avez installé et configuré un appareil de communication CM ou CB, l'identificateur de port apparaît dans la liste déroulante d'aide pour les paramètres disponible au niveau du connecteur PORT de la boîte. La valeur de port CM ou CB affectée est la propriété de configuration d'appareil "identificateur matériel". Le nom symbolique du port est défini dans l'onglet "Constantes système" de la table de variables API. (valeur par défaut : 0)
BUFFER	IN	Variant	Ce paramètre pointe sur l'adresse de début de la mémoire tampon de réception. Cette mémoire tampon doit être suffisamment grande pour recevoir la longueur de message maximale. Les données booléennes et les tableaux booléens ne sont pas acceptés. (valeur par défaut : 0)
NDR	OUT	Bool	TRUE pour un cycle lorsque de nouvelles données sont prêtes et que l'opération s'est achevée sans erreur.
ERROR	OUT	Bool	TRUE pour un cycle lorsque l'opération s'est achevée avec une erreur.
STATUS	OUT	Word	Code d'erreur d'exécution (valeur par défaut : 0)
LENGTH	OUT	UInt	Longueur du message renvoyé en octets (valeur par défaut : 0)

La valeur STATUS est valide si soit NDR soit ERROR est TRUE. La valeur STATUS fournit la raison de l'arrêt de l'opération de réception dans le CM ou le CB. Il s'agira typiquement d'une valeur positive indiquant que l'opération de réception a réussi et que le processus de réception s'est terminé normalement. Si la valeur STATUS est négative (le bit de poids fort de la valeur hexadécimale est à 1), cela signifie que l'opération de réception a été interrompue en raison d'une situation d'erreur, telles des erreurs de parité, de trame ou de débordement.

12.3 Communication point à point (PtP)

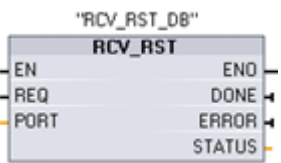
Chaque interface de communication point à point peut mettre jusqu'à 1024 octets en mémoire tampon. Il peut s'agir d'un seul grand message ou de plusieurs petits messages. S'il y a plus d'un message disponible dans le CM ou le CB, l'instruction RCV\_PTP renvoie le message disponible le plus ancien. Une nouvelle exécution de l'instruction RCV\_PTP renverra le message le plus ancien suivant.

Tableau 12- 25 Codes d'erreur

STATUS (W#16#...)	Description
0000	Pas de mémoire tampon
80E0	Message interrompu car la mémoire tampon de réception est pleine
80E1	Message interrompu en raison d'une erreur de parité
80E2	Message interrompu en raison d'une erreur de trame
80E3	Message interrompu en raison d'une erreur de débordement
80E4	Message interrompu car la longueur calculée dépasse la taille de la mémoire tampon
80E5	Signal de polarisation inverse (rupture de fil)
0094	Message interrompu car la longueur de caractères maximum a été reçue
0095	Message interrompu en raison de l'expiration du délai de message
0096	Message interrompu en raison de l'expiration du délai intercaractère
0097	Message interrompu en raison de l'expiration du délai de réponse
0098	Message interrompu car la condition de longueur "N+LEN+M" était satisfaite
0099	Message interrompu car la séquence de fin était satisfaite
833A	Le DB pour le paramètre BUFFER n'existe pas.

12.3.1.7 Instruction RCV\_RST

Tableau 12- 26 Instruction RCV\_RST (Réinitialiser récepteur)

CONT/LOG	SCL	Description
	<pre>"RCV_RST_DB" (     REQ:=_bool_in_,     PORT:=_uint_in_,     DONE=&gt;_bool_out_,     ERROR=&gt;_bool_out_,     STATUS=&gt;_word_out_);</pre>	<p>RCV_RST efface les mémoires tampons de réception dans le CM ou le CB.</p>

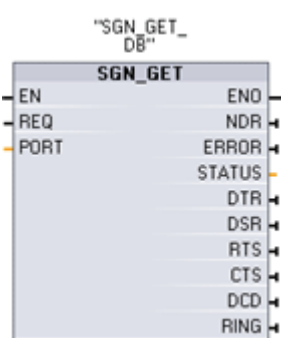
<sup>1</sup> STEP 7 crée automatiquement le DB lorsque vous insérez l'instruction.

Tableau 12- 27 Types de données pour les paramètres

Paramètre et type		Type de données	Description
REQ	IN	Bool	Activation de la réinitialisation du récepteur en cas de front montant de cette entrée de validation (valeur par défaut : False)
PORT	IN	PORT	Une fois que vous avez installé et configuré un appareil de communication CM ou CB, l'identificateur de port apparaît dans la liste déroulante d'aide pour les paramètres disponible au niveau du connecteur PORT de la boîte. La valeur de port CM ou CB affectée est la propriété de configuration d'appareil "identificateur matériel". Le nom symbolique du port est défini dans l'onglet "Constantes système" de la table de variables API. (valeur par défaut : 0)
DONE	OUT	Bool	Cette sortie est TRUE pour un cycle lorsque la dernière requête s'est achevée sans erreur.
ERROR	OUT	Bool	Cette sortie est TRUE lorsque la dernière requête s'est achevée avec des erreurs. Dans ce cas, la sortie STATUS contient en outre les codes d'erreur correspondants.
STATUS	OUT	Word	Code d'erreur (valeur par défaut : 0)

### 12.3.1.8 Instruction SGN\_GET

Tableau 12- 28 Instruction SGN\_GET (Lire signaux RS232)

CONT/LOG	SCL	Description
	<pre>"SGN_GET_DB" (   REQ:=_bool_in_,   PORT:=_uint_in_,   NDR=&gt;_bool_out_,   ERROR=&gt;_bool_out_,   STATUS=&gt;_word_out_,   DTR=&gt;_bool_out_,   DSR=&gt;_bool_out_,   RTS=&gt;_bool_out_,   CTS=&gt;_bool_out_,   DCD=&gt;_bool_out_,   RING=&gt;_bool_out_ );</pre>	<p>SGN_GET lit l'état en cours des signaux de communication RS232.</p> <p>Cette fonction n'est valable que pour le module de communication RS232.</p>

<sup>1</sup> STEP 7 crée automatiquement le DB lorsque vous insérez l'instruction.

12.3 Communication point à point (PtP)

Tableau 12- 29 Types de données pour les paramètres


Paramètre et type		Type de données	Description
REQ	IN	Bool	Lecture des valeurs d'état de signal RS232 en cas de front montant de cette entrée (valeur par défaut : False)
PORT	IN	PORT	Une fois que vous avez installé et configuré un appareil de communication CM ou CB, l'identificateur de port apparaît dans la liste déroulante d'aide pour les paramètres disponible au niveau du connecteur PORT de la boîte. La valeur de port CM ou CB affectée est la propriété de configuration d'appareil "identificateur matériel". Le nom symbolique du port est défini dans l'onglet "Constantes système" de la table de variables API.
NDR	OUT	Bool	TRUE pour un cycle lorsque de nouvelles données sont prêtes et que l'opération s'est achevée sans erreur.
ERROR	OUT	Bool	TRUE pour un cycle lorsque l'opération s'est achevée avec une erreur.
STATUS	OUT	Word	Code d'erreur d'exécution (valeur par défaut : 0)
DTR	OUT	Bool	Terminal de données prêt, module prêt (sortie). Valeur par défaut : False
DSR	OUT	Bool	Modem prêt, partenaire de communication prêt (entrée). Valeur par défaut : False
RTS	OUT	Bool	Demande pour émettre, module prêt à émettre (sortie). Valeur par défaut : False
CTS	OUT	Bool	Prêt à émettre, le partenaire de communication peut recevoir les données (entrée). Valeur par défaut : False
DCD	OUT	Bool	Détection de porteuse, réception du niveau de signal (toujours faux, non pris en charge)
RING	OUT	Bool	Indicateur d'appel, signale un appel entrant (toujours faux, non pris en charge)

Tableau 12- 30 Codes d'erreur

STATUS (W#16#...)	Description
80F0	Le CM ou le CB est un module RS485 et il n'y a pas de signaux disponibles.

### 12.3.1.9 Instruction SGN\_SET

Tableau 12- 31 Instruction SGN\_SET (Activer signaux RS232)

CONT/LOG	SCL	Description
	<pre>"SGN_SET_DB" (   REQ:=_bool_in_,   PORT:=_uint_in_,   SIGNAL:=_byte_in_,   RTS:=_bool_in_,   DTR:=_bool_in_,   DSR:=_bool_in_,   DONE=&gt;_bool_out_,   ERROR=&gt;_bool_out_,   STATUS=&gt; word_out_ );</pre>	<p>SGN_SET active l'état des signaux de communication RS232.</p> <p>Cette fonction n'est valable que pour le module de communication RS232.</p>

<sup>1</sup> STEP 7 crée automatiquement le DB lorsque vous insérez l'instruction.

Tableau 12- 32 Types de données pour les paramètres

Paramètre et type		Type de données	Description
REQ	IN	Bool	Lancement de l'opération d'activation des signaux RS232 en cas de front montant de cette entrée (valeur par défaut : False)
PORT	IN	PORT	Une fois que vous avez installé et configuré un appareil de communication CM ou CB, l'identificateur de port apparaît dans la liste déroulante d'aide pour les paramètres disponible au niveau du connecteur PORT de la boîte. La valeur de port CM ou CB affectée est la propriété de configuration d'appareil "identificateur matériel". Le nom symbolique du port est défini dans l'onglet "Constantes système" de la table de variables API. (valeur par défaut : 0)
SIGNAL	IN	Byte	Sélection du signal à activer : (plusieurs autorisés). Valeur par défaut : 0 <ul style="list-style-type: none"> <li>• 01H = Activer RTS</li> <li>• 02H = Activer DTR</li> <li>• 04H = Activer DSR</li> </ul>
RTS	IN	Bool	Valeur à définir Demande pour émettre, module prêt à émettre (true ou false), valeur par défaut : False
DTR	IN	Bool	Valeur à définir Terminal de données prêt, module prêt (true ou false). Valeur par défaut : False
DSR	IN	Bool	Modem prêt (valable uniquement pour les interfaces de type DCE), non utilisé.
DONE	OUT	Bool	TRUE pour un cycle lorsque la dernière demande s'est achevée sans erreur.
ERROR	OUT	Bool	TRUE pour un cycle lorsque la dernière demande s'est achevée avec une erreur.
STATUS	OUT	Word	Code d'erreur d'exécution (valeur par défaut : 0)

Tableau 12- 33 Codes d'erreur

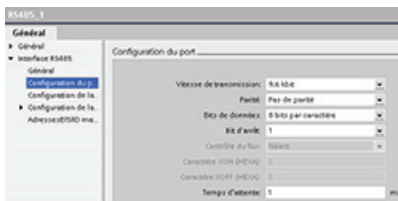
STATUS (W#16#....)	Description
80F0	Le CM ou le CB est un module RS485 et il n'y a pas de signaux pouvant être activés.
80F1	Les signaux ne peuvent être activés en raison d'un contrôle de flux matériel.
80F2	DSR ne peut pas être activé car le module est un équipement terminal de données (DTE).
80F3	DTR ne peut pas être activé car le module est un équipement de transmission de données (DCE).

### 12.3.2 Configuration des ports de communication

Il est possible de configurer les interfaces de communication selon deux méthodes :

- Utilisez la configuration des appareils dans STEP 7 pour configurer les paramètres des ports (débit et parité), les paramètres d'émission et les paramètres de réception. Les paramétrages de la configuration des appareils sont stockés dans la CPU. Ces paramétrages sont appliqués après une mise hors tension puis sous tension et un passage de l'état MARCHE à l'état ARRÊT.
- Utilisez les instructions PORT\_CFG (Page 605) , SEND\_CFG (Page 607) et RCV\_CFG (Page 609) pour définir les paramètres. Les paramétrages des ports définis par les instructions sont valables tant que la CPU reste à l'état MARCHE. Ils reprennent les valeurs de la configuration d'appareil après un passage à l'état ARRÊT ou une mise hors tension et sous tension.

Une fois les unités matérielles configurées (Page 127), vous paramétrez les interfaces de communication en sélectionnant l'un des CM dans le châssis ou le CB le cas échéant.

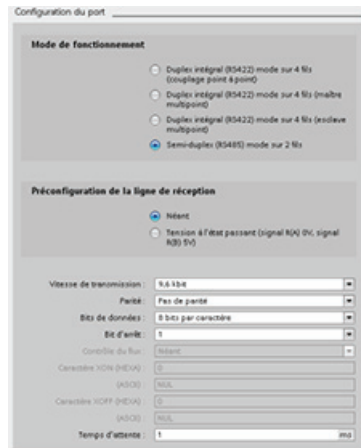


L'onglet "Propriétés" de la fenêtre d'inspection affiche les paramètres du CM ou CB choisi. Sélectionnez "Configuration du port" pour éditer les paramètres suivants :

- Vitesse de transmission
- Parité
- Bits de données par caractère
- Nombre de bits d'arrêt
- Contrôle de flux (RS232 seulement)
- Temps d'attente

Pour les CM 1241 RS232 et RS485 et le CB RS485 (à l'exception du contrôle de flux, pris en charge uniquement par le CM 1241 RS232), les paramètres de configuration de port sont identiques que vous configureriez un module de communication RS232 ou RS485 ou un Communication Board RS485. Les valeurs des paramètres peuvent différer.

Pour le CM 1241 RS422/485, la configuration de port s'effectue différemment comme illustré dans la figure ci-dessous. Le mode 422 du module CM 1241 RS422/485 prend également en charge le contrôle de flux logiciel.



Sélectionnez "Configuration du port" pour éditer les paramètres RS422/485 suivants :

- "Mode de fonctionnement" :
  - Duplex intégral (RS 422) mode sur 4 fils (couplage point à point)
  - Duplex intégral (RS422) mode sur 4 fils (maître multipoint)
  - Duplex intégral (RS422) mode sur 4 fils (esclave multipoint)
  - Semi-duplex (RS485) mode sur 2 fils
- "Préconfiguration de la ligne de réception" :
  - Aucune
  - Tension à l'état passant (signal R(A) 0V, signal R(B) 5V)

Le programme utilisateur STEP 7 peut également configurer le port ou en modifier la configuration à l'aide de l'instruction PORT\_CFG (Page 605).

### Remarque

Les valeurs de paramètres définies par l'instruction PORT\_CFG dans le programme utilisateur ont priorité sur les paramétrages de configuration du port effectués dans la configuration des appareils. Notez que le S7-1200 ne conserve pas les paramètres définis par l'instruction PORT\_CFG en cas de mise hors tension.

Paramètre	Définition
Vitesse de transmission	La valeur par défaut pour la vitesse de transmission est de 9,6 Kbits par seconde. Les choix autorisés sont : 300 bauds, 600 bauds, 1,2 Kbits, 2,4 Kbits, 4,8 Kbits, 9,6 Kbits, 19,2 Kbits, 38,4 Kbits, 57,6 Kbits, 76,8 Kbits et 115,2 Kbits
Parité	La valeur par défaut pour la parité est "Pas de parité". Les choix autorisés sont : Sans parité, parité paire, parité impaire, parité marque (bit de parité toujours à 1) et parité espace (bit de parité toujours à 0)
Bits de données par caractère	Nombre de bits de données dans un caractère. Les choix autorisés sont 7 ou 8.
Nombre de bits d'arrêt	Il peut y avoir un ou deux bits d'arrêt. Le nombre par défaut de bits d'arrêt est 1.
Contrôle du flux	Vous pouvez choisir soit le contrôle de flux matériel soit le contrôle de flux logiciel pour le module de communication RS232 comme décrit au paragraphe "Gestion du contrôle de flux (Page 624)". Si vous sélectionnez le contrôle de flux matériel, vous pouvez choisir si le signal RTS est toujours activé ou si RTS est commuté. Si vous sélectionnez le contrôle de flux logiciel, vous pouvez définir les caractères XON et XOFF.  Les interfaces de communication RS485 n'acceptent pas le contrôle du flux. Le mode 422 du module CM 1241 RS422/485 prend en charge le contrôle de flux logiciel.

Paramètre	Définition
Temps d'attente	Le temps d'attente indique la durée pendant laquelle le CM ou le CB attend de recevoir CTS après avoir affirmé RTS ou de recevoir un caractère XON après avoir reçu un caractère XOFF, selon le type de contrôle de flux. Si le temps d'attente expire avant que l'interface de communication n'ait reçu le CTS ou XON attendu, le CM ou le CB interrompt la transmission et renvoie une erreur au programme utilisateur. Vous indiquez le temps d'attente en millisecondes. La plage va de 0 à 65535 millisecondes.
Mode de fonctionnement	Ce paramètre permet de sélectionner le mode de fonctionnement RS422 ou RS485 et les configurations réseau.
Préconfiguration de la ligne de réception	Ce paramètre permet de sélectionner les options de polarisation. Les valeurs valides sont Aucune, Tension à l'état passant et Polarisation inverse. La polarisation inverse permet la détection de rupture de câble.

### 12.3.2.1 Gestion du contrôle de flux

"Contrôle de flux" fait référence à un mécanisme pour équilibrer l'envoi et la réception de transmissions de données de manière à éviter la perte de données. Le contrôle de flux garantit qu'un émetteur n'envoie pas plus d'informations que le récepteur ne peut en gérer. Le contrôle de flux peut être matériel ou logiciel. Le CM RS232 prend en charge le contrôle de flux matériel et le contrôle de flux logiciel. Le CM et le CB RS485 n'acceptent pas le contrôle du flux. Le mode 422 du module CM 1241 RS422/485 prend en charge le contrôle de flux logiciel. Vous indiquez le type de contrôle de flux lorsque vous configurez le port (Page 622) ou avec l'instruction PORT\_CFG (Page 605).

Le contrôle de flux matériel opère via les signaux de communication Demande pour émettre (RTS) et Prêt à émettre (CTS). Sur le CM RS232, le signal RTS est envoyé à partir de la broche 7 et le signal CTS est reçu par le biais de la broche 8. Le CM RS232 est un équipement terminal de données (DTE) qui affirme RTS en sortie et surveille CTS en entrée.

#### Contrôle de flux matériel : commutation RTS

Si vous activez le contrôle de flux matériel à commutation RTS pour un CM RS232, le module active le signal RTS pour envoyer des données. Il écoute le signal CTS pour déterminer si l'appareil récepteur peut accepter des données. Lorsque le signal CTS est actif, le module peut envoyer des données tant que le signal CTS reste actif. L'émission doit être interrompue si le signal CTS devient inactif.

L'émission reprend lorsque le signal CTS redevient actif. Si le signal CTS n'est pas activé pendant le temps d'attente configuré, le module interrompt la transmission et renvoie une erreur au programme utilisateur. Vous indiquez le temps d'attente dans la configuration du port (Page 622).

Le contrôle de flux à commutation RTS est utile pour les appareils qui ont besoin d'un signal indiquant que l'émission est active, par exemple pour un modem radio qui utilise RTS en tant que signal "clé" pour exciter l'émetteur radio. Le contrôle de flux à commutation RTS ne fonctionne pas avec les modems téléphoniques standard. Utilisez l'option RTS toujours activé pour les modems téléphoniques.



### Contrôle de flux matériel : RTS toujours activé

En mode RTS toujours activé, le CM 1241 active RTS par défaut. Un appareil tel qu'un modem téléphonique écoute le signal RTS provenant du CM et utilise ce signal en tant que signal "prêt à émettre". Le modem n'émet vers le CM que lorsque RTS est actif, c'est-à-dire lorsque le modem téléphonique voit un CTS actif. Si RTS est inactif, le modem téléphonique n'émet pas vers le CM.

Pour permettre au modem d'envoyer des données au CM à tout moment, configurez le contrôle de flux matériel "RTS toujours activé". Le CM active ainsi le signal RTS tout le temps. Le CM n'inactive pas RTS même si le module ne peut pas accepter de caractères. L'appareil émetteur doit s'assurer qu'il ne fait pas déborder le tampon de réception du CM.

### Utilisation des signaux Terminal de données prêt (DTR) et Modem prêt (DSR)

Le CM active DTR pour les deux types de contrôle de flux matériel. Le module n'émet que lorsque le signal DSR devient actif. L'état de DSR n'est évalué qu'au début de l'opération d'émission. Si DSR devient inactif une fois l'émission commencée, cette dernière n'est pas suspendue.

### Contrôle de flux logiciel

Le contrôle de flux logiciel utilise des caractères spéciaux dans les messages pour fournir le contrôle de flux. Vous configurez les caractères hexadécimaux qui représentent XON et XOFF.

XOFF indique qu'une émission doit s'arrêter. XON indique qu'une émission peut reprendre. XOFF et XON doivent être des caractères différents.

Lorsque l'émetteur reçoit un caractère XOFF du récepteur, il arrête d'émettre. L'émission reprend lorsque l'émetteur reçoit un caractère XON. S'il ne reçoit pas de caractère XON pendant le temps d'attente indiqué dans la configuration du port (Page 622), le CM arrête l'émission et renvoie une erreur au programme utilisateur.

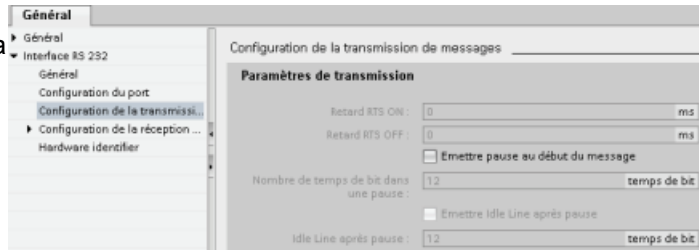
Le contrôle de flux logiciel requiert une communication duplex intégral puisque le récepteur doit être capable d'envoyer XOFF à l'émetteur alors qu'une transmission est en cours. Le contrôle de flux logiciel n'est possible qu'avec les messages qui contiennent uniquement des caractères ASCII. Les protocoles binaires ne peuvent pas utiliser le contrôle de flux logiciel.

## 12.3.3 Configuration des paramètres d'émission et de réception

Avant que la CPU ne puisse entamer une communication point à point, vous devez configurer les paramètres pour l'émission et la réception de messages. Ces paramètres conditionnent la façon dont la communication fonctionne lorsque des messages sont envoyés à un appareil cible ou reçu d'un appareil cible.

### 12.3.3.1 Configuration des paramètres d'émission

Dans la configuration des appareils, vous configurez la manière dont une interface de communication envoie des données en paramétrant la propriété "Configuration de la transmission de messages" pour l'interface sélectionnée.



Vous pouvez également configurer ou modifier dynamiquement les paramètres d'émission de messages à partir du programme utilisateur à l'aide de l'instruction SEND\_CFG (Page 607).

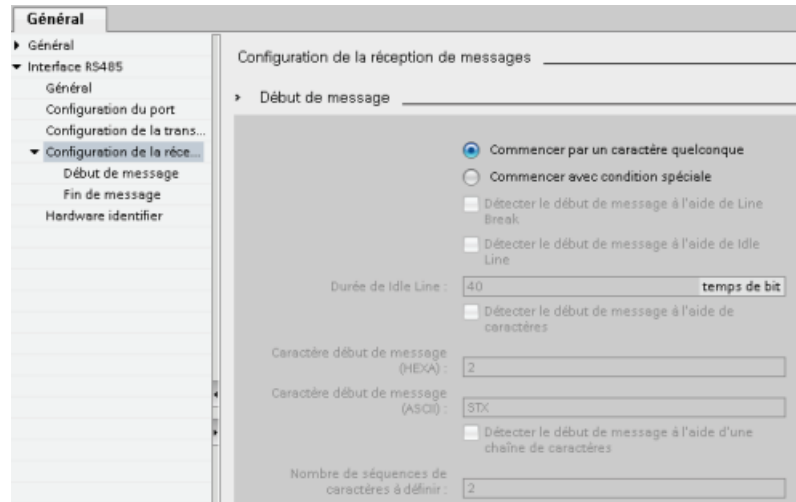
#### Remarque

Les valeurs de paramètres définies par l'instruction SEND\_CFG dans le programme utilisateur ont priorité sur les paramétrages de configuration du port. Notez que la CPU ne conserve pas les paramètres définis par l'instruction SEND\_CFG en cas de mise hors tension.

Paramètre	Définition
Retard RTS ON	Indique le temps à attendre après l'activation de RTS avant de déclencher l'émission. La plage va de 0 à 65535 ms, 0 étant la valeur par défaut. Ce paramètre n'est valable que lorsque le contrôle de flux matériel est activé dans la configuration du port (Page 622). CTS est évalué après expiration du temps de retard RTS activé. Ce paramètre ne s'applique qu'aux modules RS232.
Retard RTS OFF	Indique le temps à attendre avant de désactiver RTS après l'achèvement de l'émission. La plage va de 0 à 65535 ms, 0 étant la valeur par défaut. Ce paramètre n'est valable que lorsque le contrôle de flux matériel est activé dans la configuration du port (Page 622). Ce paramètre ne s'applique qu'aux modules RS232.
Emettre Pause au début des messages Nombre de Bit Times dans une pause	Indique qu'une pause sera envoyée au début de chaque message après expiration du retard RTS ON (si configuré) et activation de CTS. Vous indiquez le nombre de temps bit qui constituent une pause où la ligne est maintenue en état de repos. La valeur par défaut est 12 et la valeur maximale est 65535, avec une limite de huit secondes.
Emettre Idle Line après Pause Idle Line après Pause	Indique qu'une ligne inactive sera envoyée avant le début du message. Elle sera envoyée après la pause si une pause est configurée. Le paramètre "Idle Line après Pause" indique le nombre de temps bit qui constituent une ligne inactive où la ligne est maintenue en état de travail. La valeur par défaut est 12 et la valeur maximale est 65535, avec une limite de huit secondes.

### 12.3.3.2 Configuration des paramètres de réception

Dans la configuration d'appareil, vous définissez comment l'interface de communication reçoit des données et comment elle reconnaît le début et la fin d'un message. Indiquez ces paramètres dans la Configuration de la réception de messages pour l'interface sélectionnée.



Vous pouvez également configurer ou modifier dynamiquement les paramètres de réception de messages à partir du programme utilisateur à l'aide de l'instruction RCV\_CFG (Page 609).

#### Remarque

Les valeurs de paramètres définies par l'instruction RCV\_CFG dans le programme utilisateur ont priorité sur les paramétrages de configuration du port. Notez que la CPU ne conserve pas les paramètres définis par l'instruction RCV\_CFG en cas de mise hors tension.

### Conditions de début de message

Vous pouvez déterminer comment l'interface de communication reconnaît le début d'un message. Les caractères de début et les caractères contenant le message sont inscrits dans la mémoire tampon de réception jusqu'à apparition d'une condition de fin configurée.

Vous pouvez indiquer plusieurs conditions de début. Si vous définissez plusieurs conditions de début, toutes les conditions de début doivent être satisfaites pour que le message soit considéré comme commencé. Par exemple, si vous configurez un temps de ligne inactive et un caractère de début spécifique, le CM ou le CB vérifie d'abord que la condition concernant le temps de ligne inactive est satisfaite, puis il recherche le caractère de début indiqué. S'il reçoit un caractère autre que le caractère de début indiqué, le CM ou le CB relance la recherche de début de message en surveillant à nouveau le temps de ligne inactive.

Paramètre	Définition
Commencer par un caractère quelconque	La condition Caractère quelconque indique que tout caractère reçu avec succès signale le début d'un message. Ce caractère est le premier caractère dans le message.
Pause	La condition Pause indique qu'une opération de réception de message commence après réception d'un caractère de pause.

Paramètre	Définition
Ligne inactive	<p>La condition Ligne inactive indique qu'une réception de message commence lorsque la ligne de réception a été inactive ou au repos pendant le nombre de temps bit indiqué. Lorsque cette condition apparaît, le message commence.</p> <p>① Caractères                  ② Redémarre la temporisation de ligne inactive                  ③ Une ligne inactive est détectée et la réception de message commence</p>
Condition spéciale : Détecter le début de message à l'aide d'un caractère unique	<p>Indique qu'un caractère particulier signale le début d'un message. Ce caractère est alors le premier caractère dans le message. Tout caractère reçu avant ce caractère spécifique est rejeté. Le caractère STX est pris par défaut.</p>
Condition spéciale : Détecter le début de message à l'aide d'une chaîne de caractères	<p>Indique qu'une séquence de caractères particulière parmi quatre séquences configurées au plus signale le début d'un message. Vous pouvez indiquer jusqu'à cinq caractères pour chaque séquence. Pour chaque position de caractère, vous indiquez soit un caractère hexadécimal spécifique soit que le caractère n'est pas pris en compte dans la comparaison de séquence (caractère joker). Le dernier caractère spécifique d'une séquence met fin à cette séquence de début.</p> <p>Les séquences entrantes sont comparées aux conditions de début configurées jusqu'à ce qu'une condition de début soit satisfaite. Une fois la séquence de début satisfaite, l'évaluation des conditions de fin commence.</p> <p>Vous pouvez configurer jusqu'à quatre séquences de caractères spécifiques. Vous utilisez une condition de début multi-séquence quand différentes séquences de caractères peuvent indiquer le début d'un message. Si l'une quelconque des séquences de caractères est détectée, le message commence.</p>

L'ordre de vérification des conditions de début est défini comme suit :

- Ligne inactive
- Pause
- Caractères ou séquences de caractères

Si plusieurs conditions de début ont été définies et que l'une de ces conditions n'est pas satisfaite, le CM ou le CB relance la vérification en commençant par la première condition requise. Une fois que le CM ou le CB a établi que les conditions de début ont été satisfaites, il commence l'évaluation des conditions de fin.

## Exemple de configuration avec début de message à l'apparition d'une séquence de caractères parmi deux

Considérons la configuration de condition de début de message suivante :

Détecter le début de message à l'aide d'une chaîne de caractères

Nombre de séquences de caractères à définir :

**Séquence de 5 caractères pour début de message**

**Début de message séquence 1**

Contrôler ce caractère : 1

Caractère (HEXA) :

Caractère (ASCII) :

Contrôler ce caractère : 2

Caractère (HEXA) :

Caractère (ASCII) :

Contrôler ce caractère : 3

Caractère (HEXA) :

Caractère (ASCII) :

Contrôler ce caractère : 4

Caractère (HEXA) :

Caractère (ASCII) :

Contrôler ce caractère : 5

Caractère (HEXA) :

Caractère (ASCII) :

**Début de message séquence 2**

Contrôler ce caractère : 1

Caractère (HEXA) :

Caractère (ASCII) :

Contrôler ce caractère : 2

Caractère (HEXA) :

Caractère (ASCII) :

Contrôler ce caractère : 3

Caractère (HEXA) :

Caractère (ASCII) :

Contrôler ce caractère : 4

Caractère (HEXA) :

Caractère (ASCII) :

Contrôler ce caractère : 5

Caractère (HEXA) :

Caractère (ASCII) :

### 12.3 Communication point à point (PtP)

Dans cette configuration, la condition de début est satisfaite lorsque l'un ou l'autre des motifs suivants est détecté :

- Une séquence de cinq caractères est reçue : le premier caractère est 0x6A, le cinquième caractère est 0x1C et les positions 2, 3 et 4 peuvent contenir n'importe quel caractère. L'évaluation des conditions de fin commence une fois le cinquième caractère reçu.
- Deux caractères 0x6A consécutifs, précédés de n'importe quel caractère, sont reçus. Dans ce cas, l'évaluation des conditions de fin commence après la réception du deuxième 0x6A (3 caractères). Le caractère précédant le premier 0x6A est inclus dans la condition de début.

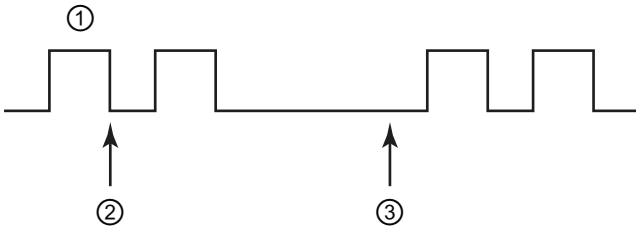
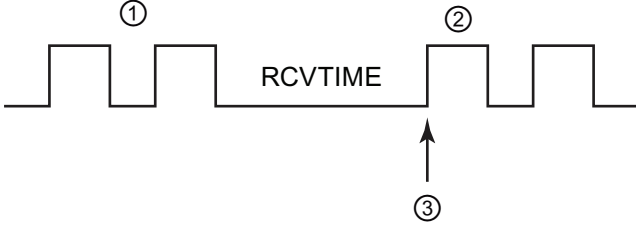
Exemples de séquences qui satisfont à cette condition de début :

- <tout caractère> 6A 6A
- 6A 12 14 18 1C
- 6A 44 A5 D2 1C

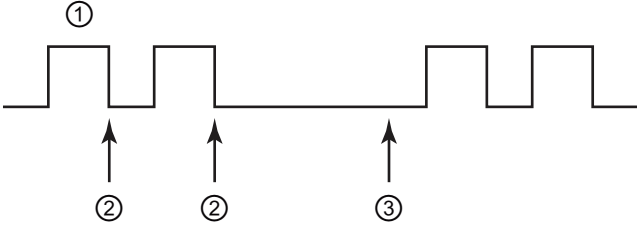
### Conditions de fin de message

Vous déterminez également comment l'interface de communication reconnaît la fin d'un message. Vous pouvez configurer plusieurs conditions de fin de message. Si l'une quelconque des conditions configurées apparaît, le message s'arrête.

Par exemple, vous pouvez indiquer une condition de fin avec un dépassement du délai d'attente de message de 300 millisecondes, un dépassement du délai inter-caractères de 40 temps bit et une longueur maximale de 50 octets. La fin de message se produira si la réception du message prend plus de 300 millisecondes ou si le délai entre deux caractères quelconques dépasse 40 temps bit ou si 50 octets sont reçus.

Paramètre	Définition
<p>Détecter la fin de message à l'aide du dépassement de temps message</p>	<p>La fin de message se produit lorsque la durée configurée d'attente de la fin du message expire. Le délai d'attente de message commence à s'écouler lorsqu'une condition de début a été satisfaite. La valeur par défaut est de 200 ms, la plage étant comprise entre 0 et 65535 ms.</p>  <p>① Caractères reçus                  ② Condition de début de message satisfaite : la temporisation de message commence                  ③ La temporisation de message expire et termine le message</p>
<p>Détecter la fin de message à l'aide du dépassement de temps de réponse</p>	<p>La fin de message se produit lorsque la durée configurée d'attente d'une réponse expire avant qu'une séquence de début valide ne soit reçue. Le délai d'attente de réponse commence à s'écouler lorsqu'une émission s'achève et que le CM ou le CB commence l'opération de réception. Le délai d'attente de réponse par défaut est de 200 ms, la plage étant comprise entre 0 et 65535 ms. Si aucun caractère n'est reçu pendant le délai d'attente de réponse RCVTIME, une erreur est renvoyée à l'instruction RCV_PTP correspondante. Le dépassement du délai d'attente de la réponse ne définit pas une condition de fin spécifique. Il indique seulement qu'un caractère doit être reçu avec succès pendant le temps mentionné. Vous devez configurer une autre condition de fin pour signaler la fin réelle d'un message.</p>  <p>① Caractères émis                  ② Caractères reçus                  ③ Le premier caractère devrait être reçu maintenant.</p>

12.3 Communication point à point (PtP)

Paramètre	Définition
<p>Détecter la fin de message à l'aide d'un dépassement du délai inter-caractères</p>	<p>La fin de message se produit lorsque la durée maximale configurée entre deux caractères consécutifs quelconques d'un message expire. La valeur par défaut du délai inter-caractères est 12 temps bit et la valeur maximale est 65535 temps bit, avec un maximum de huit secondes.</p>  <p>The diagram shows a digital signal representing character reception. It consists of several pulses of equal width and height, separated by gaps. The first pulse is labeled with a circled '1'. The gaps between the first and second pulses, and between the second and third pulses, are each labeled with a circled '2' and an upward-pointing arrow. The gap between the third and fourth pulses is significantly longer than the previous ones and is labeled with a circled '3' and an upward-pointing arrow. Below the diagram, a legend explains the labels: ① Caractères reçus, ② Redémarre la temporisation inter-caractères, and ③ La temporisation inter-caractères expire et met fin au message.</p> <p>① Caractères reçus                  ② Redémarre la temporisation inter-caractères                  ③ La temporisation inter-caractères expire et met fin au message.</p>
<p>Détecter la fin de message à l'aide de la longueur maximale</p>	<p>La fin de message se produit lorsque le nombre configuré maximum de caractères a été reçu. La plage autorisée pour la longueur maximale va de 1 à 1023.</p> <p>Cette condition peut être utilisée pour empêcher le débordement de la mémoire tampon de messages. Lorsque cette condition de fin est combinée à des conditions de fin par dépassement de délai et qu'il y a dépassement de délai, tous les caractères reçus valides sont fournis même si la longueur maximale n'a pas été atteinte. Cela permet la prise en charge de protocoles de longueur variable lorsque seule la longueur maximale est connue.</p>
<p>Lire la longueur dans le message</p>	<p>Le message lui-même indique sa longueur. La fin de message se produit lorsqu'un message de la longueur indiquée a été reçu. La méthode pour indiquer et interpréter la longueur du message est décrite ci-après.</p>
<p>Détecter la fin de message à l'aide d'un caractère</p>	<p>La fin de message se produit lorsque le caractère indiqué est reçu.</p>
<p>Détecter la fin de message à l'aide d'une chaîne de caractères</p>	<p>La fin de message se produit lorsque la séquence de caractères indiquée est reçue. Vous pouvez indiquer une séquence de cinq caractères au maximum. Pour chaque position de caractère, vous indiquez soit un caractère hexadécimal spécifique soit que le caractère n'est pas pris en compte dans la comparaison de séquence.</p> <p>Les caractères en tête qui sont des caractères ignorés ne font pas partie de la condition de fin. Les caractères à droite qui sont des caractères ignorés font partie de la condition de fin.</p>



## Exemple de configuration avec fin de message à l'apparition d'une séquence de caractères

Considérons la configuration de condition de fin de message suivante :

Détection de la fin de message à l'aide d'une chaîne de caractères

**Séquence de 5 caractères pour fin de message**

Contrôler ce caractère : 1

Caractère (HEXA) : 0

Caractère (ASCII) : QUELCONQUE

Contrôler ce caractère : 2

Caractère (HEXA) : 6A

Caractère (ASCII) : |

Contrôler ce caractère : 3

Caractère (HEXA) : 6A

Caractère (ASCII) : |

Contrôler ce caractère : 4

Caractère (HEXA) : 0

Caractère (ASCII) : QUELCONQUE

Contrôler ce caractère : 5

Caractère (HEXA) : 0

Caractère (ASCII) : QUELCONQUE

Dans ce cas, la condition de fin est satisfaite lorsque deux caractères 0x7A consécutifs sont reçus et suivis de deux caractères quelconques. Le caractère précédant le motif 0x7A 0x7A ne fait pas partie de la séquence de caractères de fin. Les deux caractères suivant le motif 0x7A 0x7A sont nécessaires pour achever la séquence de caractères de fin. Les valeurs reçues aux positions 4 et 5 sont indifférentes mais doivent être reçues pour satisfaire la condition de fin.

## Indication de la longueur de message à l'intérieur du message

Lorsque vous sélectionnez la condition spéciale pour laquelle la longueur du message est incluse dans le message, vous devez fournir trois paramètres qui renseignent sur la longueur du message.

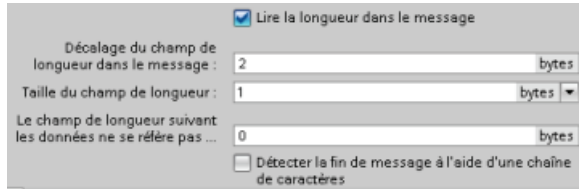
La structure réelle du message varie selon le protocole utilisé. Les trois paramètres sont les suivants :

- n : position de caractère (base 1) dans le message qui commence l'indication de longueur
- Taille du champ de longueur : nombre d'octets (un, deux ou quatre) de l'indication de longueur
- Longueur m : nombre de caractères suivant l'indication de longueur qui ne sont pas inclus dans le décompte de longueur

Les caractères de fin n'ont pas besoin d'être contigus. La valeur de "Longueur m" peut servir à indiquer la longueur d'un champ de total de contrôle dont la taille n'est pas incluse dans le champ de longueur.

12.3 Communication point à point (PtP)

Ces champs apparaissent dans la Configuration de la réception de messages des propriétés d'appareil :



**Exemple 1 :** Considérons un message structuré selon le protocole suivant :

STX	Lg (n)	Caractères 3 à 14 comptés par la longueur											
		ADR	PKE		INDEX		PWD		STW		HSW		BCC
1	2	3	4	5	6	7	8	9	10	11	12	13	14
STX	0x0C	xx	xxxx		xxxx		xxxx		xxxx		xxxx		xx

Configurez les paramètres de longueur pour la réception de ce message comme suit :

- n = 2 (l'indication de longueur du message commence au niveau de l'octet 2)
- Taille du champ de longueur = 1 (la longueur du message est définie dans un octet)
- Longueur m = 0 (il n'y a pas d'autres caractères après l'indication de longueur qui ne sont pas inclus dans le décompte de longueur ; douze caractères suivent l'indication de longueur)

Dans cet exemple, les caractères 3 à 14 inclus sont les caractères comptés par Lg (n).

**Exemple 2 :** Considérons un autre message structuré selon le protocole suivant :

SD1	Lg (n)	Lg (n)	SD2	Caractères 5 à 10 comptés par la longueur						FCS	ED
				DA	SA	FA	Unité de données=3 octets				
1	2	3	4	5	6	7	8	9	10	11	12
xx	0x06	0x06	xx	xx	xx	xx	xx	xx	xx	xx	xx

Configurez les paramètres de longueur pour la réception de ce message comme suit :

- n = 3 (l'indication de longueur du message commence au niveau de l'octet 3)
- Taille du champ de longueur = 1 (la longueur du message est définie dans un octet)
- Longueur m = 3 (il y a trois caractères après l'indication de longueur qui ne sont pas comptés dans la longueur. Dans le protocole de cet exemple, les caractères SD2, FCS et ED ne sont pas comptés dans le décompte de longueur. Les six autres caractères sont comptés dans le décompte de longueur ; ainsi le nombre total de caractères suivant l'indication de longueur est neuf.)

Dans cet exemple, les caractères 5 à 10 inclus sont les caractères comptés par Lg (n).

### 12.3.4 Programmation de la communication point à point

STEP 7 fournit des instructions avancées qui permettent au programme utilisateur d'exécuter la communication point à point avec un protocole conçu et indiqué dans le programme utilisateur. Ces instructions peuvent être classées dans deux catégories :

- Instructions de configuration
- Instructions de communication

#### Instructions de configuration

Avant que votre programme utilisateur ne puisse exécuter une communication point à point, vous devez configurer le port de l'interface de communication ainsi que les paramètres pour l'émission et la réception de données.

Vous pouvez effectuer la configuration du port et des messages pour chaque CM ou CB via la configuration d'appareil ou par le biais de ces instructions dans votre programme utilisateur :

- PORT\_CFG (Page 605)
- SEND\_CFG (Page 607)
- RCV\_CFG (Page 609)

#### Instructions de communication

Les instructions de communication point à point permettent au programme utilisateur d'envoyer des messages aux interfaces de communication et d'en recevoir. Pour plus d'informations sur le transfert de données à l'aide de ces instructions, reportez-vous au paragraphe sur la cohérence des données (Page 163).

Toutes les fonctions point à point opèrent de manière asynchrone. Le programme utilisateur peut utiliser une architecture d'interrogation pour déterminer l'état des émissions et des réceptions. SEND\_PTP et RCV\_PTP peuvent s'exécuter simultanément. Les modules de communication et le Communication Board mettent les messages d'émission et de réception en mémoire tampon selon les besoins, jusqu'à une taille de mémoire tampon maximale de 1024 octets.

Les CM et le CB envoient des messages aux appareils point à point réels et en reçoivent. Le protocole du message est dans une mémoire tampon qui est reçue d'un port de communication spécifique ou lui est envoyée. La mémoire tampon et le port sont des paramètres des instructions d'émission et de réception :

- SEND\_PTP (Page 613)
- RCV\_PTP (Page 617)

D'autres instructions permettent de réinitialiser la mémoire tampon de réception ainsi que de lire et d'activer des signaux RS232 spécifiques :

- RCV\_RST (Page 618)
- SGN\_GET (Page 619)
- SGN\_SET (Page 621)

### 12.3.4.1 Architecture d'interrogation

Les instructions point à point du S7-1200 doivent être appelées cycliquement/périodiquement pour vérifier si des messages ont été reçus. L'interrogation de l'émission indiquera au programme utilisateur lorsque l'émission s'est achevée.

#### Architecture d'interrogation : maître

La séquence typique pour un maître se présente comme suit :

1. Une instruction SEND\_PTP déclenche une émission vers le CM ou le CB.
2. L'instruction SENT\_PTP est exécutée lors des cycles suivants pour interroger l'état d'achèvement de l'émission.
3. Lorsque l'instruction SEND\_PTP indique que l'émission est achevée, le code utilisateur peut préparer la réception de la réponse.
4. L'instruction RCV\_PTP est exécutée de manière répétée pour vérifier s'il y a une réponse. Lorsque le CM ou le CB a collecté un message de réponse, l'instruction RCV\_PTP copie la réponse dans la CPU et signale que de nouvelles données ont été reçues.
5. Le programme utilisateur peut traiter la réponse.
6. Retournez à l'étape 1 et répétez le cycle.

#### Architecture d'interrogation : esclave

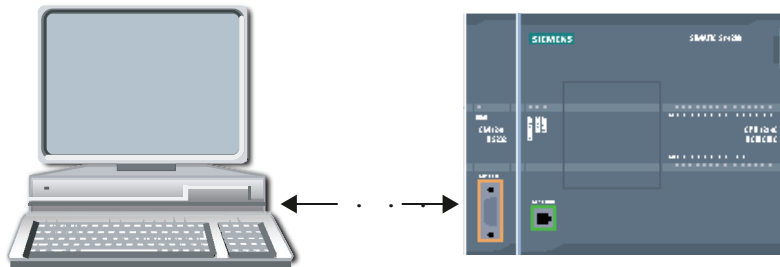
La séquence typique pour un esclave se présente comme suit :

1. Le programme utilisateur exécute l'instruction RCV\_PTP à chaque cycle.
2. Lorsque le CM ou le CB a reçu une demande, l'instruction RCV\_PTP signale que de nouvelles données sont prêtes et la demande est copiée dans la CPU.
3. Le programme utilisateur honore la demande et génère une réponse.
4. Utilisez une instruction SEND\_PTP pour renvoyer la réponse au maître.
5. Exécutez SEND\_PTP de manière répétée pour être sûr que l'émission a lieu.
6. Retournez à l'étape 1 et répétez le cycle.

L'esclave doit appeler RCV\_PTP suffisamment souvent pour recevoir une émission du maître avant que le délai d'attente de réponse n'expire chez le maître. A cet effet, le programme utilisateur peut appeler RCV\_PTP à partir d'un OB d'alarme cyclique dont la période est suffisante pour recevoir une émission du maître avant que le délai d'attente n'expire. Si vous définissez la période de l'OB de manière à permettre deux exécutions pendant le délai d'attente du maître, le programme utilisateur peut recevoir les émissions sans en manquer aucune.

### 12.3.5 Exemple : Communication point à point

Dans cet exemple, une CPU S7-1200 communique avec un PC muni d'un émulateur de terminal par le biais d'un module CM 1241 RS232. La configuration point à point et le programme STEP 7 dans cet exemple illustrent comment la CPU peut recevoir un message du PC et renvoyer le message en écho au PC.



Vous devez connecter l'interface de communication du module CM 1241 RS232 à l'interface RS232 du PC qui est normalement COM1. Comme ces deux ports sont des équipements terminaux de données (DTE), vous devez permuter les broches d'émission et de réception (2 et 3) lorsque vous connectez les ports, ce que vous faites à l'aide de l'une des deux méthodes suivantes :

- Utilisez un adaptateur de modem nul pour permuter les broches 2 et 3 avec un câble RS232 standard.
- Utilisez un câble de modem nul dont les broches 2 et 3 sont déjà permutées. Vous pouvez habituellement identifier un câble de modem nul par ses deux extrémités à connecteur D 9 broches femelle.

### 12.3.5.1 Configuration du module de communication

Vous pouvez configurer le CM 1241 dans la configuration des appareils de STEP 7 ou par le biais d'instructions dans le programme utilisateur. Dans cet exemple, nous utilisons la configuration des appareils.

- Configuration du port : Cliquez sur le port de communication du module CM dans la configuration des appareils et configurez le port comme illustré :

#### Remarque

Les paramètres pour "Mode de fonctionnement" et "Préconfiguration de la ligne de réception" valent uniquement pour le module CM 1241 (RS422/RS485). Ces paramètres de configuration de port n'existent pas pour les autres modules CM 1241. Reportez-vous à Configuration de RS422 et RS485 (Page 641)

- Configuration de l'émission de messages : Acceptez les valeurs par défaut pour la configuration de l'émission de messages. Aucune pause ne doit être envoyée en début de message.

- Configuration du début de la réception de message : Configurez le CM 1241 pour qu'il commence la réception de message lorsque la ligne de communication a été inactive pendant au moins 50 temps bit (environ 5 millisecondes à 9600 bauds =  $50 * 1/9600$ ) :

Début de message

Commencer par un caractère quelconque  
 Commencer avec condition spéciale  
 Détection le début de message à l'aide de Line Break  
 Détection le début de message à l'aide de Idle Line  
 Durée de Idle Line : 50 temps de bit  
 Détection le début de message à l'aide de caractères  
 Caractère début de message (HEXA) : 2  
 Caractère début de message (ASCII) : STX  
 Détection le début de message à l'aide d'une chaîne de caractères  
 Nombre de séquences de caractères à définir : 1

- Configuration de la fin de la réception de message : Configurez le CM 1241 pour qu'il mette fin à un message lorsqu'il a reçu 100 octets au maximum ou un caractère de changement de ligne (10 décimal ou A hexadécimal). La séquence de fin autorise jusqu'à cinq caractères de fin à la suite. Le cinquième caractère de la séquence est le caractère de changement de ligne. Les quatre caractères précédents de la séquence sont des caractères indifférents. Le CM 1241 n'évalue pas les caractères indifférents mais recherche un caractère de changement de ligne précédé de zéro ou de plusieurs caractères indifférents pour indiquer la fin de message.

Fin de message

Définir les conditions de fin de message

Détection la fin de message à l'aide du dépassement de temps message  
 Dépassement de temps message : 200 ms  
 Détection la fin de message à l'aide du dépassement de temps de réponse  
 Dépassement du temps de réponse : 200 ms  
 Détection la fin de message à l'aide d'un dépassement du délai inter-caractères  
 Dépassement du délai inter-caractères : 12 temps de bit  
 Détection la fin de message à l'aide de la longueur maximale  
 Longueur maximale du message : 100 bytes  
 Lire la longueur dans le message  
 Décalage du champ de longueur dans le message : 1 bytes  
 Taille du champ de longueur : 1 bytes  
 Le champ de longueur suivant les données ne se réfère pas ... : 0 bytes  
 Détection la fin de message à l'aide d'une chaîne de caractères

12.3 Communication point à point (PtP)

**Séquence de 5 caractères pour fin de message**

<input type="checkbox"/> Contrôler ce caractère : 1	
Caractère (HEXA) :	0
Caractère (ASCII) :	QUELCONQUE
<input type="checkbox"/> Contrôler ce caractère : 2	
Caractère (HEXA) :	0
Caractère (ASCII) :	QUELCONQUE
<input type="checkbox"/> Contrôler ce caractère : 3	
Caractère (HEXA) :	0
Caractère (ASCII) :	QUELCONQUE
<input type="checkbox"/> Contrôler ce caractère : 4	
Caractère (HEXA) :	0
Caractère (ASCII) :	QUELCONQUE
<input checked="" type="checkbox"/> Contrôler ce caractère : 5	
Caractère (HEXA) :	A
Caractère (ASCII) :	LF



### 12.3.5.2 Configuration de RS422 et RS485

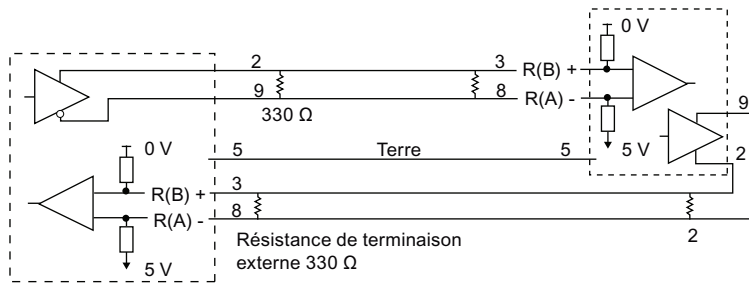
#### Configuration de RS422

Pour le mode RS422, il existe trois modes de fonctionnement selon la configuration de votre réseau. Sélectionnez l'un de ces modes de fonctionnement en fonction des appareils dans votre réseau. Les différents choix pour Préconfiguration de la ligne de réception font référence aux cas détaillés plus loin.

- Duplex intégral (RS422) mode sur 4 fils (couplage point à point) : Sélectionnez cette option lorsqu'il y a deux appareils sur votre réseau. Pour Préconfiguration de la ligne de réception :
  - Sélectionnez Aucune lorsque vous fournissez la polarisation et la terminaison (cas 3).
  - Sélectionnez Tension à l'état passant pour utiliser la polarisation et la terminaison internes (cas 2).
  - Sélectionnez Polarisation inverse pour utiliser la polarisation et la terminaison internes et activer la détection de rupture de câble pour les deux appareils (cas 1).
- Duplex intégral (RS422) mode sur 4 fils (maître multipoint) : Sélectionnez cette option pour le maître lorsque vous avez un réseau avec un maître et plusieurs esclaves. Pour Préconfiguration de la ligne de réception :
  - Sélectionnez Aucune lorsque vous fournissez la polarisation et la terminaison (cas 3).
  - Sélectionnez Tension à l'état passant pour utiliser la polarisation et la terminaison internes (cas 2).
  - La détection de rupture de câble n'est pas possible dans ce mode.
- Duplex intégral (RS422) mode sur 4 fils (esclave multipoint) : Sélectionnez cette option pour tous les esclaves lorsque vous avez un réseau avec un maître et plusieurs esclaves. Pour Préconfiguration de la ligne de réception :
  - Sélectionnez Aucune lorsque vous fournissez la polarisation et la terminaison (cas 3).
  - Sélectionnez Tension à l'état passant pour utiliser la polarisation et la terminaison internes (cas 2).
  - Sélectionnez Polarisation inverse pour utiliser la polarisation et la terminaison internes et activer la détection de rupture de câble pour les esclaves (cas 1).

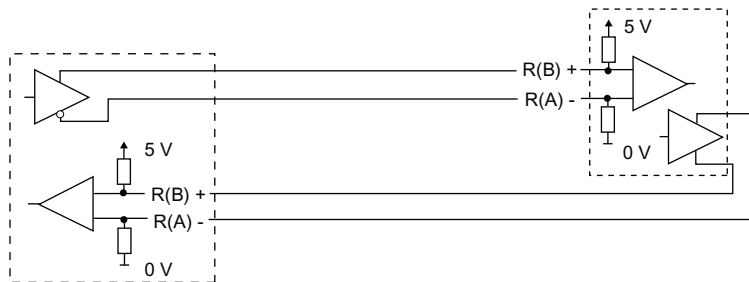
**Cas 1 : RS422 avec détection de rupture de câble**

- Mode de fonctionnement : RS422
- Préconfiguration de la ligne de réception : polarisation inverse (polarisé avec  $R(A) > R(B) > 0V$ )
- Rupture de câble : détection de rupture de câble activée (émetteur toujours actif)



**Cas 2 : RS422 sans détection de rupture de câble, tension à l'état passant**

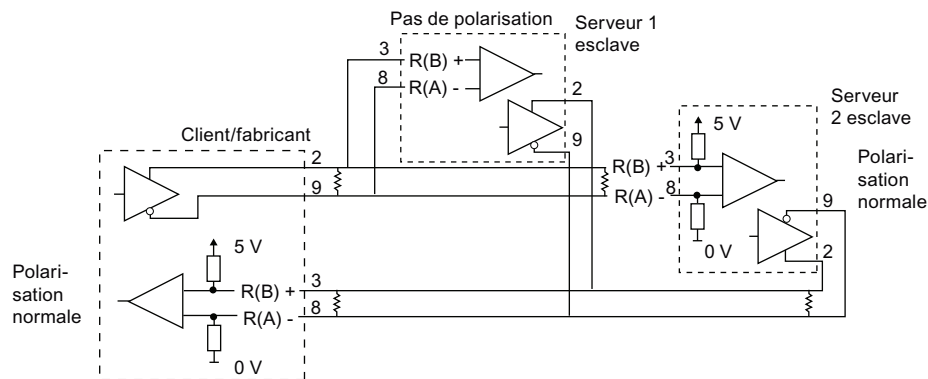
- Mode de fonctionnement : RS422
- Préconfiguration de la ligne de réception : tension à l'état passant (polarisé avec  $R(B) > R(A) > 0V$ )
- Rupture de câble : pas de détection de rupture de câble (émetteur actif uniquement lors de l'émission)



### Cas 3 : RS422 sans détection de rupture de câble, pas de polarisation

- Mode de fonctionnement : RS422
- Préconfiguration de la ligne de réception : pas de polarisation
- Rupture de câble : pas de détection de rupture de câble (émetteur actif uniquement lors de l'émission)

La polarisation et la terminaison sont ajoutées par l'utilisateur aux noeuds d'extrémité du réseau.



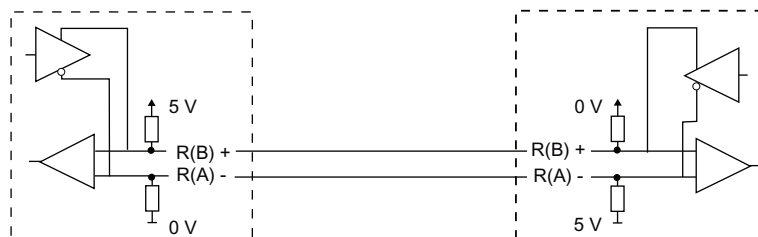
### Configuration de RS485

Il n'y a qu'un mode de fonctionnement pour le mode RS485. Les différents choix pour Préconfiguration de la ligne de réception font référence aux cas détaillés plus loin.

- Semi-duplex (RS485) mode sur 2 fils. Pour Préconfiguration de la ligne de réception :
  - Sélectionnez Aucune lorsque vous fournissez la polarisation et la terminaison (cas 5).
  - Sélectionnez Tension à l'état passant pour utiliser la polarisation et la terminaison internes (cas 4).

### Cas 4 : RS485 : tension à l'état passant

- Mode de fonctionnement : RS485
- Préconfiguration de la ligne de réception : tension à l'état passant (polarisé avec  $R(B) > R(A) > 0V$ )



**Cas 5 : RS485 : pas de polarisation (polarisation externe)**

- Mode de fonctionnement : RS485
- Préconfiguration de la ligne de réception : pas de polarisation (polarisation externe requise)

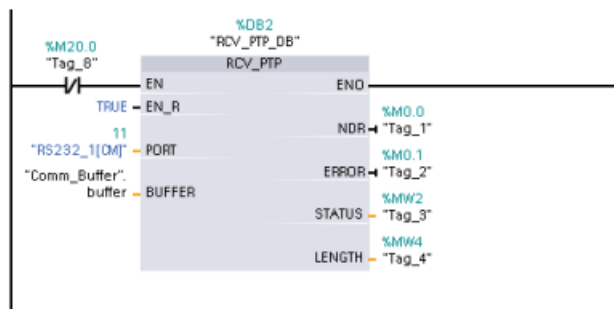


**12.3.5.3 Programmation du programme STEP 7**

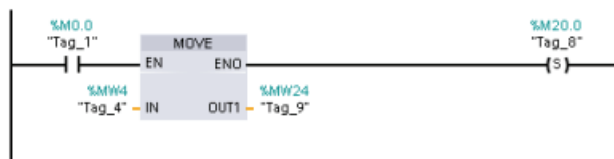
L'exemple de programme utilise un bloc de données global pour la mémoire tampon de communication, une instruction RCV\_PTP (Page 617) pour recevoir des données de l'émulateur de terminal, ainsi qu'une instruction SEND\_PTP (Page 613) pour renvoyer la mémoire tampon en écho à l'émulateur de terminal. Pour programmer l'exemple, ajoutez la configuration du bloc de données et programmez l'OB1 comme décrit ci-dessous.

**Bloc de données global "Comm\_Buffer"** : Créez un bloc de données global (DB) et nommez-le "Comm\_Buffer". Dans le bloc de données, créez une valeur appelée "buffer" et ayant le type de données "array [0 .. 99] of byte".

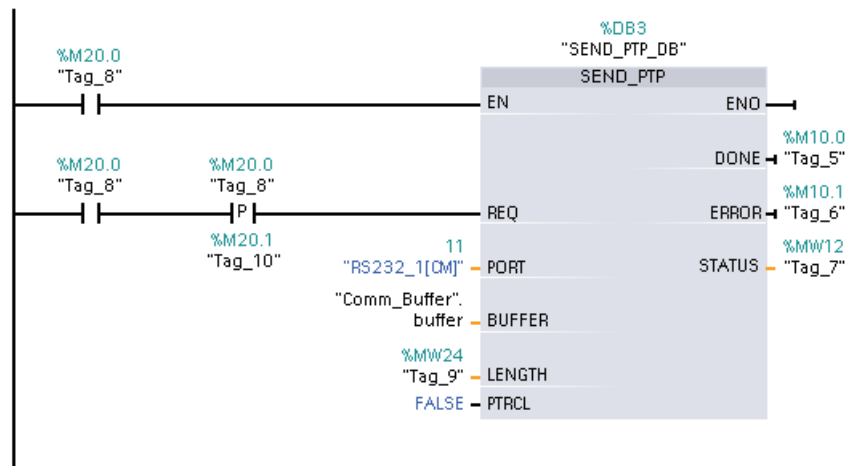
**Réseau 1** : Activez l'instruction RCV\_PTP à chaque fois que SEND\_PTP n'est pas actif. Tag\_8 en MW20.0 signale lorsque l'émission est achevée dans le réseau 4 et lorsque le module de communication est ainsi prêt à recevoir un message.



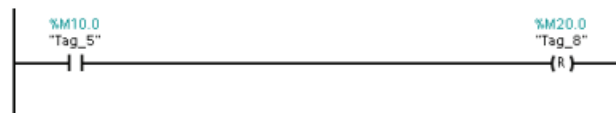
**Réseau 2** : Utilisez la valeur NDR (Tag\_1 en M0.0) définie par l'instruction RCV\_PTP pour copier le nombre d'octets reçus et pour mettre un memento à 1 (Tag\_8 en M20.0) afin de déclencher l'instruction SEND\_PTP.



**Réseau 3 :** Activez l'instruction SEND\_PTP lorsque le memento M20.0 est à 1. Utilisez également ce memento pour mettre l'entrée REQ à VRAI pour un cycle. L'entrée REQ indique à l'instruction SEND\_PTP qu'une nouvelle demande doit être émise. L'entrée REQ ne doit être à VRAI que pendant une exécution de SEND\_PTP. L'instruction SEND\_PTP est exécutée à chaque cycle jusqu'à ce que l'émission s'achève. L'émission est achevée lorsque le dernier octet du message a été émis depuis le CM 1241. Lorsque l'émission est achevée, la sortie DONE (Tag\_5 en M10.0) est mise à VRAI pour une exécution de SEND\_PTP.



**Réseau 4 :** Surveillez la sortie DONE de SEND\_PTP et remettez le memento d'émission à 0 (Tag\_8 en M20.0) lorsque l'opération d'émission est achevée. Lorsque le memento d'émission est à 0, l'instruction RCV\_PTP dans le réseau 1 est activée afin de recevoir le message suivant.



#### 12.3.5.4 Configuration de l'émulateur de terminal

Vous devez configurer l'émulateur de terminal afin qu'il prenne en charge l'exemple de programme. Vous pouvez utiliser presque tout émulateur de terminal sur votre PC, HyperTerminal par exemple. Assurez-vous que l'émulateur de terminal est en mode déconnecté avant d'éditer les paramètres comme suit :

1. Réglez l'émulateur de terminal afin qu'il utilise le port RS232 sur le PC (COM1 normalement).
2. Configurez le port à 9600 bauds, 8 bits de données, sans parité, 1 bit d'arrêt et sans contrôle de flux.
3. Modifiez les paramètres de l'émulateur de terminal afin qu'il émule un terminal ANSI.
4. Réglez la configuration ASCII de l'émulateur de terminal pour l'envoi d'un changement de ligne après chaque ligne (lorsque l'utilisateur appuie sur la touche Entrée).
5. Renvoyez les caractères localement de sorte que l'émulateur de terminal affiche ce qui est tapé.

### 12.3.5.5 Exécution de l'exemple

Procédez comme suit pour exécuter l'exemple de programme :

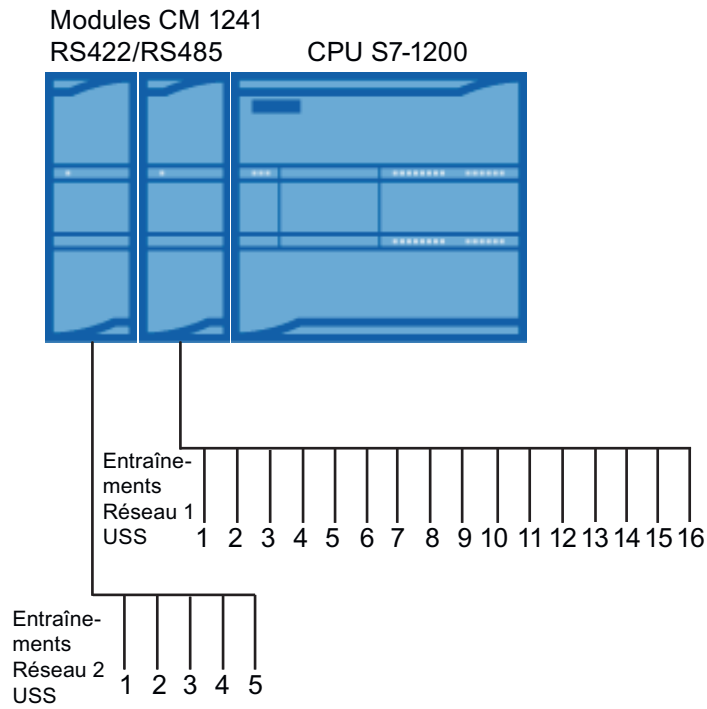
1. Chargez le programme STEP 7 dans la CPU et assurez-vous qu'elle est à l'état MARCHE.
2. Cliquez sur le bouton "connecter" de l'émulateur de terminal pour appliquer les modifications de configuration et ouvrir une session de terminal vers le CM 1241.
3. Tapez des caractères sur le PC et appuyez sur Entrée.

L'émulateur de terminal envoie les caractères au CM 1241 et à la CPU. Le programme de la CPU renvoie alors les caractères en écho à l'émulateur de terminal.

## 12.4 Communication USS (interface série universelle)

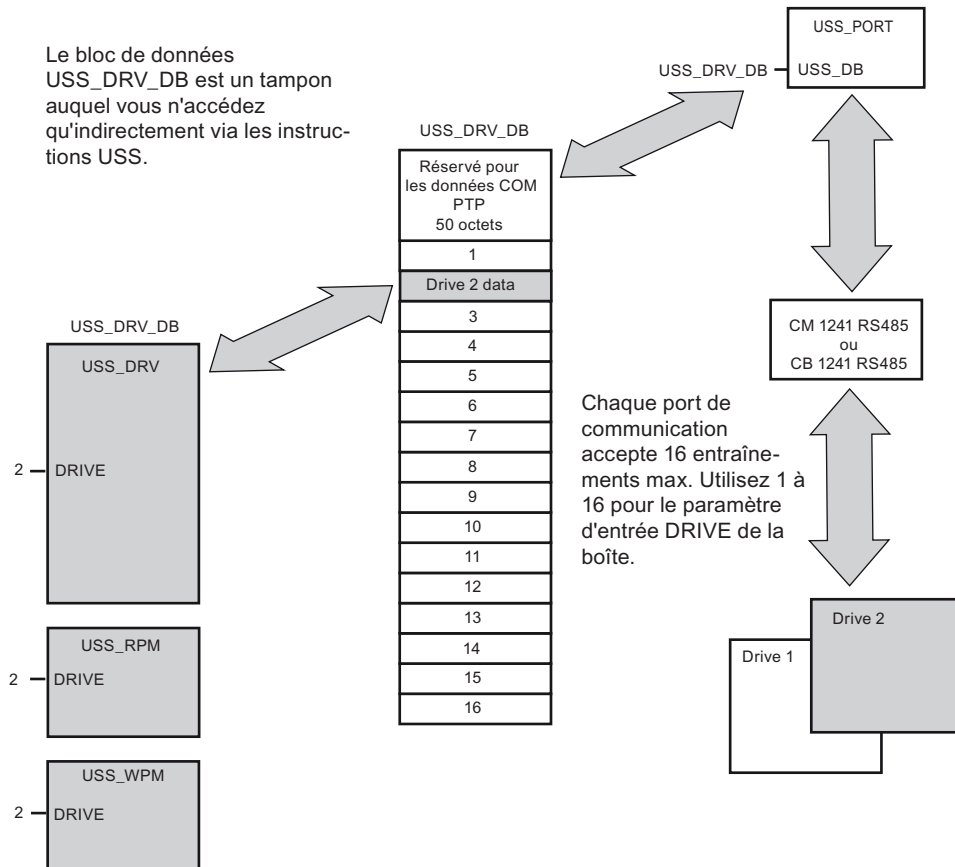
Les instructions USS commandent le fonctionnement d'entraînements de moteur qui prennent en charge le protocole USS (interface série universelle). Vous pouvez utiliser les instructions USS pour communiquer avec plusieurs entraînements via des liaisons RS485 vers des modules de communication CM 1241 RS485 ou un Communication Board CB 1241 RS485. Il est possible d'installer jusqu'à trois modules CM 1241 RS422/RS485 et un CB 1241 RS485 dans une CPU S7-1200. Chaque port RS485 peut faire fonctionner seize entraînements au plus.

Le protocole USS utilise un réseau maître-esclave pour communiquer par le biais d'un bus série. Le maître utilise un paramètre adresse pour envoyer un message à un esclave sélectionné. Un esclave ne peut lui jamais émettre s'il n'y a pas d'abord été invité par un maître. Un transfert de message direct entre les esclaves individuels est impossible. La communication USS opère en mode semi-duplex. L'illustration suivante de USS montre un schéma de réseau pour un exemple d'application d'entraînement.



### 12.4.1 Conditions requises pour l'utilisation du protocole USS

Les quatre instructions USS utilisent un FB et trois FC pour la prise en charge du protocole USS. Un bloc de données d'instance USS\_PORT est utilisé pour chaque réseau USS. Il contient de la mémoire temporaire et des mémoires tampons pour tous les entraînements de ce réseau USS. Les instructions USS partagent les informations dans ce bloc de données.



Tous les entraînements (16 au maximum) connectés à un même port RS485 font partie du même réseau USS. Tous les entraînements connectés à un port RS485 différent font partie d'un réseau USS différent. Chaque réseau USS est géré à l'aide d'un bloc de données unique. Toutes les instructions associées à un même réseau USS doivent partager ce bloc de données. Cela inclut toutes les instructions USS\_DRV, USS\_PORT, USS\_RPM et USS\_WPM utilisées pour commander tous les entraînements dans un même réseau USS.

L'instruction USS\_DRV est un bloc fonctionnel (FB). Lorsque vous placez l'instruction USS\_DRV dans l'éditeur de programmes, la boîte de dialogue "Options d'appel" s'affiche et vous demande d'affecter un DB à ce FB. S'il s'agit de la première instruction USS\_DRV dans ce programme pour ce réseau USS, vous pouvez accepter le DB indiqué par défaut (ou modifier son nom si vous le désirez) et ce nouveau DB est alors créé pour vous. En revanche, s'il ne s'agit pas de la première instruction USS\_DRV pour cette voie, vous devez utiliser la liste déroulante dans la boîte de dialogue "Options d'appel" pour sélectionner le nom de DB précédemment affecté à ce réseau USS.



Les instructions USS\_PORT, USS\_RPM et USS\_WPM sont des fonctions (FC). Il n'y a pas d'affectation de DB lorsque vous placez ces FC dans l'éditeur. En revanche, vous devez affecter le DB approprié à l'entrée "USS\_DB" de ces instructions. Double-cliquez sur le champ de paramètre, puis cliquez sur l'icône d'aide pour voir les noms de DB disponibles.

La fonction USS\_PORT gère la communication effective entre la CPU et les entraînements par le biais du port de communication RS485 point à point. Chaque appel de cette fonction gère une communication avec un entraînement. Votre programme doit appeler cette fonction suffisamment rapidement pour éviter un dépassement du délai de communication par les entraînements. Vous pouvez appeler cette fonction dans un OB de cycle de programme principal ou dans n'importe quel OB d'alarme.

Typiquement, vous appellerez la fonction USS\_PORT dans un OB d'alarme cyclique. Définissez la période de cet OB d'alarme cyclique à environ la moitié de l'intervalle d'appel minimum. Une communication à 1200 bauds devrait, par exemple, utiliser une période d'au plus 350 ms).

Le bloc fonctionnel USS\_DRV permet à votre programme d'accéder à un entraînement spécifié dans le réseau USS. Ses entrées et sorties constituent les états et les commandes pour l'entraînement. S'il y a 16 entraînements dans le réseau, votre programme doit comporter au moins 16 appels de USS\_DRV, un pour chaque entraînement. Vous devez appeler ces blocs à la fréquence nécessaire pour commander le fonctionnement de l'entraînement.

Vous ne pouvez appeler le bloc fonctionnel USS\_DRV que dans un OB de cycle de programme principal.

 **PRUDENCE**

Appelez USS\_DRV, USS\_RPM et USS\_WPM uniquement à partir d'un OB de cycle de programme. La fonction USS\_PORT peut être appelée dans n'importe quel OB, généralement un OB d'alarme cyclique.

N'utilisez pas les instructions USS\_DRV, USS\_RPM ou USS\_WPM dans un OB de priorité supérieure à celle de l'instruction USS\_PORT correspondante. Ainsi, ne placez pas USS\_PORT dans l'OB de cycle programme principal et USS\_RPM dans un OB d'alarme cyclique. L'interruption possible de USS\_PORT peut produire des erreurs inattendues.

Les fonctions USS\_RPM et USS\_WPM lisent et écrivent les paramètres de fonctionnement de l'entraînement distant. Ces paramètres pilotent le fonctionnement interne de l'entraînement. Vous trouverez la définition de ces paramètres dans le manuel de l'entraînement. Votre programme peut contenir autant de ces fonctions que nécessaire, mais une seule demande de lecture ou d'écriture peut être active par entraînement à un moment donné. Vous ne pouvez appeler les fonctions USS\_RPM et USS\_WPM que dans un OB de cycle de programme principal.

### Calcul du temps requis pour communiquer avec l'entraînement

La communication avec l'entraînement est asynchrone par rapport au cycle S7-1200. Typiquement, le S7-1200 exécute plusieurs cycles avant qu'une transaction de communication avec un entraînement s'achève.

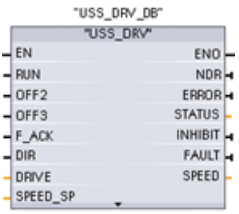

L'intervalle USS\_PORT est le temps requis pour une transaction avec un entraînement. Le tableau ci-dessous montre l'intervalle USS\_PORT minimum pour chaque vitesse de transmission. Appeler la fonction USS\_PORT plus fréquemment que l'intervalle USS\_PORT n'augmente pas le nombre de transactions. Le délai d'attente de l'entraînement est la quantité de temps qui pourrait être utilisée pour une transaction si des erreurs de communication entraînaient 3 tentatives pour achever la transaction. Par défaut, la bibliothèque du protocole USS exécute automatiquement jusqu'à 2 nouvelles tentatives pour chaque transaction.

Tableau 12- 34 Calcul des exigences en temps

Débit	Intervalle d'appel USS_PORT minimum calculé ( millisecondes )	Délai d'attente de message par entraînement ( millisecondes )
1200	790	2370
2400	405	1215
4800	212.5	638
9600	116.3	349
19200	68.2	205
38400	44.1	133
57600	36.1	109
115200	28.1	85

## 12.4.2 Instruction USS\_DRV

Tableau 12- 35 Instruction USS\_DRV

CONT/LOG	SCL	Description
<p>Vue par défaut</p>  <p>Vue étendue</p> 	<pre>"USS_DRV_DB" (   RUN:=_bool_in_,   OFF2:=_bool_in_,   OFF3:=_bool_in_,   F_ACK:=_bool_in_,   DIR:=_bool_in_,   DRIVE:=_usint_in_,   PZD_LEN:=_usint_in_,   SPEED_SP:=_real_in_,   CTRL3:=_word_in_,   CTRL4:=_word_in_,   CTRL5:=_word_in_,   CTRL6:=_word_in_,   CTRL7:=_word_in_,   CTRL8:=_word_in_,   NDR=&gt;_bool_out_,   ERROR=&gt;_bool_out_,   STATUS=&gt;_word_out_,   RUN_EN=&gt;_bool_out_,   D_DIR=&gt;_bool_out_,   INHIBIT=&gt;_bool_out_,   FAULT=&gt;_bool_out_,   SPEED=&gt;_real_out_,   STATUS1=&gt;_word_out_,   STATUS3=&gt;_word_out_,   STATUS4=&gt;_word_out_,   STATUS5=&gt;_word_out_,   STATUS6=&gt;_word_out_,   STATUS7=&gt;_word_out_,   STATUS8=&gt;_word_out_);</pre>	<p>L'instruction USS_DRV échange des données avec un entraînement en créant des messages de demande et en interprétant les messages de réponse de l'entraînement. Il faut utiliser un bloc fonctionnel distinct pour chaque entraînement, mais toutes les fonctions USS associées à un réseau USS et un port de communication point à point doivent utiliser le même bloc de données d'instance. Vous devez créer le nom du DB lorsque vous insérez la première instruction USS_DRV ; vous faites ensuite appel au DB créé par la première utilisation de l'instruction.</p> <p>STEP 7 crée automatiquement le DB lorsque vous insérez l'instruction.</p>

1 CONT et LOG : Affichez le détail de la boîte pour visualiser tous les paramètres en cliquant au bas de la boîte. Les connecteurs de paramètre qui sont grisés sont facultatifs ; l'affectation de paramètre n'y est pas exigée.

Tableau 12- 36 Types de données pour les paramètres

Paramètre et type		Type de données	Description
RUN	IN	Bool	Bit de démarrage d'entraînement. Lorsqu'elle est vraie, cette entrée permet le fonctionnement de l'entraînement à la vitesse prédéfinie. Lorsque RUN passe à FALSE pendant le fonctionnement d'un entraînement , le moteur décélère jusqu'à l'arrêt complet. Ce comportement diffère de l'arrêt électrique (OFF2) ou du freinage du moteur (OFF3).
OFF2	IN	Bool	Bit d'arrêt électrique. Lorsqu'il est faux, ce bit permet à l'entraînement de tourner en roue libre jusqu'à l'arrêt sans freinage.

12.4 Communication USS (interface série universelle)

Paramètre et type		Type de données	Description
OFF3	IN	Bool	Bit d'arrêt rapide. Lorsqu'il est faux, ce bit provoque un arrêt rapide par freinage de l'entraînement plutôt que par passage en roue libre jusqu'à l'arrêt.
F_ACK	IN	Bool	Bit d'acquiescement de défaut. Ce bit est mis à 1 pour réinitialiser le bit de défaut d'un entraînement. Le bit est mis à 1 après que le défaut a été corrigé afin de signaler à l'entraînement qu'il n'a plus besoin de signaler le défaut précédent.
DIR	IN	Bool	Commande du sens de l'entraînement. Ce bit est mis à 1 pour indiquer que le sens est la marche avant (pour SPEED_SP positif).
DRIVE	IN	USInt	Adresse de l'entraînement. Cette entrée est l'adresse de l'entraînement USS. La plage valide va de entraînement 1 à entraînement 16.
PZD_LEN	IN	USInt	Longueur en mots. Il s'agit du nombre de mots des données PZD. Les valeurs valides sont 2, 4, 6 ou 8 mots. La valeur par défaut est 2.
SPEED_SP	IN	Real	Consigne de vitesse. Il s'agit de la vitesse de l'entraînement sous forme de pourcentage de la fréquence configurée. Une valeur positive correspond à une marche avant (lorsque DIR est vrai). La plage valide va de 200,00 à -200,00.
CTRL3	IN	Word	Mot de commande 3. Valeur écrite dans un paramètre configurable de l'entraînement. Vous devez le configurer dans l'entraînement. Il s'agit d'un paramètre facultatif.
CTRL4	IN	Word	Mot de commande 4. Valeur écrite dans un paramètre configurable de l'entraînement. Vous devez le configurer dans l'entraînement. Il s'agit d'un paramètre facultatif.
CTRL5	IN	Word	Mot de commande 5. Valeur écrite dans un paramètre configurable de l'entraînement. Vous devez le configurer dans l'entraînement. Il s'agit d'un paramètre facultatif.
CTRL6	IN	Word	Mot de commande 6. Valeur écrite dans un paramètre configurable de l'entraînement. Vous devez le configurer dans l'entraînement. Il s'agit d'un paramètre facultatif.
CTRL7	IN	Word	Mot de commande 7. Valeur écrite dans un paramètre configurable de l'entraînement. Vous devez le configurer dans l'entraînement. Il s'agit d'un paramètre facultatif.
CTRL8	IN	Word	Mot de commande 8. Valeur écrite dans un paramètre configurable de l'entraînement. Vous devez le configurer dans l'entraînement. Il s'agit d'un paramètre facultatif.
NDR	OUT	Bool	Nouvelles données prêtes (New data ready) : Lorsqu'il est vrai, ce bit signale que les sorties contiennent des données pour une nouvelle demande de communication.
ERROR	OUT	Bool	Une erreur s'est produite. Lorsqu'elle est vraie, cette sortie signale qu'une erreur s'est produite et que la sortie STATUS est valide. Toutes les autres sorties sont mises à zéro en cas d'erreur. Les erreurs de communication sont signalées uniquement au niveau des sorties ERROR et STATUS de l'instruction USS_PORT.
STATUS	OUT	Word	La valeur STATUS de la demande indique le résultat du cycle. Il ne s'agit pas d'un mot d'état provenant de l'entraînement.
RUN_EN	OUT	Bool	Marche activée. Ce bit signale si l'entraînement fonctionne ou non.
D_DIR	OUT	Bool	Sens de l'entraînement. Ce bit signale si l'entraînement fonctionne en marche avant ou non.

Paramètre et type		Type de données	Description
INHIBIT	OUT	Bool	Entraînement inhibé. Ce bit indique l'état du bit d'inhibition dans l'entraînement.
FAULT	OUT	Bool	Défaut de l'entraînement. Ce bit signale que l'entraînement a enregistré un défaut. Vous devez résoudre le problème puis mettre le bit F_ACK à 1 pour effacer ce bit lorsqu'il a été mis à 1.
SPEED	OUT	Real	Vitesse en cours de l'entraînement (valeur mise à l'échelle du mot d'état d'entraînement 2). Il s'agit de la valeur de la vitesse de l'entraînement sous forme de pourcentage de la vitesse configurée.
STATUS1	OUT	Word	Mot d'état 1 de l'entraînement. Cette valeur contient des bits d'état fixes d'un entraînement.
STATUS3	OUT	Word	Mot d'état 3 de l'entraînement. Cette valeur contient un mot d'état configurable de l'entraînement.
STATUS4	OUT	Word	Mot d'état 4 de l'entraînement. Cette valeur contient un mot d'état configurable de l'entraînement.
STATUS5	OUT	Word	Mot d'état 5 de l'entraînement. Cette valeur contient un mot d'état configurable de l'entraînement.
STATUS6	OUT	Word	Mot d'état 6 de l'entraînement. Cette valeur contient un mot d'état configurable de l'entraînement.
STATUS7	OUT	Word	Mot d'état 7 de l'entraînement. Cette valeur contient un mot d'état configurable de l'entraînement.
STATUS8	OUT	Word	Mot d'état 8 de l'entraînement. Cette valeur contient un mot d'état configurable de l'entraînement.

Lors de l'exécution de la première instruction USS\_DRV, l'entraînement indiqué par l'adresse USS (paramètre DRIVE) est initialisé dans le DB d'instance. Après cette initialisation, les exécutions suivantes de USS\_PORT peuvent commencer la communication avec l'entraînement à ce numéro d'entraînement.

La modification du numéro d'entraînement nécessite un passage de la CPU de ARRET à MARCHE afin d'initialiser le DB d'instance. Les paramètres d'entrée sont écrits dans la mémoire tampon USS de message à émettre et les sorties sont lues d'une mémoire tampon de réponse valide "précédente" s'il en existe une. Il n'y a pas de transmission de données pendant l'exécution de USS\_DRV. Les entraînements communiquent lorsque USS\_PORT est exécuté. USS\_DRV ne fait que configurer les messages à envoyer et interpréter les données éventuelles reçues d'une précédente demande.

Vous pouvez commander le sens de rotation de l'entraînement via l'entrée DIR (Bool) ou via le signe (positif ou négatif) de l'entrée SPEED\_SP (Real). Le tableau suivant montre comment ces entrées déterminent ensemble le sens de l'entraînement, dans l'hypothèse où le moteur est câblé pour une rotation avant.

Tableau 12- 37 Interaction entre les paramètres SPEED\_SP et DIR

SPEED_SP	DIR	Sens de rotation de l'entraînement
Valeur > 0	0	Arrière
Valeur > 0	1	Avant
Valeur < 0	0	Avant
Valeur < 0	1	Arrière

### 12.4.3 Instruction USS\_PORT

Tableau 12- 38 Instruction USS\_PORT

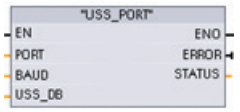
CONT/LOG	SCL	Description
	<pre>USS_PORT (     PORT:= _uint_in_,     BAUD:= _dint_in_,     ERROR=&gt; _bool_out_,     STATUS=&gt; _word_out_,     USS DB:= _fbtref inout_ );</pre>	<p>L'instruction USS_PORT gère la communication par le biais d'un réseau USS.</p>

Tableau 12- 39 Types de données pour les paramètres

Paramètre et type		Type de données	Description
PORT	IN	Port	Une fois que vous avez installé et configuré un appareil de communication CM ou CB, l'identificateur de port apparaît dans la liste déroulante d'aide pour les paramètres disponible au niveau du connecteur PORT de la boîte. La valeur de port CM ou CB affectée est la propriété de configuration d'appareil "identificateur matériel". Le nom symbolique du port est défini dans l'onglet "Constantes système" de la table de variables API.
BAUD	IN	DInt	Vitesse de transmission à utiliser pour la communication USS
USS_DB	INOUT	USS_BASE	Nom du DB d'instance créé et initialisé lorsque vous insérez une instruction USS_DRV dans votre programme.
ERROR	OUT	Bool	Lorsqu'elle est vraie, cette sortie signale qu'une erreur s'est produite et que la sortie STATUS est valide.
STATUS	OUT	Word	La valeur STATUS de la demande indique le résultat du cycle ou de l'initialisation. Des informations supplémentaires sont disponibles dans la variable "USS_Extended_Error" pour certains codes d'état.

Typiquement, il n'y a qu'une instruction USS\_PORT par port de communication point à point dans le programme et chaque appel de cette fonction gère une transmission vers ou depuis un entraînement unique. Toutes les fonctions USS associées à un réseau USS et un port de communication point à point doivent utiliser le même bloc de données d'instance.

Votre programme doit exécuter la fonction USS\_PORT suffisamment souvent pour empêcher les dépassements de délai des entraînements. USS\_PORT est généralement appelé dans un OB d'alarme cyclique pour éviter les dépassements de délai des entraînements et mettre les mises à jour les plus récentes des données USS à disposition des appels USS\_DRV.

### 12.4.4 Instruction USS\_RPM

Tableau 12- 40 Instruction USS\_RPM

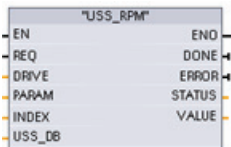
CONT/LOG	SCL	Description
	<pre>USS_RPM(REQ:=_bool_in_,         DRIVE:=_usint_in_,         PARAM:=_uint_in_,         INDEX:=_uint_in_,         DONE=&gt;_bool_out_,         ERROR=&gt;_bool_out_,         STATUS=&gt;_word_out_,         VALUE=&gt;_variant_out_,         USS_DB:=_fbtref_inout_);</pre>	<p>L'instruction USS_RPM lit un paramètre provenant de l'entraînement. Toutes les fonctions USS associées à un réseau USS et un port de communication point à point doivent utiliser le même bloc de données. Il faut appeler USS_RPM dans un OB de cycle de programme principal.</p>

Tableau 12- 41 Types de données pour les paramètres

Type de paramètre		Type de données	Description
REQ	IN	Bool	Envoi de la demande. Lorsqu'elle est vraie, l'entrée REQ indique qu'une nouvelle demande de lecture est désirée. Il n'en est pas tenu compte si la demande pour ce paramètre est déjà en attente.
DRIVE	IN	USInt	Adresse de l'entraînement. DRIVE est l'adresse de l'entraînement USS. La plage valide va de entraînement 1 à entraînement 16.
PARAM	IN	UInt	Numéro du paramètre. PARAM désigne le paramètre d'entraînement à écrire. La plage de ce paramètre est comprise entre 0 et 2047. Sur certains entraînements, l'octet de poids fort peut accéder à des valeurs PARAM supérieures à 2047. Vous trouverez plus de détails sur la manière d'accéder à une plage étendue dans le manuel de l'entraînement.
INDEX	IN	UInt	Indice du paramètre. INDEX désigne l'indice du paramètre d'entraînement à écrire. Valeur de 16 bits dans laquelle l'octet de poids faible est la valeur d'indice réelle avec une plage de 0 à 255. L'octet de poids fort peut être utilisé par l'entraînement et lui est spécifique. Consultez le manuel de l'entraînement pour plus de détails.
USS_DB	INOUT	USS_BASE	Nom du DB d'instance créé et initialisé lorsque vous insérez une instruction USS_DRV dans votre programme.
VALUE	IN	Word, Int, UInt, DWord, DInt, UDIInt, Real	Il s'agit de la valeur du paramètre qui a été lue et qui est uniquement valide lorsque le bit DONE est vrai.
DONE <sup>1</sup>	OUT	Bool	S'il est vrai, ce bit signale que la sortie VALUE contient la valeur du paramètre lue suite à la demande précédente. Cet bit est mis à 1 lorsque USS_DRV voit les données de réponse de lecture provenant de l'entraînement. Ce bit est mis à 0 lorsque vous demandez des données de réponse via une autre interrogation USS_RPM ou lors du deuxième des deux appels suivants de USS_DRV.

12.4 Communication USS (interface série universelle)

Type de paramètre		Type de données	Description
ERROR	OUT	Bool	Une erreur s'est produite. Lorsqu'elle est vraie, la sortie ERROR signale qu'une erreur s'est produite et que la sortie STATUS est valide. Toutes les autres sorties sont mises à zéro en cas d'erreur. Les erreurs de communication sont signalées uniquement au niveau des sorties ERROR et STATUS de l'instruction USS_PORT.
STATUS	OUT	Word	STATUS indique le résultat de la demande de lecture. Des informations supplémentaires sont disponibles dans la variable "USS_Extended_Error" pour certains codes d'état.

<sup>1</sup> Le bit DONE indique que des données valides ont été lues dans l'entraînement de moteur référencé et transmises à la CPU. Il ne signifie pas que la bibliothèque USS est capable de lire immédiatement un autre paramètre. Une demande PKW vide doit être envoyée à l'entraînement et doit également être acquittée par l'instruction avant que la voie de paramètre pour l'entraînement spécifique ne soit mise à disposition. L'appel immédiat d'une instruction USS\_RPM ou USS\_WPM pour l'entraînement spécifié provoque une erreur 0x818A.

12.4.5 Instruction USS\_WPM

**Remarque**

**Opérations d'écriture en EEPROM (pour l'EEPROM à l'intérieur d'un entraînement USS)**

N'utilisez pas la fonction d'écriture permanente en EEPROM de manière abusive. Réduisez au minimum le nombre d'opérations d'écriture en EEPROM pour allonger la durée de vie de cette dernière.

Tableau 12- 42 Instruction USS\_WPM

CONT/LOG	SCL	Description
	<pre> USS_WPM(REQ:=_bool_in_,         DRIVE:=_usint_in_,         PARAM:=_uint_in_,         INDEX:=_uint_in_,         EEPROM:=_bool_in_,         VALUE:=_variant_in_,         DONE=&gt;_bool_out_,         ERROR=&gt;_bool_out_,         STATUS=&gt;_word_out_,         USS_DB:=_fbtref_inout_);                     </pre>	<p>L'instruction USS_WPM modifie un paramètre dans l'entraînement. Toutes les fonctions USS associées à un réseau USS et un port de communication point à point doivent utiliser le même bloc de données.</p> <p>Il faut appeler USS_WPM dans un OB de cycle de programme principal.</p>



Tableau 12- 43 Types de données pour les paramètres

Paramètre et type		Type de données	Description
REQ	IN	Bool	Envoi de la demande. Lorsqu'elle est vraie, l'entrée REQ indique qu'une nouvelle demande d'écriture est désirée. Il n'en est pas tenu compte si la demande pour ce paramètre est déjà en attente.
DRIVE	IN	USInt	Adresse de l'entraînement. DRIVE est l'adresse de l'entraînement USS. La plage valide va de entraînement 1 à entraînement 16.
PARAM	IN	UInt	Numéro du paramètre. PARAM désigne le paramètre d'entraînement à écrire. La plage de ce paramètre est comprise entre 0 et 2047. Sur certains entraînements, l'octet de poids fort peut accéder à des valeurs PARAM supérieures à 2047. Vous trouverez plus de détails sur la manière d'accéder à une plage étendue dans le manuel de l'entraînement.
INDEX	IN	UInt	Indice du paramètre. INDEX désigne l'indice du paramètre d'entraînement à écrire. Valeur de 16 bits dans laquelle l'octet de poids faible est la valeur d'indice réelle avec une plage de 0 à 255. L'octet de poids fort peut être utilisé par l'entraînement et lui est spécifique. Consultez le manuel de l'entraînement pour plus de détails.
EEPROM	IN	Bool	Stockage dans l'EEPROM de l'entraînement. Lorsque cette entrée est vraie, une transaction d'écriture dans le paramètre de l'entraînement est stockée dans l'EEPROM de l'entraînement. Lorsqu'elle est fausse, l'écriture est temporaire et ne sera pas conservée en cas de mise hors tension puis sous tension de l'entraînement.
VALUE	IN	Word, Int, UInt, DWord, DInt, UInt, Real	Valeur du paramètre qui doit être écrite. Elle doit être valide lors de la transition de REQ.
USS_DB	INOUT	USS_BASE	Nom du DB d'instance créé et initialisé lorsque vous insérez une instruction USS_DRV dans votre programme.
DONE <sup>1</sup>	OUT	Bool	S'il est vrai, le bit DONE signale que l'entrée VALUE a été écrite dans l'entraînement. Ce bit est mis à 1 lorsque USS_DRV voit les données de réponse d'écriture provenant de l'entraînement. Ce bit est mis à 0 lorsque vous demandez les données de réponse via une autre interrogation USS_WPM ou lors du deuxième des deux appels suivants de USS_DRV.
ERROR	OUT	Bool	Lorsqu'elle est vraie, la sortie ERROR signale qu'une erreur s'est produite et que la sortie STATUS est valide. Toutes les autres sorties sont mises à zéro en cas d'erreur. Les erreurs de communication sont signalées uniquement au niveau des sorties ERROR et STATUS de l'instruction USS_PORT.
STATUS	OUT	Word	STATUS indique le résultat de la demande d'écriture. Des informations supplémentaires sont disponibles dans la variable "USS_Extended_Error" pour certains codes d'état.

<sup>1</sup> Le bit DONE indique que des données valides ont été lues dans l'entraînement de moteur référencé et transmises à la CPU. Il ne signifie pas que la bibliothèque USS est capable de lire immédiatement un autre paramètre. Une demande PKW vide doit être envoyée à l'entraînement et doit également être acquittée par l'instruction avant que la voie de paramètre pour l'entraînement spécifique ne soit mise à disposition. L'appel immédiat d'une instruction USS\_RPM ou USS\_WPM pour l'entraînement spécifié provoque une erreur 0x818A.

### 12.4.6 Codes d'état USS

Les codes d'état des instructions USS sont renvoyés dans la sortie STATUS des fonctions USS.

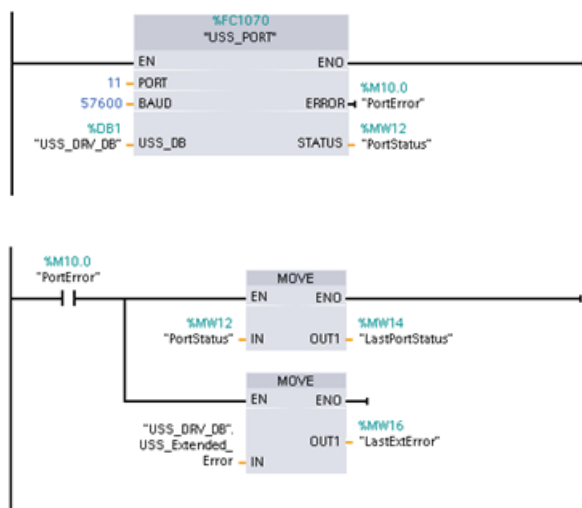
Tableau 12- 44 Codes STATUS <sup>1</sup>

STATUS (W#16#....)	Description
0000	Pas d'erreur
8180	La longueur de la réponse de l'entraînement ne correspondait pas aux caractères reçus de l'entraînement. Le numéro de l'entraînement où s'est produite l'erreur est renvoyé dans la variable "USS_Extended_Error". Consultez la description des erreurs étendues après ce tableau.
8181	Le paramètre VALUE n'avait pas le type de données Word, Real ou DWord.
8182	L'utilisateur a fourni un mot pour une valeur de paramètre et a reçu un double mot ou un réel dans la réponse provenant de l'entraînement.
8183	L'utilisateur a fourni un double mot ou un réel pour une valeur de paramètre et a reçu un mot dans la réponse provenant de l'entraînement.
8184	Le télégramme de réponse de l'entraînement contenait un total de contrôle erroné. Le numéro de l'entraînement où s'est produite l'erreur est renvoyé dans la variable "USS_Extended_Error". Consultez la description des erreurs étendues après ce tableau.
8185	Adresse d'entraînement illicite (plage d'adresses d'entraînement valide : 1 à 16)
8186	La consigne de vitesse est hors de la plage valide (plage de consignes de vitesse valide : -200% à 200%).
8187	Un entraînement de numéro erroné a répondu à la demande envoyée. Le numéro de l'entraînement où s'est produite l'erreur est renvoyé dans la variable "USS_Extended_Error". Consultez la description des erreurs étendues après ce tableau.
8188	Longueur de mot PZD illicite (plage valide = 2, 4, 6 ou 8 mots)
8189	Vitesse de transmission illicite
818A	La voie de demande de paramètre est utilisée par une autre demande pour cet entraînement.
818B	L'entraînement n'a pas répondu aux demandes et nouvelles tentatives. Le numéro de l'entraînement où s'est produite l'erreur est renvoyé dans la variable "USS_Extended_Error". Consultez la description des erreurs étendues après ce tableau.
818C	L'entraînement a renvoyé une erreur étendue à une opération de demande de paramètre. Consultez la description des erreurs étendues après ce tableau.
818D	L'entraînement a renvoyé une erreur d'accès illicite à une opération de demande de paramètre. Consultez le manuel de l'entraînement pour savoir pourquoi l'accès au paramètre est éventuellement limité.
818E	L'entraînement n'a pas été initialisé. Ce code d'erreur est renvoyé à USS_RPM ou USS_WPM lorsque USS_DRV pour cet entraînement n'a pas été appelé au moins une fois. Cela empêche l'initialisation au premier cycle de USS_DRV d'écraser une demande de lecture ou d'écriture de paramètre en attente puisque l'entraînement est initialisé en tant que nouvelle entrée. Pour corriger cette erreur, appelez USS_DRV pour ce numéro d'entraînement.
80Ax-80Fx	Erreurs spécifiques renvoyées par les FB de communication point à point appelés par la bibliothèque USS. Ces valeurs de codes d'erreur ne sont pas modifiées par la bibliothèque USS et sont définies dans les descriptions des instructions point à point.

<sup>1</sup> En plus des erreurs d'instructions USS présentées ci-avant, des erreurs peuvent être renvoyées par les instructions de communication point à point (Page 603) sous-jacentes.

Des informations supplémentaires sont fournies dans la variable "USS\_Extended\_Error" du DB d'instance de USS\_DRV pour certains codes d'état STATUS. Pour les codes hexadécimaux STATUS 8180, 8184, 8187 et 818B, USS\_Extended\_Error contient le numéro de l'entraînement où l'erreur de communication s'est produite. Pour le code hexadécimal STATUS 818C, USS\_Extended\_Error contient un code d'erreur d'entraînement provenant de l'entraînement lié à l'utilisation d'une instruction USS\_RPM ou USS\_WPM.

Les erreurs de communication (STATUS = 16#818B) sont signalées uniquement dans l'instruction USS\_PORT et non dans l'instruction USS\_DRV. Par exemple, si le réseau n'est pas muni de résistances de terminaison correctes, un entraînement peut passer en MARCHE mais l'instruction USS\_DRV montrera uniquement des 0 pour les paramètres de sortie. Dans ce cas, vous pouvez uniquement détecter l'erreur de communication dans l'instruction USS\_PORT. Comme cette erreur n'est visible que pendant un cycle, vous devez ajouter une logique de capture comme illustré dans l'exemple suivant. Dans cet exemple, lorsque le bit d'erreur de l'instruction USS\_PORT est à VRAI, les valeurs de STATUS et de USS\_Extended\_Error sont sauvegardées en mémoire M. Le numéro de l'entraînement est placé dans la variable USS\_Extended\_Error lorsque la valeur hexadécimale du code STATUS est 8180, 8184, 8187 ou 818B.



**Réseau 1 :** Les valeurs de l'état du port "PortStatus" et du code d'erreur étendue "USS\_DRV\_DB". USS\_Extended\_Error ne sont valables que pendant un cycle du programme. Il faut les capturer en vue d'un traitement ultérieur.

**Réseau 2 :** Le contact "PortError" déclenche la mémorisation de la valeur "PortStatus" dans "LastPortStatus" et de la valeur "USS\_DRV\_DB". USS\_Extended\_Error dans "LastExtError".

Les entraînements USS acceptent l'accès en lecture et écriture aux paramètres internes d'un entraînement. Cette caractéristique permet la commande et la configuration à distance de l'entraînement. Les opérations d'accès aux paramètres d'entraînement peuvent échouer suite à des erreurs telles que des valeurs hors plage ou des demandes illicites pour le mode en cours d'un entraînement. L'entraînement génère un code d'erreur qui est renvoyé dans la variable "USS\_Extended\_Error". Ce code d'erreur ne vaut que pour la dernière exécution d'une instruction USS\_RPM ou USS\_WPM. Le code d'erreur d'entraînement est placé dans la variable "USS\_Extended\_Error" lorsque la valeur de STATUS code est 818C hexadécimal. La valeur du code d'erreur de "USS\_Extended\_Error" dépend du modèle d'entraînement. Le manuel de l'entraînement contient une description des codes des erreurs étendues pour les opérations de lecture et d'écriture de paramètres.

## 12.4.7 Informations générales sur la configuration d'un entraînement

### Exigences générales pour la configuration d'un entraînement

- Les entraînements doivent être paramétrés pour utiliser 4 mots PKW.
- Les entraînements peuvent être configurés pour 2, 4, 6 ou 8 mots PZD.
- Le nombre de mots PZD dans l'entraînement doit correspondre à l'entrée PZD\_LEN dans l'instruction USS\_DRV de cet entraînement.
- La vitesse de transmission dans tous les entraînements doit coïncider avec l'entrée BAUD dans l'instruction USS\_PORT.
- La commande à distance doit être activée pour l'entraînement.
- La consigne de fréquence à USS sur la liaison COM doit être sélectionnée pour l'entraînement.
- L'adresse de l'entraînement doit être définie entre 1 et 16 et correspondre à l'entrée DRIVE dans le bloc USS\_DRV pour cet entraînement.
- La commande du sens de l'entraînement doit utiliser la polarité de la consigne de l'entraînement.
- Le réseau RS485 doit être muni de résistances de terminaison appropriées.

### Connexion d'un entraînement MicroMaster

Ces informations sur les entraînements MicroMaster SIEMENS sont données à titre d'exemple. Pour les autres entraînements, consultez les instructions de configuration dans le manuel de l'entraînement.

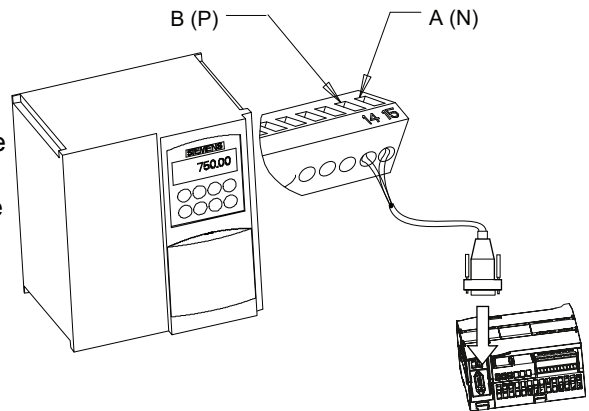
Pour effectuer la connexion à un entraînement MicroMaster série 4 (MM4), insérez les extrémités du câble RS485 dans les deux bornes sans vis à bride gainée fournies pour l'exploitation USS. Vous pouvez utiliser un câble et des connecteurs PROFIBUS standard pour connecter le S7-1200.

#### PRUDENCE

**Le fait d'interconnecter des équipements ayant des potentiels de référence différents peut provoquer des flux de courant indésirables via le câble de liaison.**

Ces courants indésirables peuvent entraîner des erreurs de programmation ou endommager l'équipement. Assurez-vous que tous les matériels que vous allez connecter avec un câble de communication partagent un même potentiel de référence ou qu'ils sont isolés afin d'empêcher les flux de courant indésirables. Le blindage doit être relié à la terre du châssis ou à la broche 1 du connecteur à 9 broches. Nous vous recommandons de relier la borne de câblage 2-0V de l'entraînement MicroMaster à la terre du châssis.

Il faut insérer les deux fils à l'extrémité opposée du câble RS485 dans les borniers de l'entraînement MM4. Pour brancher le câble sur un entraînement MM4, vous devez enlever les volets de l'entraînement afin d'accéder aux borniers. Vous trouverez dans le guide de l'utilisateur de l'entraînement MM4 plus d'informations sur la façon d'ôter les volets de protection de votre entraînement spécifique.



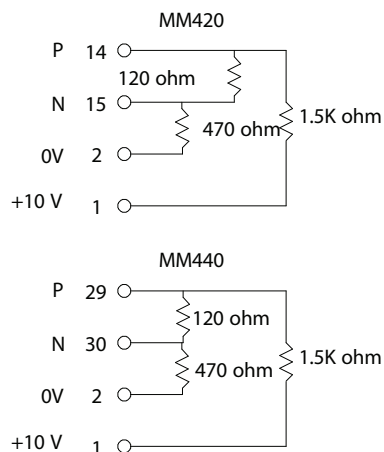
Les connexions des borniers sont désignées numériquement. Avec un connecteur PROFIBUS côté S7-1200, connectez la borne A du câble à la borne 15 d'un entraînement MM420 ou à la borne 30 d'un entraînement MM440. Connectez la borne B du connecteur de câble à la borne 14 de l'entraînement MM420 ou à la borne 29 de l'entraînement MM440.

Si le S7-1200 constitue un noeud de terminaison dans le réseau ou s'il s'agit d'une liaison point à point, il faut utiliser les bornes A1 et B1 - et non les bornes A2 et B2 - du connecteur puisqu'elles permettent de régler les paramètres de terminaison (par exemple, avec le type de connecteur DP 6ES7 972-0BA40-0X40).

**! PRUDENCE**

Veillez à replacer correctement les volets de protection des entraînements avant de mettre l'unité sous tension.

Si l'entraînement est configuré comme le noeud de terminaison dans le réseau, des résistances de terminaison et de polarisation doivent également être raccordées aux bornes appropriées. Ce schéma montre des exemples de raccordements d'entraînement MM4 nécessaires pour la terminaison et la polarisation.



### Configuration de l'entraînement MicroMaster série 4

Avant de connecter un entraînement au S7-1200, assurez-vous que l'entraînement a les paramètres système suivants. Servez-vous du clavier de l'entraînement pour définir les paramètres.

1. Restaurez les paramètres d'usine de l'entraînement (facultatif).	P0010=30 P0970=1
Si vous sautez l'étape 1, assurez-vous que ces paramètres sont définis aux valeurs indiquées.	Longueur PZD USS = P2012 indice 0=(2, 4, 6 ou 8) Longueur PKW USS = P2013 indice 0=4
2. Validez l'accès en lecture/écriture à tous les paramètres (mode expert).	P0003=3
3. Vérifiez les paramètres du moteur pour votre entraînement. Ces paramètres varient selon le ou les moteurs utilisés.  Vous devez d'abord définir le paramètre P010 à 1 (mode de mise en marche rapide) pour pouvoir définir les paramètres P304, P305, P307, P310 et P311. Remettez le paramètre P010 à 0 une fois le paramétrage achevé. En effet, les paramètres P304, P305, P307, P310 et P311 peuvent être modifiés uniquement en mode de mise en marche rapide.	P0304=Tension nominale du moteur (V) P0305=Courant nominal du moteur (A) P0307=Puissance nominale du moteur (W) P0310=Fréquence nominale du moteur (Hz) P0311=Vitesse nominale du moteur
4. Définissez le mode de commande à local ou à éloigné.	P0700 indice 0=5
5. Définissez la sélection de la consigne de fréquence à USS sur la liaison COM.	P1000 indice 0=5
6. Temps d'accélération (facultatif) Il s'agit de la durée en secondes nécessaire au moteur pour accélérer jusqu'à la fréquence maximale.	P1120=(0 à 650,00)
7. Temps de décélération (facultatif) Il s'agit de la durée en secondes nécessaire au moteur pour décélérer jusqu'à l'arrêt complet.	P1121=(0 à 650,00)
8. Définissez la fréquence de référence de la liaison série.	P2000=(1 à 650 Hz)
9. Définissez la normalisation USS.	P2009 indice 0=0
10. Définissez le débit de l'interface série RS485 :	P2010 indice 0= 4 (2400 bauds) 5 (4800 bauds) 6 (9600 bauds) 7 (19200 bauds) 8 (38400 bauds) 9 (57600 bauds) 12 (115200 bauds)
11. Entrez l'adresse de l'esclave. Chaque entraînement (31 au maximum) peut être commandé par l'intermédiaire du bus.	P2011 indice 0=(0 à 31)
12. Définissez le délai d'attente de la liaison série. Il s'agit de la durée maximale autorisée entre deux télégrammes de données entrants. Cette fonction sert à désactiver l'inverseur en cas d'échec de communication. Le décompte commence une fois un télégramme de données valable reçu. Si un autre télégramme de données n'est pas reçu pendant l'intervalle de temps indiqué, l'inverseur se déclenche et affiche le code d'erreur F0070. Mettre la valeur à zéro désactive la commande.	P2014 indice 0=(0 à 65 535 ms) 0=délai d'attente désactivé
13. Transférez les données de la mémoire vive en EEPROM.	P0971=1 (démarrer le transfert) : enregistrer en EEPROM les modifications apportées aux valeurs des paramètres.

## 12.5 Communication Modbus

### 12.5.1 Présentation de la communication Modbus RTU et TCP

#### Codes de fonction Modbus

- Une CPU fonctionnant en tant que maître Modbus RTU (ou client Modbus TCP) peut lire/écrire à la fois des données et des états d'E/S dans un esclave Modbus RTU éloigné (ou un serveur Modbus TCP). Les données éloignées peuvent être lues et traitées dans le programme utilisateur.
- Une CPU fonctionnant en tant qu'esclave Modbus RTU (ou serveur Modbus TCP) permet à un appareil superviseur de lire/écrire à la fois des données et des états d'E/S dans une CPU éloignée. L'appareil superviseur peut écrire dans la mémoire CPU éloignée de nouvelles valeurs qui peuvent être traitées dans le programme utilisateur.

Tableau 12- 45 Fonctions de lecture de données : Lecture d'E/S éloignées et de données de programme

Code de fonction Modbus	Fonctions de lecture dans l'esclave (serveur) : adressage standard
01	Lecture de bits de sortie : 1 à 2000 bits par demande
02	Lecture de bits d'entrée : 1 à 2000 bits par demande
03	Lecture de registres de maintien : 1 à 125 mots par demande
04	Lecture de mots d'entrée : 1 à 125 mots par demande

Tableau 12- 46 Fonctions d'écriture de données : Ecriture d'E/S éloignées et modification de données de programme

Code de fonction Modbus	Fonctions d'écriture dans l'esclave (serveur) : adressage standard
05	Ecriture d'un bit de sortie : 1 bit par demande
06	Ecriture d'un registre de maintien : 1 mot par demande
15	Ecriture d'un ou de plusieurs bits de sortie : 1 à 1968 bits par demande
16	Ecriture d'un ou de plusieurs registres de maintien : 1 à 123 mots par demande

- Les fonctions Modbus de codes 08 à 11 fournissent des informations de diagnostic sur la communication avec l'esclave.
- La fonction Modbus de code 0 diffuse un message à tous les esclaves (sans réponse de ces derniers). La fonction de diffusion générale n'est pas disponible pour Modbus TCP, car la communication est orientée liaison.

Tableau 12- 47 Adresses de stations de réseau Modbus

Station		Adresse
Station RTU	Adresse de station standard	1 à 247
	Adresse de station étendue	1 à 65535
Station TCP	Adresse de station	Adresse IP et numéro de port

### Adresses de mémoire Modbus

Le nombre réel d'adresses de mémoire Modbus disponibles dépend du modèle de la CPU, de la taille de la mémoire de travail et de la part de mémoire CPU utilisée par d'autres données de programme. Le tableau suivant indique la valeur nominale de la plage d'adresses.

Tableau 12- 48 Adresses de mémoire Modbus

Station		Plage d'adresses
Station RTU	Adresse de mémoire standard	10K
	Adresse de mémoire étendue	64K
Station TCP	Adresse de mémoire standard	10K

### Communication Modbus RTU

Modbus RTU (Remote Terminal Unit : unité terminale distante) est un protocole de communication de réseau standard qui utilise la liaison électrique RS232 ou RS485 pour le transfert de données série entre dispositifs réseau Modbus. Vous pouvez ajouter des ports de réseau point à point (PtP) à une CPU avec un CM RS232 ou RS485 ou un CB RS485.

Modbus RTU utilise un réseau maître/esclave où toutes les communications sont déclenchées par un maître unique et où les esclaves peuvent uniquement répondre à une demande du maître. Le maître envoie une demande à une adresse d'esclave et seule cette adresse d'esclave répond à la commande.

### Communication Modbus TCP

Modbus TCP (Transmission Control Protocol : protocole de gestion de transmission) est un protocole de communication de réseau standard qui utilise le connecteur PROFINET sur la CPU pour la communication TCP/IP. Aucun module matériel de communication supplémentaire n'est nécessaire.

Modbus TCP utilise des liaisons de la communication ouverte (OUC) comme voie de communication Modbus. Plusieurs liaisons client-serveur peuvent exister en plus de la liaison entre STEP 7 et la CPU. Les liaisons mixtes client et serveur sont prises en charge jusqu'au nombre maximum de liaisons autorisé par le modèle de CPU (Page 452).



---

Chaque liaison MB\_SERVER doit utiliser un DB d'instance et un numéro de port IP uniques. Une seule liaison est acceptée par port IP. Chaque instruction MB\_SERVER (avec son DB d'instance et son port IP uniques) doit être exécutée individuellement pour chaque liaison.

---

#### Remarque

Modbus TCP ne fonctionne correctement qu'avec une version de firmware CPU V1.02 ou plus. Tenter d'exécuter les instructions Modbus avec une version de firmware antérieure provoque une erreur.

---

Un client Modbus TCP (maître) doit gérer la liaison client-serveur à l'aide du paramètre DISCONNECT. Les actions de base Modbus client sont présentées ci-dessous.

1. Etablir une liaison vers l'adresse IP et le numéro de port IP d'un serveur particulier (esclave)
2. Déclencher l'émission client de messages Modbus et recevoir les réponses du serveur
3. Si désiré, déclencher la déconnexion du client et du serveur pour permettre la liaison avec un serveur différent

#### Instructions Modbus RTU dans votre programme

- MB\_COMM\_LOAD : Une exécution de MB\_COMM\_LOAD sert à configurer les paramètres de port point à point, tels que la vitesse de transmission, la parité et le contrôle de flux. Une fois qu'un port CPU est configuré pour le protocole Modbus RTU, il peut être utilisé uniquement par les instructions MB\_MASTER ou MB\_SLAVE.
- MB\_MASTER : L'instruction Modbus maître permet à la CPU d'agir en tant que maître Modbus RTU et de communiquer avec un ou plusieurs esclaves Modbus.
- MB\_SLAVE : L'instruction Modbus esclave permet à la CPU d'agir en tant qu'esclave Modbus RTU et de communiquer avec un maître Modbus.

#### Instructions Modbus TCP dans votre programme

- MB\_CLIENT : Etablir une liaison TCP client-serveur, envoyer un message de commande, recevoir la réponse et gérer la déconnexion du serveur
- MB\_SERVER : Se connecter à un client Modbus TCP sur demande, recevoir un message Modbus et envoyer la réponse

## 12.5.2 Modbus TCP

### 12.5.2.1 MB\_CLIENT (Modbus TCP)

Tableau 12- 49 Instruction MB\_CLIENT

CONT/LOG	SCL	Description
	<pre>"MB_CLIENT_DB" (     REQ:= bool_in_,     DISCONNECT:= bool_in_,     CONNECT_ID:= uint_in_,     IP_OCTET_1:= byte_in_,     IP_OCTET_2:= byte_in_,     IP_OCTET_3:= byte_in_,     IP_OCTET_4:= byte_in_,     IP_PORT:= uint_in_,     MB_MODE:= usint_in_,     MB_DATA_ADDR:= udint_in_,     MB_DATA_LEN:= uint_in_,     DONE=&gt; bool_out_,     BUSY=&gt; bool_out_,     ERROR=&gt; bool_out_,     STATUS=&gt; word_out_,     MB_DATA_PTR:= variant_inout );</pre>	<p>MB_CLIENT permet de communiquer en tant que client Modbus TCP par l'intermédiaire du connecteur PROFINET sur la CPU S7-1200. Aucun module matériel de communication supplémentaire n'est nécessaire.</p> <p>MB_CLIENT permet d'établir une liaison client-serveur, d'envoyer une demande de fonction Modbus, de recevoir une réponse et de gérer la déconnexion d'un serveur Modbus TCP.</p>

Tableau 12- 50 Types de données pour les paramètres

Paramètre et type	Type de données	Description
REQ	In	Bool FALSE = Aucune demande de communication Modbus TRUE = Demande de communication avec un serveur Modbus TCP
DISCONNECT	IN	Bool Le paramètre DISCONNECT permet à votre programme de gérer la connexion et la déconnexion à un serveur Modbus. Si DISCONNECT est égal à 0 et qu'il n'existe pas de liaison, MB_CLIENT tente d'établir une liaison à l'adresse et au numéro de port IP affectés. Si DISCONNECT est égal à 1 et qu'une liaison existe, une opération de déconnexion est tentée. Aucune autre opération n'est tentée lorsque cette entrée est activée.
CONNECT_ID	IN	UInt Le paramètre CONNECT_ID doit identifier de manière unique chaque liaison dans l'API. Chaque instance unique de l'instruction MB_CLIENT ou MB_SERVER doit contenir un paramètre CONNECT_ID unique.
IP_OCTET_1	IN	USInt Adresse IP du serveur Modbus TCP : Octet 1 8 bits de l'adresse IPv4 de 32 bits du serveur Modbus TCP auquel le client se connectera et avec lequel il communiquera à l'aide du protocole Modbus TCP.
IP_OCTET_2	IN	USInt Adresse IP du serveur Modbus TCP : Octet 2
IP_OCTET_3	IN	USInt Adresse IP du serveur Modbus TCP : Octet 3
IP_OCTET_4	IN	USInt Adresse IP du serveur Modbus TCP : Octet 4

Paramètre et type		Type de données	Description
IP_PORT	IN	UInt	Valeur par défaut = 502 : Numéro de port IP du serveur auquel le client tentera de se connecter dans le but de communiquer à l'aide du protocole TCP/IP.
MB_MODE	IN	USInt	Sélection du mode : Définit le type de demande (lecture, écriture ou diagnostic). Voir le tableau des fonctions Modbus ci-après pour plus de détails.
MB_DATA_ADDR	IN	UDInt	Adresse de début Modbus : Définit l'adresse de début des données auxquelles on accédera via MB_CLIENT. Voir les adresses valides dans le tableau des fonctions Modbus ci-après.
MB_DATA_LEN	IN	UInt	Longueur des données Modbus : Définit le nombre de bits ou mots auxquels accéder dans cette demande. Voir les longueurs valides dans le tableau des fonctions Modbus ci-après.
MB_DATA_PTR	IN_OUT	Variant	Pointeur désignant le registre de données Modbus : Le registre stocke temporairement les données allant vers un serveur Modbus ou venant d'un serveur Modbus. Le pointeur doit désigner un DB global standard ou une adresse en mémoire M.
DONE	OUT	Bool	Le bit DONE est TRUE pour un cycle lorsque la dernière demande s'est achevée sans erreur.
BUSY	OUT	Bool	<ul style="list-style-type: none"> <li>• 0 : Pas d'opération MB_CLIENT en cours</li> <li>• 1 : Opération MB_CLIENT en cours</li> </ul>
ERROR	OUT	Bool	Le bit ERROR est TRUE pour un cycle lorsque l'exécution de MB_CLIENT s'est achevée avec une erreur. La valeur de code d'erreur dans le paramètre STATUS ne vaut que pendant le cycle où ERROR est à TRUE.
STATUS	OUT	Word	Code d'erreur d'exécution

### Paramètre REQ

FALSE = Aucune demande de communication Modbus

TRUE = Demande de communication avec un serveur Modbus TCP

Si aucune instance de MB\_CLIENT n'est active et que le paramètre DISCONNECT est égal à 0, une nouvelle demande Modbus sera lancée si REQ est égal à 1. Si la liaison n'est pas déjà établie, une nouvelle liaison le sera.

Si la même instance de MB\_CLIENT est à nouveau exécutée avec DISCONNECT=0 et REQ=1, et ce avant l'achèvement de la demande en cours, il n'y aura pas d'émission Modbus suivante. Toutefois, dès que la demande en cours aura été achevée, une nouvelle demande pourra être traitée si MB\_CLIENT est exécuté avec l'entrée REQ à 1.

Lorsque la demande de communication MB\_CLIENT en cours est achevée, le bit DONE est TRUE pour un cycle. Le bit DONE peut être utilisé comme porte temporelle pour ordonner plusieurs demandes MB\_CLIENT.

**Remarque**

**Cohérence des données d'entrée pendant le traitement de MB\_CLIENT**

Lorsqu'un client Modbus a déclenché une opération Modbus, l'état de toutes les entrées est sauvegardé en interne, puis comparé à chaque appel successif. Cette comparaison sert à déterminer si cet appel particulier était à l'origine de la demande client active. Plusieurs appels MB\_CLIENT peuvent être effectués à l'aide d'un DB d'instance commun.

En conséquence, il est important que les entrées ne soient pas modifiées tant qu'une opération MB\_CLIENT est traitée activement. Si cette règle n'est pas respectée, un appel MB\_CLIENT ne peut pas déterminer qu'il est l'instance active.

**Sélection de la fonction de communication Modbus via les paramètres MB\_MODE et MB\_DATA\_ADDR**

MB\_DATA\_ADDR définit l'adresse Modbus de début des données auxquelles accéder. L'instruction MB\_CLIENT utilise une entrée MB\_MODE et non une entrée de code de fonction.

La combinaison des valeurs MB\_MODE et MB\_DATA\_ADDR détermine le code de fonction qui est utilisé dans le message Modbus réel. Le tableau suivant montre la correspondance entre le paramètre MB\_MODE, la fonction Modbus et la plage d'adresses Modbus.

Tableau 12- 51 Fonctions Modbus

MB_MODE	Fonction Modbus	Longueur des données	Opération et données	MB_DATA_ADDR
0	01	1 à 2000	Lecture de bits de sortie : 1 à 2000 bits par demande	1 à 9999
0	02	1 à 2000	Lecture de bits d'entrée : 1 à 2000 bits par demande	10001 à 19999
0	03	1 à 125	Lecture de registres de maintien : 1 à 125 mots par demande	40001 à 49999 ou 400001 à 465535
0	04	1 à 125	Lecture de mots d'entrée : 1 à 125 mots par demande	30001 à 39999
1	05	1	Écriture d'un bit de sortie : 1 bit par demande	1 à 9999
1	06	1	Écriture d'un registre de maintien : 1 mot par demande	40001 à 49999 ou 400001 à 465535
1	15	2 à 1968	Écriture de plusieurs bits de sortie : 2 à 1968 bits par demande	1 à 9999
1	16	2 à 123	Écriture de plusieurs registres de maintien : 2 à 123 mots par demande	40001 à 49999 ou 400001 à 465535
2	15	1 à 1968	Écriture d'un ou de plusieurs bits de sortie : 1 à 1968 bits par demande	1 à 9999

MB_MODE	Fonction Modbus	Longueur des données	Opération et données	MB_DATA_ADDR
2	16	1 à 123	Ecriture d'un ou de plusieurs registres de maintien : 1 à 123 mots par demande	40001 à 49999 ou 400001 à 465535
11	11	0	Lecture du mot d'état de communication serveur et du compteur d'événements. Le mot d'état indique si le serveur est occupé (0 : non occupé, 0xFFFF : occupé). Le compteur d'événements est incrémenté à chaque fois qu'un message s'achève avec succès.  Les deux paramètres MB_DATA_ADDR et MB_DATA_LEN de MB_CLIENT ne sont pas pris en compte pour cette fonction.	
80	08	1	Contrôle de l'état du serveur à l'aide du code de diagnostic de données 0x0000 (test de retour : le serveur renvoie la demande en écho)  1 mot par demande	
81	08	1	Réinitialisation du compteur d'événements du serveur à l'aide du code de diagnostic de données 0x000A  1 mot par demande	
3 à 10, 12 à 79, 82 à 255			Réservé	

**Remarque**

**MB\_DATA\_PTR définit une mémoire tampon pour stocker les données lues/écrites dans un serveur Modbus TCP**

La mémoire tampon de données peut se situer dans un DB global standard ou à une adresse de mémoire M.

Si la mémoire tampon est en mémoire M, utilisez le format de pointeur standard Any, à savoir P#"adresse bit" "type de données" "longueur" (par exemple, P#M1000.0 WORD 500).

### Définition d'une mémoire tampon de communication avec MB\_DATA\_PTR

- Fonctions de communication MB\_CLIENT :
  - Lecture et écriture de données d'un bit dans des adresses de serveur Modbus (00001 à 09999)
  - Lecture de données d'un bit dans des adresses de serveur Modbus (10001 à 19999)
  - Lecture de données mots de 16 bits dans des adresses de serveur Modbus (30001 à 39999) et (40001 à 49999)
  - Ecriture de données mots de 16 bits dans des adresses de serveur Modbus (40001 à 49999)
- Les données mots ou bits sont transférées vers/depuis la mémoire tampon de DB ou de mémoire M définie par MB\_DATA\_PTR.
- Si vous avez défini un DB comme mémoire tampon avec MB\_DATA\_PTR, vous devez affecter un type à tous les éléments de données du DB.
  - Le type de données Bool de 1 bit représente une adresse de bit Modbus.
  - Les types de données de 16 bits (mot unique), tels que WORD, UInt et Int, représentent une adresse de mot Modbus.
  - Les types de données de 32 bits (double mot), tels que DWORD, DInt et Real, représentent deux adresses de mot Modbus.
- MB\_DATA\_PTR permet d'indiquer des éléments de DB complexes, tels que :
  - tableaux standard
  - structures nommées où chaque élément est unique
  - structures complexes nommées où chaque élément a un nom unique et un type de données de 16 ou 32 bits
- Il n'est pas obligatoire que les zones de données MB\_DATA\_PTR soient dans le même bloc de données global (ou la même zone de mémoire M). Vous pouvez indiquer un bloc de données pour les lectures Modbus, un autre bloc de données pour les écritures Modbus ou un bloc de données pour chaque station MB\_CLIENT.

### Liaisons client multiples

Un client Modbus TCP peut accepter des liaisons simultanées jusqu'au nombre maximum de liaisons de communication ouverte autorisé par l'API. Le nombre total de liaisons pour un API, clients et serveurs Modbus TCP compris, ne doit pas dépasser le nombre maximum de liaisons de communication ouverte prises en charge (Page 452). Les liaisons Modbus TCP peuvent être partagées entre liaisons de type client et/ou serveur.

Les liaisons client individuelles doivent respecter les règles suivantes :

- Chaque liaison MB\_CLIENT doit utiliser un DB d'instance unique.
- Chaque liaison MB\_CLIENT doit indiquer une adresse IP de serveur unique.
- Chaque liaison MB\_CLIENT doit indiquer un ID de liaison unique.
- Des numéros de port IP uniques peuvent ou non être obligatoires selon la configuration du serveur.

L'ID de liaison doit être unique pour chaque liaison individuelle. Cela signifie qu'un ID de liaison unique ne doit être utilisé qu'avec le DB d'instance individuel correspondant. En résumé, le DB d'instance et l'ID de liaison forment une paire qui doit être unique pour chaque liaison.

Tableau 12- 52 Variables statiques accessibles à l'utilisateur du bloc de données d'instance MB\_CLIENT

Variable	Type de données	Valeur par défaut	Description
Blocked_Proc_Timeout	Real	3.0	Durée (en secondes) pendant laquelle attendre une instance de client Modbus bloquée avant d'annuler l'état ACTIF de cette instance. Cela peut, par exemple, se produire lorsqu'une demande client a été émise et que l'application arrête d'exécuter la fonction client avant l'achèvement complet de la demande. La limite maximale du S7-1200 est de 55 secondes.
MB_Unit_ID	Word	255	Identificateur d'unité Modbus : On accède à un serveur Modbus TCP à l'aide de son adresse IP. Le paramètre MB_UNIT_ID n'est donc pas utilisé pour l'adressage Modbus TCP.  Le paramètre MB_UNIT_ID correspond à l'adresse d'esclave dans le protocole Modbus RTU. Si un serveur Modbus TCP est utilisé comme passerelle vers un protocole Modbus RTU, on peut utiliser MB_UNIT_ID pour identifier l'esclave connecté sur le réseau série. On utiliserait MB_UNIT_ID pour transmettre la demande à l'adresse d'esclave Modbus RTU correcte.  Notez que certains appareils Modbus TCP peuvent nécessiter le paramètre MB_UNIT_ID afin d'être initialisés à l'intérieur d'une plage de valeurs restreinte.
RCV_TIMEOUT	Real	2.0	Durée en secondes pendant laquelle MB_CLIENT attend qu'un serveur réponde à une demande
Connected	Bool	0	Indique si la liaison au serveur affecté est établie ou coupée : 1=liaison établie, 0=liaison coupée

Tableau 12- 53 Erreurs de protocole MB\_CLIENT :

STATUS (W#16#)	Code réponse envoyé au client Modbus (B#16#)	Erreurs de protocole Modbus
8381	01	Code de fonction non pris en charge
8382	03	Erreur de longueur de données
8383	02	Erreur d'adresse de données ou accès en dehors des limites de la zone d'adresses de MB_HOLD_REG
8384	03	Erreur de valeur de données
8385	03	Valeur du code de diagnostic de données non pris en charge (code de fonction 08)

Tableau 12- 54 Codes d'erreur d'exécution de MB\_CLIENT <sup>1</sup>

STATUS (W#16#)	Erreurs de paramètres MB_CLIENT
7001	MB_CLIENT attend une réponse du serveur Modbus à une demande de connexion ou de déconnexion sur le port TCP affecté. Ce code n'est envoyé qu'à la première exécution d'une opération de connexion ou de déconnexion.
7002	MB_CLIENT attend une réponse du serveur Modbus à une demande de connexion ou de déconnexion pour le port TCP affecté. Ce code est envoyé pour les exécutions suivantes en attendant l'achèvement d'une opération de connexion ou de déconnexion.
7003	Une opération de déconnexion s'est achevée avec succès (valable pendant un seul cycle API).
80C8	Le serveur n'a pas répondu dans le temps imparti. MB_CLIENT doit recevoir une réponse avec l'ID de transaction initialement transmis dans le temps imparti ; sinon, cette erreur est renvoyée. Vérifiez la liaison au serveur Modbus. Cette erreur n'est signalée qu'après que le nombre configuré de nouvelles tentatives (le cas échéant) a été tenté.
8188	Valeur de mode invalide
8189	Valeur d'adresse de données invalide
818A	Valeur de longueur de données invalide
818B	Pointeur invalide sur la zone DATA_PTR. Il peut s'agir de la combinaison de MB_DATA_ADDRESS + MB_DATA_LEN.
818C	Pointeur désignant une zone DATA_PTR optimisée (ce doit être une zone DB standard ou une zone de mémoire M)
8200	Le port est occupé à traiter une demande Modbus existante.
8380	La trame Modbus reçue est mal formée ou trop peu d'octets ont été reçus.
8387	Le paramètre ID de liaison défini est différent de l'ID utilisé pour les demandes précédentes. Un seul ID de liaison peut être utilisé dans chaque DB d'instance MB_CLIENT. Ce code est également utilisé comme erreur interne si l'ID de protocole Modbus TCP reçu d'un serveur est différent de 0.
8388	Un serveur Modbus a renvoyé une quantité de données différente de celle demandée. Ce code s'applique aux fonctions Modbus 15 et 16 uniquement.

<sup>1</sup> En plus des erreurs MB\_CLIENT présentées ci-avant, des erreurs peuvent être renvoyées par les instructions de communication par blocs T sous-jacentes (TCON, TDISCON, TSEND et TRCV).

**Voir aussi**

TCON, TDISCON, TSEND et TRCV (Page 467)



### 12.5.2.2 MB\_SERVER (Modbus TCP)

Tableau 12- 55 Instruction MB\_SERVER

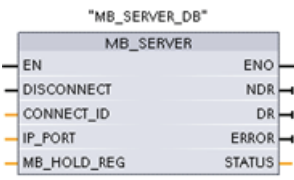
CONT/LOG	SCL	Description
	<pre>"MB_SERVER_DB" (     DISCONNECT:=_bool_in_,     CONNECT_ID:=_uint_in_,     IP_PORT:=_uint_in_,     NDR=&gt;_bool_out_,     DR=&gt;_bool_out_,     ERROR=&gt;_bool_out_,     STATUS=&gt;_word_out_,     MB_HOLD_REG:=_variant_inout_);</pre>	<p>MB_SERVER permet de communiquer en tant que serveur Modbus TCP par l'intermédiaire du connecteur PROFINET sur la CPU S7-1200. Aucun module matériel de communication supplémentaire n'est nécessaire.</p> <p>MB_SERVER permet d'accepter une demande de liaison à un client Modbus TCP, de recevoir une demande de fonction Modbus et d'envoyer un message de réponse.</p>

Tableau 12- 56 Types de données pour les paramètres

Paramètre et type	Type de données	Description
DISCONNECT   IN	Bool	MB_SERVER tente d'établir une liaison "passive" à un dispositif partenaire. Cela signifie que le serveur est passivement à l'écoute d'une demande de connexion TCP provenant de toute adresse IP demandeuse. Si DISCONNECT est égal à 0 et qu'il n'existe pas de liaison, une liaison passive peut être déclenchée. Si DISCONNECT est égal à 1 et qu'une liaison existe, une opération de déconnexion est lancée. Cela permet à votre programme de gérer le moment où une liaison est acceptée. Aucune autre opération n'est tentée lorsque cette entrée est activée.
CONNECT_ID   IN	UInt	Le paramètre CONNECT_ID identifie de manière unique chaque liaison dans l'API. Chaque instance unique de l'instruction MB_CLIENT ou MB_SERVER doit contenir un paramètre CONNECT_ID unique.
IP_PORT   IN	UInt	Valeur par défaut = 502 : Numéro qui identifie le port IP qui sera surveillé dans l'attente d'une demande de liaison provenant d'un client Modbus. Les numéros de port TCP suivants ne sont pas autorisés pour une liaison passive MB_SERVER : 20, 21, 25, 80, 102, 123, 5001, 34962, 34963 et 34964.
MB_HOLD_REG   IN_OUT	Variant	Pointeur désignant le registre de maintien Modbus pour MB_SERVER. Le registre de maintien peut être soit un DB global standard, soit une adresse en mémoire M. Cette zone de mémoire sert à stocker les valeurs auxquelles un client Modbus a le droit d'accéder à l'aide des fonctions Modbus de registre 3 (lecture), 6 (écriture) et 16 (écriture).
NDR   OUT	Bool	Nouvelles données prêtes (New Data Ready) : 0 = Pas de nouvelles données, 1 = Indique que de nouvelles données ont été écrites par un client Modbus
DR   OUT	Bool	Données lues : 0 = Pas de données lues, 1 = Indique que des données ont été lues par un client Modbus

Paramètre et type		Type de données	Description
ERROR	OUT	Bool	Le bit ERROR est TRUE pour un cycle lorsque l'exécution de MB_SERVER s'est achevée avec une erreur. La valeur de code d'erreur dans le paramètre STATUS ne vaut que pendant le cycle où ERROR est à TRUE.
STATUS	OUT	Word	Code d'erreur d'exécution

MB\_SERVER permet à des fonctions Modbus entrantes de codes 1, 2, 4, 5 et 15 de lire ou d'écrire des bits et des mots directement dans la mémoire image des entrées et la mémoire image des sorties de la CPU S7-1200. Pour les fonctions de transfert de données de codes 3, 6 et 16, le paramètre MB\_HOLD\_REG doit être défini avec un type de données supérieur à un octet. Le tableau suivant montre la correspondance entre les adresses Modbus et la mémoire image dans la CPU.

Tableau 12- 57 Correspondance entre adresses Modbus et mémoire image

Fonctions Modbus					S7-1200		
Codes	Fonction	Zone de données	Plage d'adresses			Zone de données	Adresse de la CPU
01	Lecture de bits	Sortie	1	à	8192	Mémoire image des sorties	Q0.0 à Q1023.7
02	Lecture de bits	Entrée	10001	à	18192	Mémoire image des entrées	I0.0 à I1023.7
04	Lecture de mots	Entrée	30001	à	30512	Mémoire image des entrées	IW0 à IW1022
05	Ecriture d'un bit	Sortie	1	à	8192	Mémoire image des sorties	Q0.0 à Q1023.7
15	Ecriture de bits	Sortie	1	à	8192	Mémoire image des sorties	Q0.0 à Q1023.7

Les fonctions de message Modbus entrantes de codes 3, 6 et 16 lisent ou écrivent des mots dans un registre de maintien Modbus qui peut être une plage d'adresses en mémoire M ou un bloc de données. Le type de registre de maintien est indiqué par le paramètre MB\_HOLD\_REG.

**Remarque**

**Paramétrage de MB\_HOLD\_REG**

Le registre de maintien Modbus peut être soit un DB global standard, soit une adresse en mémoire M.

Si le registre de maintien Modbus est en mémoire M, utilisez le format de pointeur standard Any, à savoir P#"adresse bit" "type de données" "longueur" (par exemple, P#M1000.0 WORD 500).

Le tableau suivant montre des exemples de correspondance entre adresses Modbus et registre de maintien pour les codes de fonction Modbus 03 (lecture de mots), 06 (écriture d'un mot) et 16 (écriture de mots). La limite supérieure effective des adresses de DB dépend des limites de mémoire de travail et de mémoire M maximum pour chaque modèle de CPU.

Tableau 12- 58 Exemples de correspondance entre adresse Modbus et adresse de mémoire CPU

Adresse Modbus	Exemples pour le paramètre MB_HOLD_REG		
	P#M100.0 Word 5	P#DB10.DBx0.0 Word 5	"Recette".ingredient
40001	MW100	DB10.DBW0	"Recette".ingredient[1]
40002	MW102	DB10.DBW2	"Recette".ingredient[2]
40003	MW104	DB10.DBW4	"Recette".ingredient[3]
40004	MW106	DB10.DBW6	"Recette".ingredient[4]
40005	MW108	DB10.DBW8	"Recette".ingredient[5]

### Liaisons serveur multiples

Plusieurs liaisons serveur peuvent être créées. Cela permet à un API unique d'établir des liaisons simultanées à plusieurs clients Modbus TCP.

Un serveur Modbus TCP peut accepter des liaisons simultanées jusqu'au nombre maximum de liaisons de communication ouverte autorisé par l'API. Le nombre total de liaisons pour un API, clients et serveurs Modbus TCP compris, ne doit pas dépasser le nombre maximum de liaisons de communication ouverte prises en charge (Page 452). Les liaisons Modbus TCP peuvent être partagées entre liaisons de type client et/ou serveur.

Les liaisons serveur individuelles doivent respecter les règles suivantes :

- Chaque liaison MB\_SERVER doit utiliser un DB d'instance unique.
- Chaque liaison MB\_SERVER doit être établie avec un numéro de port IP unique. Une seule liaison est acceptée par port.
- Chaque liaison MB\_SERVER doit utiliser un ID de liaison unique.
- L'instruction MB\_SERVER doit être appelée individuellement pour chaque liaison (avec son DB d'instance respectif).

L'ID de liaison doit être unique pour chaque liaison individuelle. Cela signifie qu'un ID de liaison unique ne doit être utilisé qu'avec le DB d'instance individuel correspondant. En résumé, le DB d'instance et l'ID de liaison forment une paire qui doit être unique pour chaque liaison.

Tableau 12- 59 Codes de fonction de diagnostic Modbus

Fonctions de diagnostic Modbus pour MB_SERVER		
Codes	Sous-fonction	Description
08	0x0000	Renvoi d'un test d'écho des données de requête : MB_SERVER renvoie en écho à un client Modbus un mot de données reçu.
08	0x000A	Effacement du compteur d'événements de communication : MB_SERVER efface le compteur d'événements de communication qui est utilisé pour la fonction Modbus 11.
11		Lecture du compteur d'événements de communication : MB_SERVER utilise un compteur d'événements de communication interne pour enregistrer le nombre de demandes de lecture et d'écriture Modbus envoyées au serveur Modbus qui ont abouti. Le compteur ne s'incrémente pas pour les fonctions 8 et 11. Il ne s'incrémente pas non plus pour toute demande entraînant une erreur de communication.  La fonction de diffusion générale n'est pas disponible pour Modbus TCP, car une seule liaison client-serveur existe à un moment donné.

### Variables MB\_SERVER

Ce tableau montre les variables statiques publiques sauvegardées dans le bloc de données d'instance pour MB\_SERVER pouvant être utilisées dans votre programme.

Tableau 12- 60 Variables statiques publiques pour MB\_SERVER

Variable	Type de données	Valeur par défaut	Description
HR_Start_Offset	Word	0	Définit l'adresse de début du registre de maintien Modbus.
Request_Count	Word	0	Nombre de toutes les demandes reçues par ce serveur
Server_Message_Count	Word	0	Nombre de demandes reçues pour ce serveur spécifique
Xmt_Rcv_Count	Word	0	Nombre d'émissions ou de réceptions pendant lesquelles une erreur s'est produite. Cette variable est également incrémentée si un message reçu est un message Modbus invalide.
Exception_Count	Word	0	Erreurs Modbus spécifiques nécessitant le renvoi d'une exception
Success_Count	Word	0	Nombre de demandes reçues pour ce serveur spécifique ne contenant pas d'erreurs de protocole
Connected	Bool	0	Indique si la liaison au client affecté est établie ou coupée : 1=liaison établie, 0=liaison coupée

Votre programme peut écrire des valeurs dans la variable HR\_Start\_Offset et commander le fonctionnement du serveur Modbus. Les autres variables peuvent être lues pour visualiser l'état de Modbus.

### HR\_Start\_Offset

Les adresses de registre de maintien Modbus commencent à 40001, ce qui correspond à l'adresse de début du registre de maintien dans la mémoire API. Vous pouvez toutefois configurer la variable "HR\_Start\_Offset" afin que le registre de maintien Modbus commence à une valeur autre que 40001.

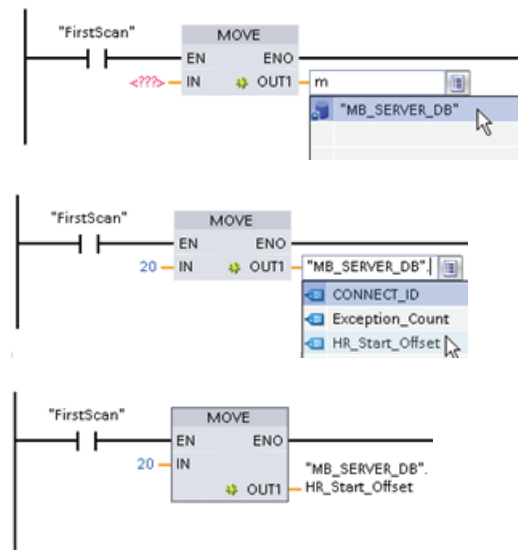
Par exemple, si le registre de maintien est configuré pour commencer à MW100 et s'il a une longueur de 100 mots. Un décalage de 20 spécifie une adresse de début de registre située à 40021 au lieu de 40001. Toute adresse inférieure à 40021 et supérieure à 40119 entraînera une erreur d'adressage.

Tableau 12- 61 Exemple d'adressage de registre de maintien Modbus

HR_Start_Offset	Adresse	Minimum	Maximum
0	Adresse Modbus (mot)	40001	40099
	Adresse S7-1200	MW100	MW298
20	Adresse Modbus (mot)	40021	40119
	Adresse S7-1200	MW100	MW298

HR\_Start\_Offset est une valeur de mot qui indique l'adresse de début du registre de maintien Modbus et est sauvegardée dans le bloc de données d'instance de MB\_SERVER. Vous pouvez définir la valeur de cette variable statique publique à l'aide de la liste déroulante d'aide pour les paramètres lorsque vous avez inséré MB\_SERVER dans votre programme.

Par exemple, une fois MB\_SERVER inséré dans un réseau CONT, vous pouvez aller à un réseau précédent et définir la valeur HR\_Start\_Offset. La valeur doit être affectée avant l'exécution de MB\_SERVER.



Entrée d'une variable pour le serveur Modbus à l'aide du nom de DB par défaut :

1. Placez le curseur dans le champ de paramètre et tapez la lettre m.
2. Sélectionnez "MB\_SERVER\_DB" dans la liste déroulante des noms de DB.
3. Sélectionnez "MB\_SERVER\_DB.HR\_Start\_Offset" dans la liste déroulante des variables de DB.

Tableau 12- 62 Codes d'erreur d'exécution de MB\_SERVER <sup>1</sup>

STATUS (W#16#)	Code réponse envoyé au serveur Modbus (B#16#)	Erreurs de protocole Modbus
7001		MB_SERVER attend qu'un client Modbus se connecte au port TCP affecté. Ce code est envoyé à la première exécution d'une opération de connexion ou de déconnexion.
7002		MB_SERVER attend qu'un client Modbus se connecte au port TCP affecté. Ce code est envoyé pour les exécutions suivantes en attendant l'achèvement d'une opération de connexion ou de déconnexion.
7003		Une opération de déconnexion s'est achevée avec succès (valable pendant un seul cycle API).
8187		Pointeur invalide sur MB_HOLD_REG : la zone est trop petite.
818C		Pointeur désignant une zone MB_HOLD_REG optimisée (ce doit être une zone DB standard ou une zone de mémoire M) ou le délai d'attente de processus bloqué dépasse la limite de 55 secondes. (spécifique au S7-1200)
8381	01	Code de fonction non pris en charge
8382	03	Erreur de longueur de données

STATUS (W#16#)	Code réponse envoyé au serveur Modbus (B#16#)	Erreurs de protocole Modbus
8383	02	Erreur d'adresse de données ou accès en dehors des limites de la zone d'adresses de MB_HOLD_REG
8384	03	Erreur de valeur de données
8385	03	Valeur du code de diagnostic de données non pris en charge (code de fonction 08)

<sup>1</sup> En plus des erreurs MB\_SERVER présentées ci-avant, des erreurs peuvent être renvoyées par les instructions de communication par blocs T sous-jacentes (TCON, TDISCON, TSEND et TRCV).

**Voir aussi**

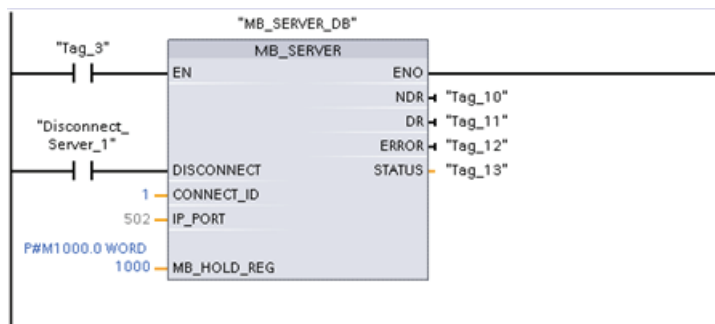
TCON, TDISCON, TSEND et TRCV (Page 467)

**12.5.2.3 Exemple pour MB\_SERVER : Liaisons TCP multiples**

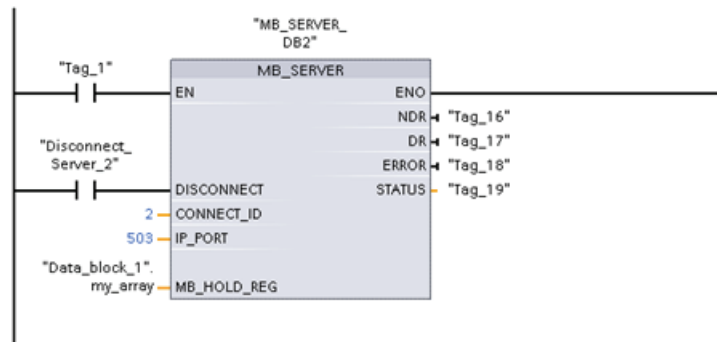
Vous pouvez avoir plusieurs liaisons serveur Modbus TCP. Pour ce faire, vous devez exécuter MB\_SERVER de manière indépendante pour chaque liaison. Chaque liaison doit utiliser un DB d'instance, un ID de liaison et un port IP qui lui soient propres. Le S7-1200 n'admet qu'une seule liaison par port IP.

Pour des performances optimales, il est préférable d'exécuter MB\_SERVER à chaque cycle pour chaque liaison.

**Réseau 1 : Liaison #1 avec port IP, ID de liaison et DB d'instance en propre**



**Réseau 2 : Liaison #2 avec port IP, ID de liaison et DB d'instance en propre**



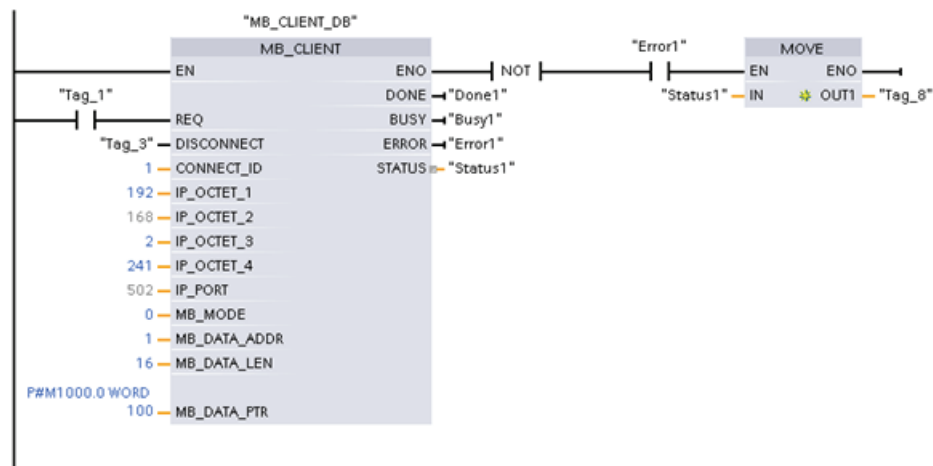
**12.5.2.4 Exemple 1 pour MB\_CLIENT : Plusieurs demandes avec une liaison TCP commune**

Vous pouvez envoyer plusieurs demandes client Modbus par la même liaison. Pour ce faire, utilisez les mêmes DB d'instance, ID de liaison et numéro de port.

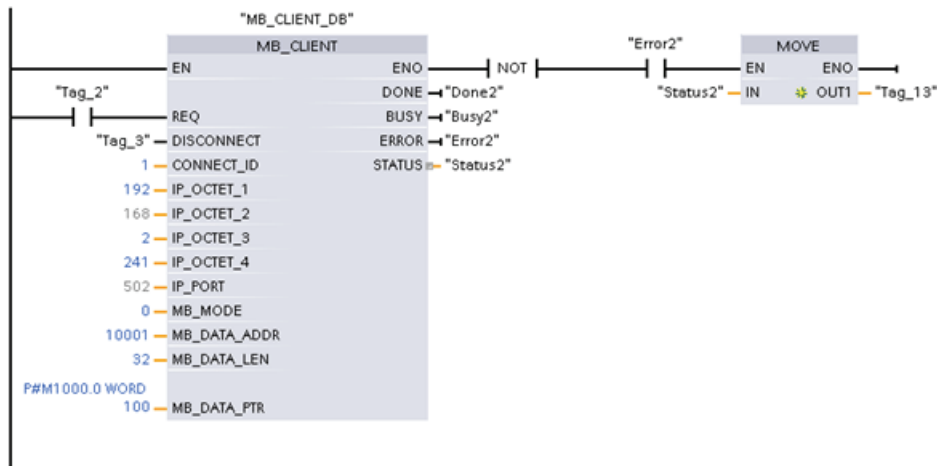
Un seul client peut être actif à un instant donné. Une fois qu'un client a achevé son exécution, le client suivant commence à s'exécuter. C'est votre programme qui gère l'ordre d'exécution.

L'exemple montre deux clients qui écrivent dans la même zone de mémoire. En outre, une erreur renvoyée est détectée, ce qui est facultatif.

**Réseau 1 : Fonction Modbus 1 : Lecture de 16 bits de la mémoire image des sorties**



Réseau 2 : Fonction Modbus 2 : Lecture de 32 bits de la mémoire image des entrées



12.5.2.5 Exemple 2 pour MB\_CLIENT : Plusieurs demandes avec des liaisons TCP différentes

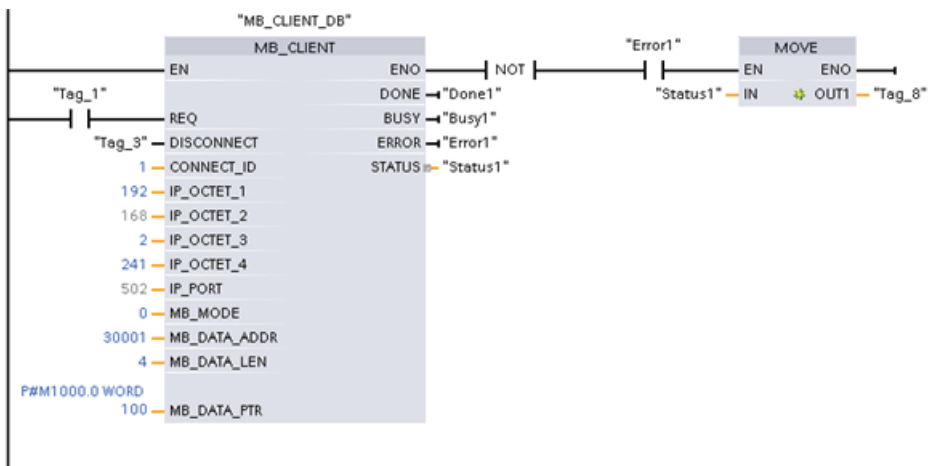
Vous pouvez envoyer plusieurs demandes client Modbus par le biais de liaisons différentes. Pour ce faire, vous devez utiliser des DB d'instance, des adresses IP et des ID de liaison différents.

Le numéro de port doit être différent si les liaisons sont établies vers le même serveur Modbus. Si les liaisons concernent des serveurs différents, aucune restriction ne s'applique au numéro de port.

L'exemple montre deux clients qui écrivent dans la même zone de mémoire. En outre, une erreur renvoyée est détectée, ce qui est facultatif.

Réseau 1 :

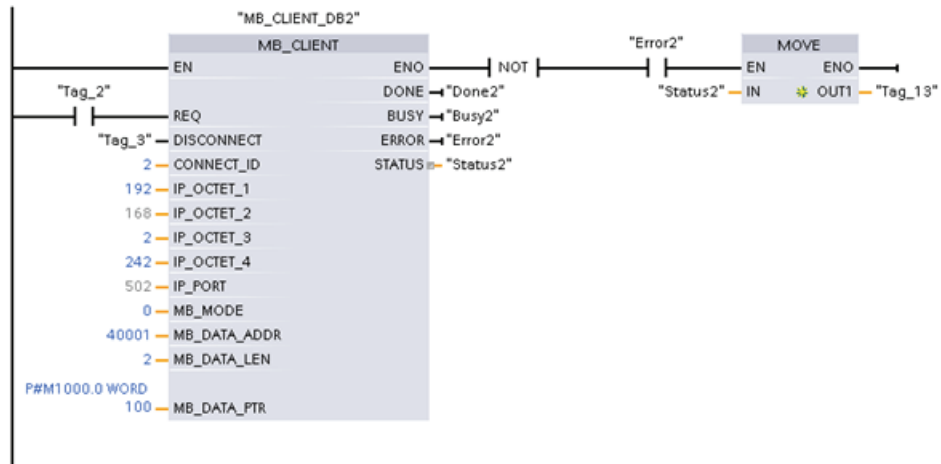
Fonction Modbus 4 : Lecture de mots d'entrée (dans la mémoire du S7-1200)





**Réseau 2 :**

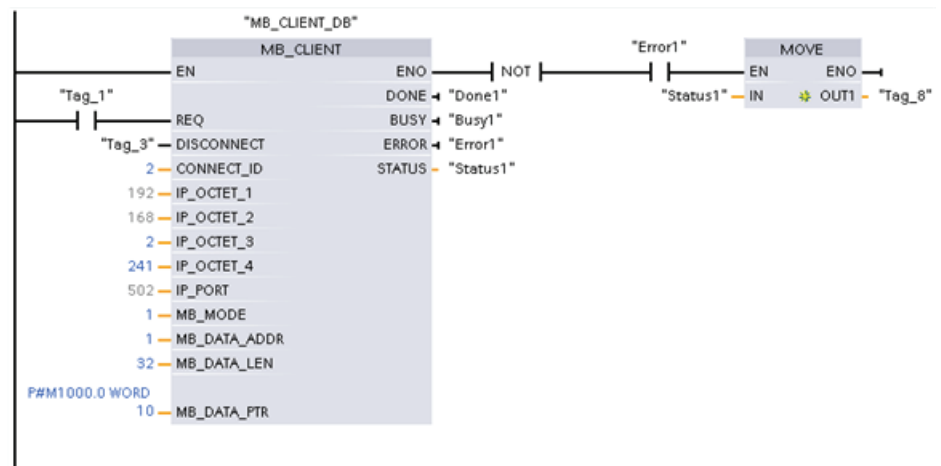
Fonction Modbus 3 : Lecture de mots d'un registre de maintien (dans la mémoire du S7-1200)



**12.5.2.6 Exemple 3 pour MB\_CLIENT : Demande d'écriture dans la mémoire image des sorties**

Cet exemple montre une demande client Modbus pour l'écriture dans la mémoire image des sorties du S7-1200.

**Réseau 1 :** Fonction Modbus 15 : Ecriture de bits dans la mémoire image des sorties du S7-1200

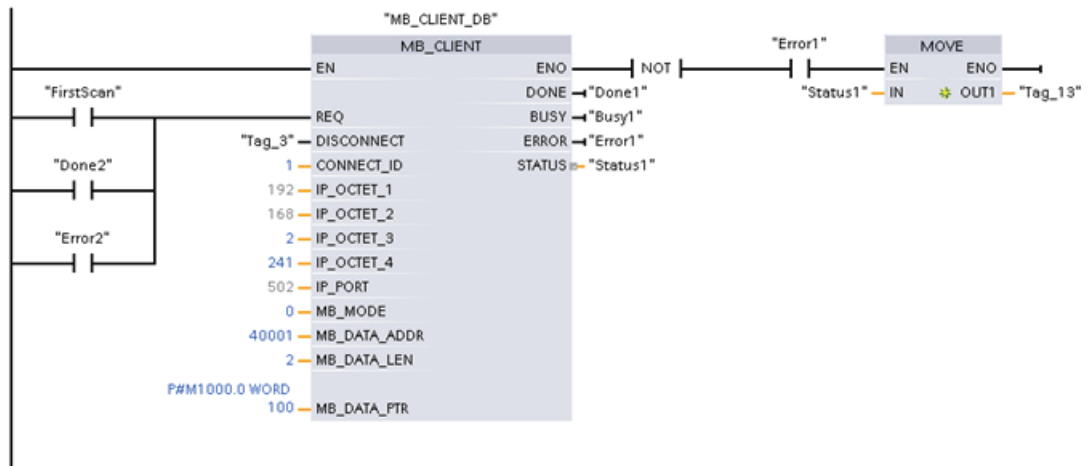


12.5.2.7 Exemple 4 pour MB\_CLIENT : Coordination de plusieurs demandes

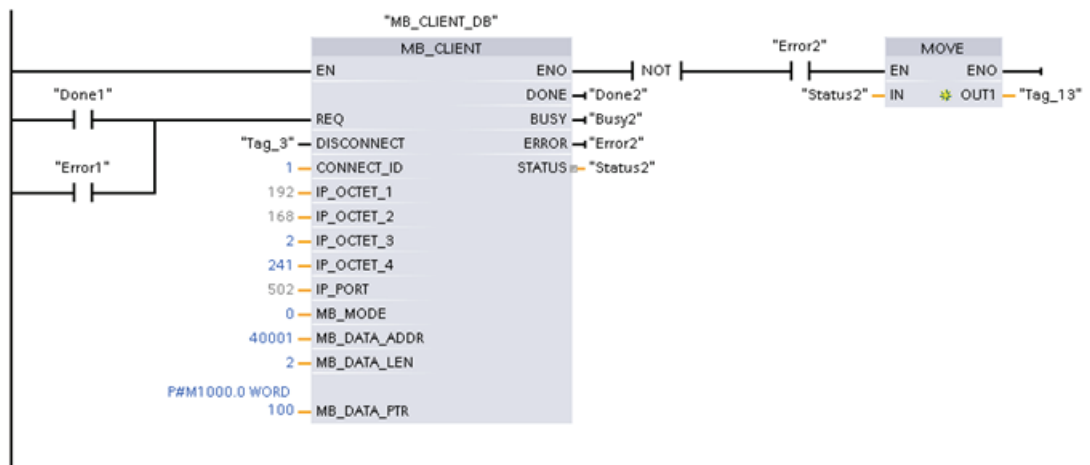
Vous devez veiller à ce que chaque demande Modbus TCP individuelle achève son exécution. Cette coordination doit être assurée par votre programme. L'exemple ci-dessous montre comment les sorties de la première et de la deuxième demande client peuvent être utilisées pour coordonner l'exécution.

L'exemple montre deux clients qui écrivent dans la même zone de mémoire. En outre, une erreur renvoyée est détectée, ce qui est facultatif.

Réseau 1 : Fonction Modbus 3 : Lecture de mots d'un registre de maintien



Réseau 2 : Fonction Modbus 3 : Lecture de mots d'un registre de maintien



### 12.5.3 Modbus RTU

Deux versions des instructions Modbus RTU sont disponibles dans STEP 7 :

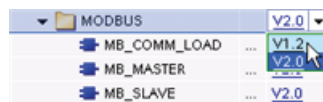
- La version 1 était initialement destinée à STEP 7 Basic V10.5.
- La version 2 concerne STEP 7 Basic/Professional V11. Dans la version 2, des paramètres REQ et DONE sont ajoutés à MB\_COMM\_LOAD. En outre, le paramètre MB\_ADDR pour MB\_MASTER et MB\_SLAVE autorise maintenant les valeurs Uint pour l'adressage étendu.

Pour des raisons de compatibilité et de facilité de migration, vous pouvez choisir la version de l'instruction à insérer dans votre programme utilisateur.

N'utilisez pas à la fois des instructions de version 1.x et 2.y dans le même programme CPU. Les instructions Modbus dans votre programme doivent avoir le même numéro de version principale (1.x, 2.y ou V.z). Toutefois, les instructions individuelles à l'intérieur d'un groupe de version principale peuvent avoir des versions secondaires différentes (1.x).



Dans la Task Card d'arborescence d'instructions, cliquez sur l'icône pour activer les en-têtes et colonnes de l'arborescence d'instructions.



Pour changer la version des instructions Modbus, sélectionnez la version désirée dans la liste déroulante. Vous pouvez sélectionner un groupe ou des instructions individuelles.

Lorsque vous utilisez l'arborescence d'instructions pour placer une instruction Modbus dans votre programme, une nouvelle instance de FB est créée dans l'arborescence du projet. Vous pouvez voir la nouvelle instance de FB dans l'arborescence de projet sous PLC\_x > Blocs de programme > Blocs système > Ressources du programme.

Pour vérifier la version d'une instruction Modbus dans un programme, vous devez vérifier les propriétés de l'arborescence de projet et non les propriétés de la boîte affichée dans l'éditeur de programmes. Sélectionnez une instance de FB Modbus dans l'arborescence de projet, cliquez avec le bouton droit de la souris, choisissez "Propriétés" et sélectionnez la page "Informations" pour voir le numéro de version de l'instruction Modbus.

### 12.5.3.1 MB\_COMM\_LOAD

Tableau 12- 63 Instruction MB\_COMM\_LOAD

CONT/LOG	SCL	Description
	<pre>"MB_COMM_LOAD_DB" (     REQ:=_bool_in_,     PORT:=_uint_in_,     BAUD:=_uint_in_,     PARITY:=_uint_in_,     FLOW_CTRL:=_uint_in_,     RTS_ON_DLY:=_uint_in_,     RTS_OFF_DLY:=_uint_in_,     RESP_TO:=_uint_in_,     DONE=&gt;_bool_out_,     ERROR=&gt;_bool_out_,     STATUS=&gt;_word_out_,     MB_DB:=_fbtref_inout_);</pre>	<p>L'instruction MB_COMM_LOAD configure un port point à point pour la communication via le protocole Modbus RTU. Options matérielles pour le port Modbus : Installez jusqu'à trois CM (RS485 ou RS232), ainsi qu'un CB (RS485). Un bloc de données d'instance est automatiquement affecté lorsque vous insérez l'instruction MB_COMM_LOAD dans votre programme.</p>

Tableau 12- 64 Types de données pour les paramètres

Paramètre et type		Type de données	Description
REQ	IN	Bool	Un front montant déclenche l'opération. (version 2.0 uniquement)
PORT	IN	Port	Une fois que vous avez installé et configuré un appareil de communication CM ou CB, l'identificateur de port apparaît dans la liste déroulante d'aide pour les paramètres disponible au niveau du connecteur PORT de la boîte. La valeur de port CM ou CB affectée est la propriété de configuration d'appareil "identificateur matériel". Le nom symbolique du port est défini dans l'onglet "Constantes système" de la table de variables API.
BAUD	IN	UDInt	Sélection du débit : 300, 600, 1200, 2400, 4800, 9600, 19200, 38400, 57600, 76800, 115200. Toutes les autres valeurs sont invalides.
PARITY	IN	UInt	Sélection de la parité : <ul style="list-style-type: none"> <li>• 0 = Pas de parité</li> <li>• 1 = Parité impaire</li> <li>• 2 = Parité paire</li> </ul>
FLOW_CTRL	IN	UInt	Sélection du contrôle de flux : <ul style="list-style-type: none"> <li>• 0 (valeur par défaut) = Pas de contrôle de flux</li> <li>• 1 = Contrôle de flux matériel avec RTS toujours activé (ne s'applique pas aux ports RS485)</li> <li>• 2 = Contrôle de flux matériel avec RTS commuté</li> </ul>

Paramètre et type		Type de données	Description
RTS_ON_DLY	IN	UInt	Sélection du retard RTS activé : <ul style="list-style-type: none"> <li>• 0 = (valeur par défaut) Pas de délai à partir de RTS actif jusqu'à ce que le premier caractère du message soit émis</li> <li>• 1 à 65535 = Délai en millisecondes à partir de RTS actif jusqu'à ce que le premier caractère du message soit émis (ne s'applique pas aux ports RS485). Les retards RTS sont appliqués indépendamment de la sélection FLOW_CTRL.</li> </ul>
RTS_OFF_DLY	IN	UInt	Sélection du retard RTS désactivé : <ul style="list-style-type: none"> <li>• 0 = (valeur par défaut) Pas de délai à partir du dernier caractère émis jusqu'à ce que RTS soit désactivé</li> <li>• 1 à 65535 = Délai en millisecondes à partir du dernier caractère émis jusqu'à ce que RTS soit désactivé (ne s'applique pas aux ports RS485). Les retards RTS sont appliqués indépendamment de la sélection FLOW_CTRL.</li> </ul>
RESP_TO	IN	UInt	Dépassement du délai d'attente de la réponse Temps en millisecondes accordé par MB_MASTER pour que l'esclave réponde. Si l'esclave ne répond pas pendant cette durée, MB_MASTER renouvellera la demande ou mettra fin à la demande avec une erreur lorsque le nombre de nouvelles tentatives indiqué aura été envoyé. 5 ms à 65535 ms (valeur par défaut = 1000 ms).
MB_DB	IN	Variant	Référence au bloc de données d'instance utilisé par les instructions MB_MASTER ou MB_SLAVE. Une fois que vous avez inséré MB_MASTER ou MB_SLAVE dans votre programme, l'identificateur de DB apparaît dans la liste déroulante d'aide pour les paramètres disponible au niveau du connecteur MB_DB de la boîte.
DONE	OUT	Bool	Le bit DONE est TRUE pour un cycle lorsque la dernière demande s'est achevée sans erreur. (version 2.0 uniquement)
ERROR	OUT	Bool	Le bit ERROR est TRUE pour un cycle lorsque la dernière demande s'est achevée avec une erreur. La valeur de code d'erreur dans le paramètre STATUS ne vaut que pendant le cycle où ERROR est à TRUE.
STATUS	OUT	Word	Code d'erreur d'exécution

MB\_COMM\_LOAD est exécuté pour configurer un port pour le protocole Modbus RTU. Une fois qu'un port est configuré pour le protocole Modbus RTU, il ne peut être utilisé que par les instructions MB\_MASTER ou MB\_SLAVE.

Il faut exécuter MB\_COMM\_LOAD une fois pour chaque port de communication utilisé pour la communication Modbus à configurer. Affectez un DB d'instance MB\_COMM\_LOAD unique pour chaque port que vous utilisez. Vous pouvez installer trois modules de communication (RS232 ou RS485) au maximum et un Communication Board (RS485) dans la CPU. Appelez MB\_COMM\_LOAD dans un OB de démarrage et exécutez-le une fois ou déclenchez l'appel via le memento système Premier cycle (Page 90) pour l'exécuter une fois. Ne réexécutez MB\_COMM\_LOAD que si des paramètres de communication tels que la vitesse de transmission ou la parité doivent changer.

Un bloc de données d'instance est affecté à MB\_MASTER ou MB\_SLAVE lorsque vous insérez ces instructions dans votre programme. Ce bloc de données d'instance est référencé lorsque vous définissez le paramètre MB\_DB pour l'instruction MB\_COMM\_LOAD.

**Variables du bloc de données MB\_COMM\_LOAD**

Le tableau suivant montre les variables statiques publiques sauvegardées dans le DB d'instance pour MB\_COMM\_LOAD pouvant être utilisées dans votre programme.

Tableau 12- 65 Variables statiques dans le DB d'instance

Variable	Type de données	Description
ICHAR_GAP	Word	Retard pour le délai inter-caractères. Ce paramètre indiqué en millisecondes sert à augmenter la durée escomptée entre caractères reçus. Le nombre correspondant de temps bit pour ce paramètre est ajouté à la valeur Modbus par défaut de 35 temps bit (3,5 temps caractère).
RETRIES	Word	Nombre de nouvelles tentatives que tentera le maître avant de renvoyer le code d'erreur Pas de réponse 0x80C8.

Tableau 12- 66 Codes d'erreur d'exécution de MB\_COMM\_LOAD <sup>1</sup>

STATUS (W#16#)	Description
0000	Pas d'erreur
8180	Valeur d'ID de port invalide (port/identificateur matériel erroné pour le module de communication)
8181	Valeur de vitesse de transmission invalide
8182	Valeur de parité invalide
8183	Valeur de contrôle de flux invalide
8184	Valeur invalide du délai de réponse (délai de réponse inférieur au minimum qui est de 5 ms)
8185	Le paramètre MB_DB n'est pas un bloc de données d'instance d'une instruction MB_MASTER ou MB_SLAVE.

<sup>1</sup> En plus des erreurs MB\_COMM\_LOAD présentées ci-avant, des erreurs peuvent être renvoyées par les instructions de communication point à point sous-jacentes.

**Voir aussi**

Instructions point à point (Page 603)

### 12.5.3.2 MB\_MASTER

Tableau 12- 67 Instruction MB\_MASTER

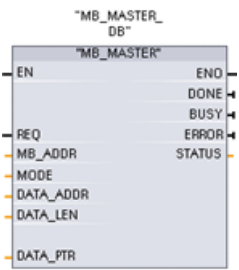
CONT/LOG	SCL	Description
	<pre>"MB_MASTER_DB" (   REQ:=_bool_in_,   MB_ADDR:=_uint_in_,   MODE:=_usint_in_,   DATA_ADDR:=_udint_in_,   DATA_LEN:=_uint_in_,   DONE=&gt;_bool_out_,   BUSY=&gt;_bool_out_,   ERROR=&gt;_bool_out_,   STATUS=&gt;_word_out_,   DATA_PTR:=_variant_inout_);</pre>	<p>L'instruction MB_MASTER permet la communication en tant que maître DB à l'aide d'un port configuré lors d'une exécution précédente de l'instruction MB_COMM_LOAD. Un bloc de données d'instance est automatiquement affecté lorsque vous insérez l'instruction MB_MASTER dans votre programme. Ce bloc de données d'instance MB_MASTER est utilisé lorsque vous définissez le paramètre MB_DB pour l'instruction MB_COMM_LOAD.</p>

Tableau 12- 68 Types de données pour les paramètres

Paramètre et type		Type de données	Description
REQ	IN	Bool	0 = Pas de demande 1 = Demande d'envoi de données à un esclave Modbus
MB_ADDR	IN	V1.0: USInt V2.0: UInt	Adresse de station Modbus RTU. Plage d'adressage standard (1 à 247) Plage d'adressage étendue (1 à 65535) La valeur 0 est réservée à la diffusion générale d'un message à tous les esclaves Modbus. Les codes de fonction Modbus 05, 06, 15 et 16 sont les seuls codes de fonction acceptés pour la diffusion générale.
MODE	IN	USInt	Sélection du mode : Indique le type de demande (lecture, écriture ou diagnostic). Voir le tableau des fonctions Modbus ci-après pour plus de détails.
DATA_ADDR	IN	UDInt	Adresse de début dans l'esclave. Indique l'adresse de début des données auxquelles accéder dans l'esclave Modbus. Voir les adresses valides dans le tableau des fonctions Modbus ci-après.
DATA_LEN	IN	UInt	Longueur des données. Indique le nombre de bits ou mots auxquels accéder dans cette demande. Voir les longueurs valides dans le tableau des fonctions Modbus ci-après.
DATA_PTR	IN	Variant	Pointeur de données. Pointe sur l'adresse de memento ou de DB (de type standard) pour les données en cours d'écriture ou de lecture.
DONE	OUT	Bool	Le bit DONE est TRUE pour un cycle lorsque la dernière demande s'est achevée sans erreur.
BUSY	OUT	Bool	<ul style="list-style-type: none"> <li>0 = Pas d'opération MB_MASTER en cours</li> <li>1 = Opération MB_MASTER en cours</li> </ul>
ERROR	OUT	Bool	Le bit ERROR est TRUE pour un cycle lorsque la dernière demande s'est achevée avec une erreur. La valeur de code d'erreur dans le paramètre STATUS ne vaut que pendant le cycle où ERROR est à TRUE.
STATUS	OUT	Word	Code d'erreur d'exécution

### Règles de communication pour le maître Modbus

- MB\_COMM\_LOAD doit être exécuté pour configurer un port avant qu'une instruction MB\_MASTER ne puisse communiquer avec ce port.
- Si un port doit être utilisé pour déclencher des demandes du maître Modbus, il ne doit pas être utilisé par MB\_SLAVE. Une ou plusieurs instances d'exécution de MB\_MASTER peuvent être utilisées avec ce port, mais toutes les exécutions de MB\_MASTER doivent utiliser le même DB d'instance MB\_MASTER pour ce port.
- Les instructions Modbus n'utilisent pas d'événements d'alarme de communication pour piloter le processus de communication. Votre programme doit interroger les conditions émission et réception achevées de l'instruction MB\_MASTER.
- Nous vous conseillons d'appeler toutes les exécutions de MB\_MASTER pour un port donné dans un OB de cycle de programme. Les instructions maître Modbus ne peuvent s'exécuter que dans un niveau d'exécution, cycle de programme ou alarme cyclique/temporisée. Elles ne peuvent pas s'exécuter dans les deux niveaux de priorité d'exécution. L'interruption d'une instruction maître Modbus par une autre instruction maître Modbus dans un niveau de priorité d'exécution supérieur entraîne un fonctionnement incorrect. Les instructions maître Modbus ne doivent pas s'exécuter dans les niveaux de priorité mise en route, diagnostic ou erreur de temps.
- Une fois qu'une instruction maître déclenche une transmission, cette instance doit s'exécuter en continu avec l'entrée EN activée jusqu'à ce qu'un état DONE égal à 1 ou qu'un état ERROR égal à 1 soit renvoyé. Une instance particulière de MB\_MASTER est considérée comme active jusqu'à ce que l'un de ces deux événements se produise. Tant que l'instance initiale est active, tout appel à une autre instance avec l'entrée REQ activée entraîne une erreur. Si l'exécution continue de l'instance initiale est interrompue, l'état de la demande reste actif pendant une durée indiquée par la variable statique "Blocked\_Proc\_Timeout". A l'expiration de cette durée, l'instruction maître suivante appelée avec l'entrée REQ activée devient l'instance active. Cela empêche une instance maître Modbus unique de monopoliser un port ou d'en verrouiller l'accès. Si l'instance active initiale n'est pas activée pendant la durée indiquée par la variable statique "Blocked\_Proc\_Timeout", l'exécution suivante par cette instance (avec REQ non activé) annulera l'état actif. Si REQ est activé, cette exécution déclenche une nouvelle demande maître comme si aucune autre instance n'était active.

### Paramètre REQ

0 = Pas de demande ; 1 = Demande d'envoi de données à un esclave Modbus

Vous pouvez commander cette entrée à l'aide d'un contact déclenché sur niveau ou sur front. Dès que cette entrée est activée, un automate fini est démarré pour garantir qu'aucune autre instruction MB\_MASTER utilisant le même DB d'instance n'a le droit d'émettre de demande, et ce jusqu'à ce que la demande en cours soit achevée. Tous les autres états d'entrée sont capturés et conservés en interne pendant la demande en cours jusqu'à ce que la réponse soit reçue ou qu'une erreur soit détectée.

Si la même instance de MB\_MASTER est à nouveau exécutée avec l'entrée REQ à 1 avant l'achèvement de la demande en cours, il n'y aura pas d'émissions suivantes. Toutefois, dès que la demande aura été achevée, une nouvelle demande sera émise si MB\_MASTER est à nouveau exécuté avec l'entrée REQ à 1.



### Sélection du type de fonction Modbus via les paramètres DATA\_ADDR et MODE

DATA\_ADDR (adresse Modbus de début dans l'esclave) : Indique l'adresse de début des données auxquelles accéder dans l'esclave Modbus.

L'instruction MB\_MASTER utilise une entrée MODE et non une entrée Code de fonction. La combinaison de MODE et de l'adresse Modbus déterminent le code de fonction qui est utilisé dans le message Modbus réel. Le tableau suivant montre la correspondance entre le paramètre MODE, le code de fonction Modbus et la plage d'adresses Modbus.

Tableau 12- 69 Fonctions Modbus

MODE	Fonction Modbus	Longueur des données	Opération et données	Adresse Modbus
0	01	1 à 2000 1 à 1992 <sup>1</sup>	Lecture de bits de sortie : 1 à 1992 (ou 2000) bits par demande	1 à 9999
0	02	1 à 2000 1 à 1992 <sup>1</sup>	Lecture de bits d'entrée : 1 à 1992 (ou 2000) bits par demande	10001 à 19999
0	03	1 à 125 1 à 124 <sup>1</sup>	Lecture de registres de maintien : 1 à 124 (ou 125) mots par demande	40001 à 49999 ou 400001 à 465535
0	04	1 à 125 1 à 124 <sup>1</sup>	Lecture de mots d'entrée : 1 à 124 (ou 125) mots par demande	30001 à 39999
1	05	1	Ecriture d'un bit de sortie : 1 bit par demande	1 à 9999
1	06	1	Ecriture d'un registre de maintien : 1 mot par demande	40001 à 49999 ou 400001 à 465535
1	15	2 à 1968 2 à 1960 <sup>1</sup>	Ecriture de plusieurs bits de sortie : 2 à 1960 (ou 1968) bits par demande	1 à 9999
1	16	2 à 123 2 à 122 <sup>1</sup>	Ecriture de plusieurs registres de maintien : 2 à 122 (ou 123) mots par demande	40001 à 49999 ou 400001 à 465535
2	15	1 à 1968 2 à 1960 <sup>1</sup>	Ecriture d'un ou de plusieurs bits de sortie : 1 à 1960 (ou 1968) bits par demande	1 à 9999
2	16	1 à 123 1 à 122 <sup>1</sup>	Ecriture d'un ou de plusieurs registres de maintien : 1 à 122 (ou 123) mots par demande	40001 à 49999 ou 400001 à 465535
11	11	0	Lecture du mot d'état de communication esclave et du compteur d'événements. Le mot d'état indique si l'esclave est occupé (0 : non occupé, 0xFFFF : occupé). Le compteur d'événements est incrémenté à chaque fois qu'un message s'achève avec succès.  Les deux opérandes DATA_ADDR et DATA_LEN de MB_MASTER ne sont pas pris en compte pour cette fonction.	
80	08	1	Contrôle de l'état de l'esclave à l'aide du code de diagnostic de données 0x0000 (test de retour : l'esclave renvoie la demande en écho)  1 mot par demande	

MODE	Fonction Modbus	Longueur des données	Opération et données	Adresse Modbus
81	08	1	Réinitialisation du compteur d'événements de l'esclave à l'aide du code de diagnostic de données 0x000A 1 mot par demande	
3 à 10, 12 à 79, 82 à 255			Réservé	

<sup>1</sup> En mode "Adressage étendu", les longueurs maximales de données sont réduites d'un octet ou d'un mot selon le type de données utilisé par la fonction.

### Paramètre DATA\_PTR

Le paramètre DATA\_PTR pointe sur l'adresse M ou DB à lire ou dans laquelle écrire. Si vous utilisez un bloc de données, vous devez créer un bloc de données global qui fournit de la mémoire pour les lectures et écritures dans les esclaves Modbus.

---

#### Remarque

**Le bloc de données DATA\_PTR doit autoriser l'adressage direct**

Le bloc de données doit autoriser à la fois l'adressage direct (absolu) et l'adressage symbolique. Vous devez donc sélectionner l'attribut d'accès "standard" lorsque vous créez le bloc de données.

---

### Structures de bloc de données pour le paramètre DATA\_PTR

- Ces types de données sont valides pour les **lectures de mot** des adresses Modbus 30001 à 39999, 40001 à 49999 et 400001 à 465536 et également pour les **écritures de mot** dans les adresses Modbus 40001 à 49999 et 400001 à 465536.
  - Tableau standard de types de données WORD, UINT ou INT
  - Structure WORD, UINT ou INT nommée dans laquelle chaque élément a un nom unique et un type de données de 16 bits
  - Structure complexe nommée dans laquelle chaque élément a un nom unique et un type de données de 16 ou 32 bits
- Pour les **lectures et écritures de bit** des adresses Modbus 00001 à 09999 et les lectures de bits des adresses 10001 à 19999
  - Tableau standard de types de données BOOL
  - Structure booléenne nommée de variables booléennes à nom unique

- Bien que cela ne soit pas obligatoire, il est recommandé que chaque instruction MB\_MASTER ait sa propre zone de mémoire distincte. La raison en est que le risque d'altération des données est plus important si plusieurs instructions MB\_MASTER lisent et écrivent dans la même zone de mémoire.
- Il n'est pas obligatoire que les zones de données DATA\_PTR soient dans le même bloc de données global. Vous pouvez créer un bloc de données à zones multiples pour les lectures Modbus, un bloc de données pour les écritures Modbus ou un bloc de données pour chaque station esclave.

### Variables du bloc de données maître Modbus

Le tableau suivant montre les variables statiques publiques sauvegardées dans le DB d'instance pour MB\_MASTER pouvant être utilisées dans votre programme.

Tableau 12- 70 Variables statiques dans le DB d'instance

Variable	Type de données	Valeur initiale	Description
Blocked_Proc_Timeout	Real	3.0	Durée (en secondes) pendant laquelle attendre une instance de maître Modbus bloquée avant d'annuler l'état ACTIF de cette instance. Cela peut, par exemple, se produire lorsqu'une demande maître a été émise et que le programme arrête d'appeler la fonction maître avant l'achèvement complet de la demande. La valeur de temps doit être supérieure à 0 et inférieure à 55 secondes ou une erreur se produit. La valeur par défaut est 0,5 seconde.
Extended_Addressing	Bool	False	Configure l'adressage d'esclave à un ou deux octets. La valeur par défaut est 0. (0=adresse à un octet, 1=adresse à deux octets)

Votre programme peut écrire des valeurs dans les variables Blocked\_Proc\_Timeout et Extended\_Addressing pour commander le fonctionnement du maître Modbus. Vous trouverez dans la description de HR\_Start\_Offset et Extended\_Addressing au paragraphe MB\_SLAVE un exemple d'utilisation de ces variables dans l'éditeur de programmes et des détails sur l'adressage étendu Modbus (Page 693).

### Codes d'erreur

Tableau 12- 71 Codes d'erreur d'exécution de MB\_MASTER (erreurs de communication et de configuration) <sup>1</sup>

STATUS (W#16#)	Description
0000	Pas d'erreur
80C8	Dépassement de délai chez l'esclave. Vérifiez la vitesse de transmission, la parité et le câblage de l'esclave.
80D1	Le récepteur a émis une demande de contrôle de flux pour suspendre l'émission active et n'a jamais réactivé la transmission durant le temps d'attente indiqué. Cette erreur est également générée pendant le contrôle de flux matériel lorsque le récepteur n'affirme pas Prêt à émettre (CTS) pendant le temps d'attente indiqué.
80D2	La demande d'émission a été annulée, car aucun signal DSR (modem prêt) n'a été reçu de l'équipement de transmission de données (DCE).
80E0	Le message a été interrompu car la mémoire tampon de réception est pleine.

STATUS (W#16#)	Description
80E1	Le message a été interrompu en raison d'une erreur de parité.
80E2	Le message a été interrompu en raison d'une erreur de trame.
80E3	Le message a été interrompu en raison d'une erreur de débordement.
80E4	Le message a été interrompu, car la longueur indiquée dépasse la taille de mémoire tampon totale.
8180	Valeur d'ID de port invalide ou erreur avec l'instruction MB_COMM_LOAD
8186	Adresse de station Modbus invalide
8188	Mode invalide indiqué pour une demande de diffusion générale
8189	Valeur d'adresse de données invalide
818A	Valeur de longueur de données invalide
818B	Pointeur invalide vers la source/destination de données locale : taille incorrecte
818C	Pointeur invalide pour DATA_PTR ou Blocked_Proc_Timeout invalide : La zone de données doit être un DB (permettant l'adressage symbolique et absolu) ou se situer en mémoire M.
8200	Le port est occupé par le traitement d'une demande d'émission.

Tableau 12- 72 Codes d'erreur d'exécution de MB\_MASTER (erreurs de protocole Modbus) <sup>1</sup>

STATUS (W#16#)	Code de réponse de l'esclave	Erreurs de protocole Modbus
8380	-	Erreur CRC
8381	01	Code de fonction non pris en charge
8382	03	Erreur de longueur de données
8383	02	Erreur d'adresse de données ou adresse en dehors de la plage valide de la zone DATA_PTR
8384	Supérieur à 03	Erreur de valeur de données
8385	03	Valeur du code de diagnostic de données non pris en charge (code de fonction 08)
8386	-	Le code de fonction dans la réponse ne correspond pas au code dans la demande.
8387	-	Ce n'est pas le bon esclave qui a répondu.
8388	-	La réponse de l'esclave à une demande d'écriture est incorrecte. La demande d'écriture renvoyée par l'esclave ne correspond pas à ce que le maître a effectivement envoyé.

<sup>1</sup> En plus des erreurs MB\_MASTER présentées ci-avant, des erreurs peuvent être renvoyées par les instructions de communication point à point sous-jacentes.

**Voir aussi**

Instructions point à point (Page 603)

### 12.5.3.3 MB\_SLAVE

Tableau 12- 73 Instruction MB\_SLAVE

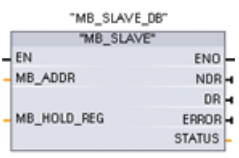
CONT/LOG	SCL	Description
	<pre>"MB_SLAVE_DB" (   MB_ADDR:=_uint_in_,   NDR=&gt;_bool_out_,   DR=&gt;_bool_out_,   ERROR=&gt;_bool_out_,   STATUS=&gt;_word_out_,   MB_HOLD_REG:=_variant_inout_);</pre>	<p>L'instruction MB_SLAVE permet à votre programme de communiquer en tant qu'esclave Modbus par le biais d'un port point à point sur le CM (RS485 ou RS232) et le CB (RS485). Lorsqu'un maître RTU Modbus éloigné émet une demande, votre programme utilisateur répond par l'exécution de MB_SLAVE. STEP 7 crée automatiquement un DB d'instance lorsque vous insérez l'instruction. Utilisez le nom MB_SLAVE_DB lorsque vous indiquez le paramètre MB_DB pour l'instruction MB_COMM_LOAD.</p>

Tableau 12- 74 Types de données pour les paramètres

Paramètre et type	Type de données	Description
MB_ADDR	IN	V1.0: USInt V2.0: UInt Adresse de station de l'esclave Modbus Plage d'adressage standard (1 à 247) Plage d'adressage étendue (0 à 65535)
MB_HOLD_REG	IN	Variant Pointeur désignant le DB de registre de maintien Modbus : Le registre de maintien Modbus peut être un memento ou un bloc de données.
NDR	OUT	Bool Nouvelles données prêtes (New Data Ready) : <ul style="list-style-type: none"> <li>0 = Pas de nouvelles données</li> <li>1 = Indique que des nouvelles données ont été écrites par le maître Modbus.</li> </ul>
DR	OUT	Bool Données lues : <ul style="list-style-type: none"> <li>0 = Pas de données lues</li> <li>1 = Indique que des données ont été lues par le maître Modbus.</li> </ul>
ERROR	OUT	Bool Le bit ERROR est TRUE pour un cycle lorsque la dernière demande s'est achevée avec une erreur. Si l'exécution s'achève avec une erreur, la valeur de code d'erreur dans le paramètre STATUS ne vaut que pendant le cycle où ERROR est à TRUE.
STATUS	OUT	Word Code d'erreur d'exécution

Les fonctions de communication Modbus de codes 1, 2, 4, 5 et 15 peuvent lire et écrire des bits et des mots directement dans la mémoire image des entrées et la mémoire image des sorties de la CPU. Pour ces fonctions, le paramètre MB\_HOLD\_REG doit être défini avec un type de données supérieur à un octet. Le tableau suivant montre en exemple la correspondance entre les adresses Modbus et la mémoire image dans la CPU.

Tableau 12- 75 Correspondance entre adresses Modbus et mémoire image

Fonctions Modbus					S7-1200		
Codes	Fonction	Zone de données	Plage d'adresses			Zone de données	Adresse de la CPU
01	Lecture de bits	Sortie	1	à	8192	Mémoire image des sorties	Q0.0 à Q1023.7
02	Lecture de bits	Entrée	10001	à	18192	Mémoire image des entrées	I0.0 à I1023.7
04	Lecture de mots	Entrée	30001	à	30512	Mémoire image des entrées	IW0 à IW1022
05	Ecriture d'un bit	Sortie	1	à	8192	Mémoire image des sorties	Q0.0 à Q1023.7
15	Ecriture de bits	Sortie	1	à	8192	Mémoire image des sorties	Q0.0 à Q1023.7

Les fonctions de communication Modbus de codes 3, 6 et 16 utilisent un registre de maintien Modbus qui peut être une zone d'adresse en mémoire M ou un bloc de données. Le type de registre de maintien est indiqué par le paramètre MB\_HOLD\_REG dans l'instruction MB\_SLAVE.

**Remarque**

**Bloc de données MB\_HOLD\_REG**

Un bloc de données faisant office de registre de maintien Modbus doit autoriser à la fois l'adressage direct (absolu) et l'adressage symbolique. Vous devez donc sélectionner l'attribut d'accès "standard" lorsque vous créez le bloc de données.

Le tableau suivant montre des exemples de correspondance entre adresses Modbus et registre de maintien pour les codes de fonction Modbus 03 (lecture de mots), 06 (écriture d'un mot) et 16 (écriture de mots). La limite supérieure effective des adresses de DB dépend des limites de mémoire de travail et de mémoire M maximum pour chaque modèle de CPU.

Tableau 12- 76 Correspondance entre adresses Modbus et mémoire CPU

Adresse du maître Modbus	Exemples pour le paramètre MB_HOLD_REG				
	MW100	DB10.DBW0	MW120	DB10.DBW50	"Recette".ingredient
40001	MW100	DB10.DBW0	MW120	DB10.DBW50	"Recette".ingredient[1]
40002	MW102	DB10.DBW2	MW122	DB10.DBW52	"Recette".ingredient[2]
40003	MW104	DB10.DBW4	MW124	DB10.DBW54	"Recette".ingredient[3]
40004	MW106	DB10.DBW6	MW126	DB10.DBW56	"Recette".ingredient[4]
40005	MW108	DB10.DBW8	MW128	DB10.DBW58	"Recette".ingredient[5]

Tableau 12- 77 Fonctions de diagnostic

Fonctions de diagnostic Modbus MB_SLAVE du S7-1200		
Codes	Sous-fonction	Description
08	0000H	Renvoi d'un test d'écho des données de requête : MB_SLAVE renvoie en écho à un maître Modbus un mot de données reçu.
08	000AH	Effacement du compteur d'événements de communication : MB_SLAVE effacera le compteur d'événements de communication qui est utilisé pour la fonction Modbus 11.
11		Lecture du compteur d'événements de communication : MB_SLAVE utilise un compteur d'événements de communication interne pour enregistrer le nombre de demandes de lecture et d'écriture Modbus envoyées à l'esclave Modbus qui ont abouti. Le compteur ne s'incrémente pas pour les fonctions 8 et 11 ni pour les demandes à diffusion générale. Il ne s'incrémente pas non plus pour toute demande entraînant une erreur de communication (erreurs de parité ou de CRC, par exemple).

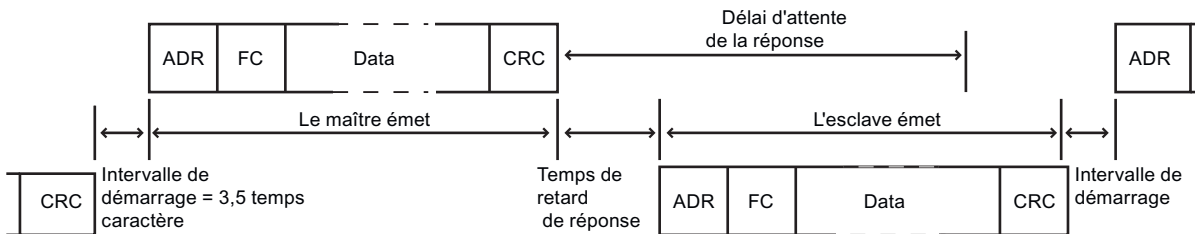
L'instruction MB\_SLAVE accepte les demandes d'écriture à diffusion générale provenant de n'importe quel maître Modbus à condition que la requête demande l'accès à des adresses valides. MB\_SLAVE génère un code d'erreur 0x8188 pour les codes de fonction non pris en charge en diffusion générale.

### Règles de communication pour l'esclave Modbus

- MB\_COMM\_LOAD doit être exécuté pour configurer un port avant qu'une instruction MB\_SLAVE ne puisse communiquer par le biais de ce port.
- Si un port doit répondre en tant qu'esclave à un maître Modbus, ne programmez pas ce port avec l'instruction MB\_MASTER.
- Une seule instance de MB\_SLAVE peut être utilisée avec un port donné, faute de quoi le comportement pourrait devenir imprévisible.
- Les instructions Modbus n'utilisent pas d'événements d'alarme de communication pour piloter le processus de communication. Votre programme doit piloter le processus de communication en interrogeant les conditions émission et réception achevées de l'instruction MB\_SLAVE.
- L'instruction MB\_SLAVE doit s'exécuter périodiquement à une fréquence permettant de répondre opportunément à des demandes entrantes provenant d'un maître Modbus. Il est recommandé d'exécuter MB\_SLAVE à chaque cycle dans un OB de cycle de programme. Il est possible d'exécuter MB\_SLAVE dans un OB d'alarme cyclique, mais ce n'est pas recommandé en raison du risque que des retards excessifs dans le programme d'alarme bloquent l'exécution d'autres programmes d'alarme.

### Synchronisation des signaux Modbus

MB\_SLAVE doit être exécuté périodiquement pour recevoir chaque demande provenant du maître Modbus et pour y répondre de manière appropriée. La fréquence d'exécution de MB\_SLAVE dépend du délai d'attente de la réponse dans le maître Modbus. Ceci est illustré dans le schéma suivant.



Le délai d'attente de la réponse RESP\_TO est la durée pendant laquelle un maître Modbus attend le début d'une réponse d'un esclave Modbus. Ce temps n'est pas défini par le protocole Modbus mais constitue un paramètre de chaque maître Modbus. La fréquence d'exécution (intervalle de temps entre une exécution et la suivante) de MB\_SLAVE doit se baser sur les paramètres spécifiques de votre maître Modbus. Vous devriez exécuter MB\_SLAVE au minimum deux fois pendant le délai d'attente de la réponse du maître Modbus.

### Variables pour l'esclave Modbus

Ce tableau montre les variables statiques publiques sauvegardées dans le bloc de données d'instance pour MB\_SLAVE pouvant être utilisées dans votre programme.

Tableau 12- 78 Variables pour l'esclave Modbus

Variable	Type de données	Description
HR_Start_Offset	Word	Indique l'adresse de début du registre de maintien Modbus (valeur par défaut = 0).
Extended_Addresssing	Bool	Configure l'adressage d'esclave à un ou deux octets. (0=adresse à un octet, 1=adresse à deux octets ; valeur par défaut = 0)
Request_Count	Word	Nombre de toutes les demandes reçues par cet esclave
Slave_Message_Count	Word	Nombre de demandes reçues pour cet esclave spécifique
Bad_CRC_Count	Word	Nombre de demandes reçues comportant une erreur CRC
Broadcast_Count	Word	Nombre de demandes à diffusion générale reçues
Exception_Count	Word	Erreurs Modbus spécifiques nécessitant le renvoi d'une exception
Success_Count	Word	Nombre de demandes reçues pour cet esclave spécifique ne contenant pas d'erreurs de protocole

Votre programme peut écrire des valeurs dans les variables HR\_Start\_Offset et Extended\_Addresssing pour commander le fonctionnement de l'esclave Modbus. Les autres variables peuvent être lues pour visualiser l'état de Modbus.



## HR\_Start\_Offset

Les adresses de registre de maintien Modbus commencent à 40001 ou 400001, ce qui correspond à l'adresse de début du registre de maintien dans la mémoire de la CPU. Vous pouvez toutefois configurer la variable "HR\_Start\_Offset" afin que le registre de maintien Modbus commence à une valeur autre que 40001 ou 400001.

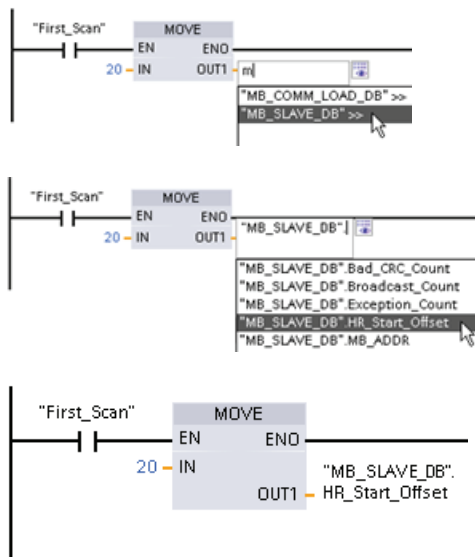
Par exemple, si le registre de maintien est configuré pour commencer à MW100 et s'il a une longueur de 100 mots. Un décalage de 20 spécifie une adresse de début de registre située à 40021 au lieu de 40001. Toute adresse inférieure à 40021 et supérieure à 400119 entraînera une erreur d'adressage.

Tableau 12- 79 Exemple d'adressage de registre de maintien Modbus

HR_Start_Offset	Adresse	Minimum	Maximum
0	Adresse Modbus (mot)	40001	40099
	Adresse S7-1200	MW100	MW298
20	Adresse Modbus (mot)	40021	40119
	Adresse S7-1200	MW100	MW298

HR\_Start\_Offset est une valeur de mot qui indique l'adresse de début du registre de maintien Modbus et est sauvegardée dans le bloc de données d'instance de MB\_SLAVE. Vous pouvez définir la valeur de cette variable statique publique à l'aide de la liste déroulante d'aide pour les paramètres lorsque vous avez inséré MB\_SLAVE dans votre programme.

Par exemple, une fois MB\_SLAVE inséré dans un réseau CONT, vous pouvez aller à un réseau précédent et définir la valeur HR\_Start\_Offset. La valeur doit être affectée avant l'exécution de MB\_SLAVE.



Entrée d'une variable pour l'esclave Modbus à l'aide du nom de DB par défaut :

1. Placez le curseur dans le champ de paramètre et tapez la lettre m.
2. Sélectionnez "MB\_SLAVE\_DB" dans la liste déroulante.
3. Placez le curseur à droite du nom de DB (après le guillemet) et entrez un point.
4. Sélectionnez "MB\_SLAVE\_DB.HR\_Start\_Offset" dans la liste déroulante.

**Extended Addressing**

Vous accédez à la variable `Extended Addressing` de la même manière qu'à la variable `HR_Start_Offset` décrite ci-avant, si ce n'est que `Extended Addressing` est une valeur booléenne. Une valeur booléenne doit être écrite par une bobine de sortie et non par une boîte MOVE.

L'adressage d'esclave Modbus peut être configuré à un octet (la norme pour Modbus) ou à deux octets. On utilise l'adressage étendu pour accéder à plus de 247 unités dans un réseau unique. La sélection de l'adressage étendu vous permet d'accéder à 64000 adresses au maximum. Une trame pour la fonction Modbus 1 est présentée ci-dessous comme exemple.

Tableau 12- 80 Adresse d'esclave à un octet (octet 0)

Fonction 1	Octet 0	Octet 1	Octet 2	Octet 3	Octet 4	Octet 5	
Demande	Adr. esclave	Code F	Adresse de début		Longueur des bobines		
Réponse valide	Adr. esclave	Code F	Longueur	Données de bobine			
Réponse erronée	Adr. esclave	0x81	Code E				

Tableau 12- 81 Adresse d'esclave à deux octets (octet 0 et octet 1)

	Octet 0	Octet 1	Octet 2	Octet 3	Octet 4	Octet 5	Octet 6
Demande	Adresse d'esclave		Code F	Adresse de début		Longueur des bobines	
Réponse valide	Adresse d'esclave		Code F	Longueur	Données de bobine		
Réponse erronée	Adresse d'esclave		0x81	Code E			

**Codes d'erreur**

Tableau 12- 82 Codes d'erreur d'exécution de `MB_SLAVE` (erreurs de communication et de configuration) <sup>1</sup>

STATUS (W#16#)	Description
80D1	Le récepteur a émis une demande de contrôle de flux pour suspendre l'émission active et n'a jamais réactivé la transmission durant le temps d'attente indiqué. Cette erreur est également générée pendant le contrôle de flux matériel lorsque le récepteur n'affirme pas Prêt à émettre (CTS) pendant le temps d'attente indiqué.
80D2	La demande d'émission a été annulée, car aucun signal DSR (modem prêt) n'a été reçu de l'équipement de transmission de données (DCE).
80E0	Le message a été interrompu car la mémoire tampon de réception est pleine.
80E1	Le message a été interrompu en raison d'une erreur de parité.
80E2	Le message a été interrompu en raison d'une erreur de trame.
80E3	Le message a été interrompu en raison d'une erreur de débordement.
80E4	Le message a été interrompu, car la longueur indiquée dépasse la taille de mémoire tampon totale.

STATUS (W#16#)	Description
8180	Valeur d'ID de port invalide ou erreur avec l'instruction MB_COMM_LOAD
8186	Adresse de station Modbus invalide
8187	Pointeur invalide vers le DB MB_HOLD_REG : La zone est trop petite.
818C	Pointeur MB_HOLD_REG invalide sur mémoire M ou DB (le DB doit autoriser à la fois l'adressage symbolique et l'adressage absolu)

Tableau 12- 83 Codes d'erreur d'exécution de MB\_SLAVE (erreurs de protocole Modbus) <sup>1</sup>

STATUS (W#16#)	Code de réponse de l'esclave	Erreurs de protocole Modbus
8380	Pas de réponse	Erreur CRC
8381	01	Code de fonction non pris en charge ou non pris en charge dans des messages à diffusion générale
8382	03	Erreur de longueur de données
8383	02	Erreur d'adresse de données ou adresse en dehors de la plage valide de la zone DATA_PTR
8384	03	Erreur de valeur de données
8385	03	Valeur du code de diagnostic de données non pris en charge (code de fonction 08)

<sup>1</sup> En plus des erreurs MB\_SLAVE présentées ci-avant, des erreurs peuvent être renvoyées par les instructions de communication point à point sous-jacentes.

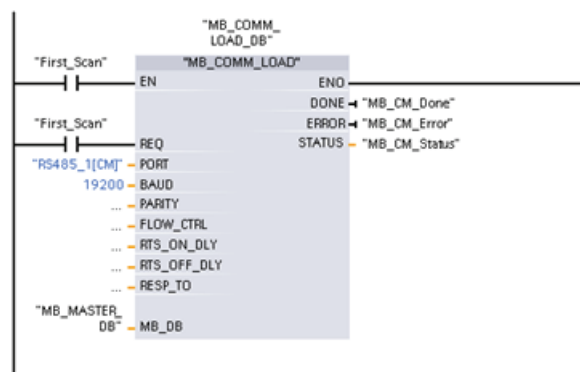
## Voir aussi

Instructions point à point (Page 603)

### 12.5.3.4 Exemple de programme maître Modbus RTU

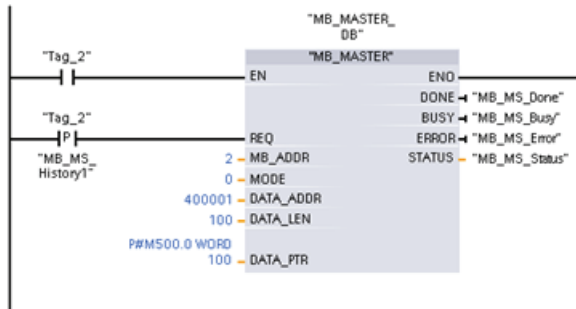
MB\_COMM\_LOAD est initialisé au démarrage à l'aide du memento Premier cycle. Cette manière d'exécuter MB\_COMM\_LOAD ne doit être utilisée que lorsque la configuration du port série n'est pas modifiée pendant l'exécution.

**Réseau 1** Initialiser les paramètres de module RS485 une seule fois pendant le premier cycle.

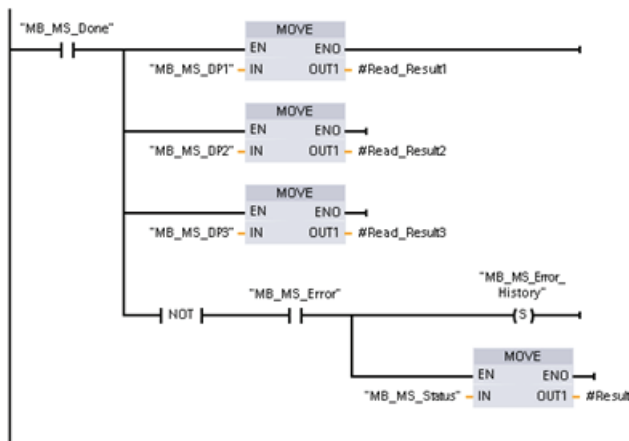


Une instruction MB\_MASTER est utilisée dans l'OB de cycle de programme pour communiquer avec un esclave unique. Il est possible d'utiliser d'autres instructions MB\_MASTER dans l'OB de cycle de programme pour communiquer avec d'autres esclaves ou de réutiliser un FB MB\_MASTER pour communiquer avec des esclaves supplémentaires.

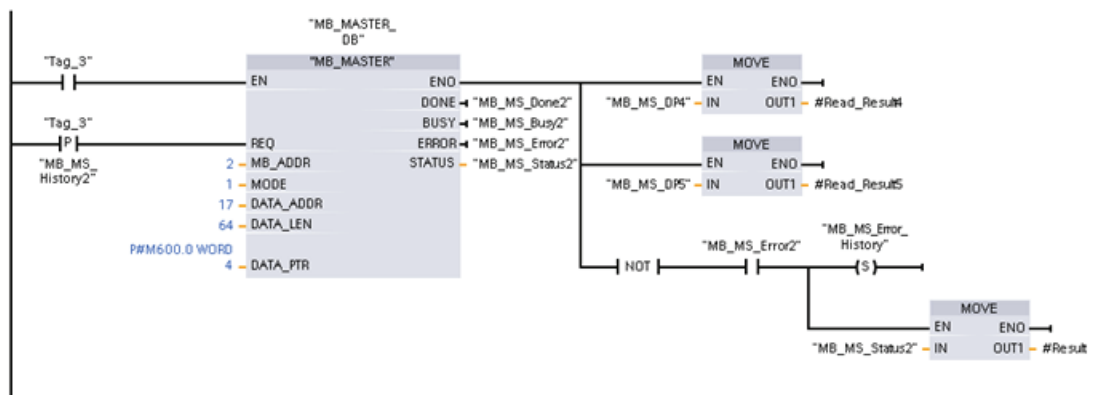
**Réseau 2** Lire 100 mots dans le registre de maintien de l'esclave.



**Réseau 3** Il s'agit d'un réseau optionnel qui montre uniquement les valeurs des 3 premiers mots une fois l'opération de lecture effectuée.



**Réseau 4** Ecrire 64 bits dans la mémoire image de sortie en commençant à l'adresse d'esclave Q2.0.

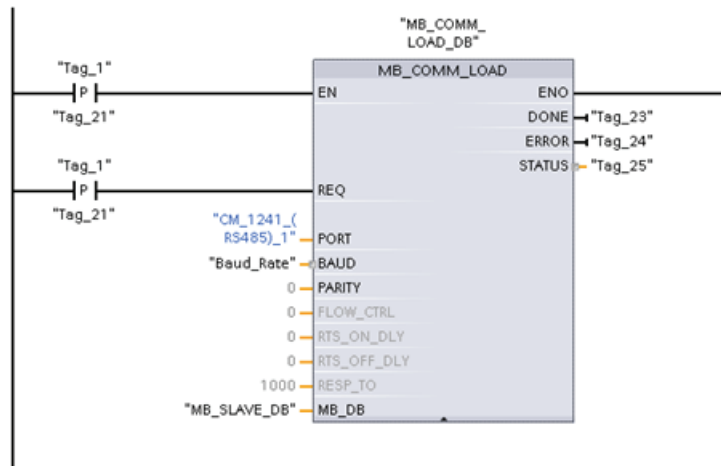


### 12.5.3.5 Exemple de programme esclave Modbus RTU

MB\_COMM\_LOAD présenté ci-dessous est initialisé à chaque fois que "Tag\_1" est activé.

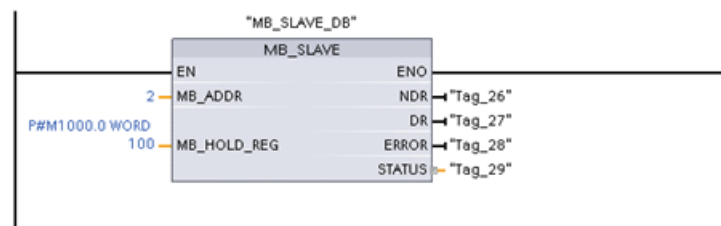
Cette manière d'exécuter MB\_COMM\_LOAD ne doit être utilisée que lorsque la configuration du port série est modifiée à l'exécution, en raison d'une configuration IHM.

**Réseau 1** Initialiser les paramètres de module RS485 à chaque fois qu'ils sont modifiés par un appareil IHM.



L'instruction MB\_SLAVE montrée ci-dessous est insérée dans un OB cyclique qui est exécuté toutes les 10 ms. Cela n'assure certes pas la réponse la plus rapide possible de l'esclave, mais cela garantit de bonnes performances à 9600 bauds pour des messages courts (20 octets au plus dans la demande).

**Réseau 2** Vérifier la présence de demandes maître Modbus à chaque cycle. Le registre de maintien Modbus est configuré pour 100 mots en commençant à MW1000.



## 12.6 Telecontrol et TeleService avec le CP 1242-7

### 12.6.1 Connexion à un réseau GSM

#### Communication WAN basée IP via GPRS

Le S7-1200 peut être connecté au réseau GSM à l'aide du processeur de communication CP 1242-7. Le CP 1242-7 permet à des stations distantes de communiquer entre elles ou avec un poste de commande central via un réseau étendu (WAN).

La communication transversale entre stations n'est possible que via le réseau GSM. Lors de la communication d'une station distante avec un poste de commande central, ce dernier doit être doté d'un PC connecté à Internet.

Le CP 1242-7 prend en charge les services suivants pour la communication via le réseau GSM :

- GPRS (General Packet Radio Service)

Ce service de transmission de données par paquets "GPRS" est exécuté sur le réseau GSM.

- SMS (Short Message Service)

Le CP 1242-7 peut recevoir et émettre des messages sous forme de SMS. Le partenaire de communication peut être un téléphone mobile ou un S7-1200.

Le CP 1242-7 est conçu pour une utilisation en tous points du globe ; il prend en charge les bandes passantes suivantes :

- 850 MHz
- 900 MHz
- 1 800 MHz
- 1 900 MHz

## Conditions préalables

L'équipement des stations ou du poste central dépend de l'application.

- La communication avec ou via un poste central présuppose que ce poste soit équipé d'un PC connecté à Internet.
- Pour qu'une station S7-1200 distante avec CP 1242-7 puisse communiquer via le réseau GSM, il faut l'avoir dotée de l'équipement de station voulu et avoir rempli les conditions suivantes :
  - avoir contracté un abonnement avec un opérateur de réseau GSM  
Il faut disposer d'un abonnement qui permette d'utiliser le service GPRS si cela est souhaité.  
Il faut que l'opérateur de réseau GSM attribue au CP une adresse IP fixe permettant aux stations de communiquer directement entre elles via le réseau GSM. Dans ce cas la communication entre stations ne transite pas par le poste central.
  - posséder la carte SIM de l'abonnement  
La carte SIM doit être installée sur le CP 1242-7.
  - recevoir le réseau GSM à l'emplacement de la station

## 12.6.2 Application du CP 1242-7

Les applications suivantes sont réalisables avec le CP 1242-7 :

### Applications Telecontrol

- Envoi de messages via SMS

Par le CP 1242-7, la CPU d'une station S7-1200 distante reçoit des SMS du réseau GSM ou envoie des messages via SMS à un téléphone mobile configuré ou à un S7-1200.

- Communication avec un poste de commande central

Les stations S7-1200 distantes communiquent via le réseau GSM et Internet avec un serveur Telecontrol du poste de commande central. La transmission de données via GPRS présuppose que l'application "TELECONTROL SERVER BASIC" soit installée sur le serveur Telecontrol du poste de commande central. Grâce à la fonction de serveur OPC intégrée, le serveur Telecontrol communique avec un système de contrôle-commande central de niveau supérieur.

- Communication entre stations S7-1200 via un réseau GSM

La communication entre stations distantes dotées d'un CP 1242-7 peut se dérouler de deux manières :

- Communication transversale via un poste de commande central

Dans cette configuration, des liaisons, sauvegardées en permanence, sont établies entre les stations S7-1200 communiquant entre elles et le serveur Telecontrol du poste de commande central. La communication entre les stations transite toujours par le serveur Telecontrol. Le CP 1242-7 fonctionne en mode "Telecontrol".

- communication directe entre stations

Pour la communication directe entre les stations, sans détour par un poste de commande central, on utilise des cartes SIM à adresse IP fixe qui permettent d'adresser les stations directement. Les services de communication et fonctions de sécurité (VPN p. ex.) disponibles dépendent de l'offre de l'opérateur de réseau. Le CP 1242-7 fonctionne en mode "GPRS direct".

### TeleService via GPRS

Une liaison TeleService peut être établie via le réseau GSM et Internet entre une station d'ingénierie avec STEP 7 et une station S7-1200 distante dotée d'un CP 1242-7. La liaison issue de la station d'ingénierie passe par un serveur Telecontrol et une passerelle TeleService qui, en tant que mandataires, retransmettent les télégrammes et se chargent de l'autorisation. Ces PC se servent des fonctions de l'application "TELECONTROL SERVER BASIC".

La liaison TeleService est utilisable pour les besoins suivants :

- chargement de données de configuration et de programme d'un projet STEP 7 sur la station
- lecture de données de diagnostic sur la station



### 12.6.3 Autres propriétés du CP

#### Autres services et fonctions du CP 1242-7

- Synchronisation d'horloge du CP via Internet  
L'heure du CP peut être réglée comme suit :
  - En mode "Telecontrol" l'heure est transmise par le serveur Telecontrol. Le CP y ajuste son heure.
  - En mode "GPRS direct", le CP peut envoyer une requête d'horodatage via SNTP.  
Pour synchroniser l'horloge de la CPU, vous pouvez lire l'heure momentanée du CP à l'aide d'un bloc de données.
- Mise en mémoire tampon des télégrammes à émettre en cas de problèmes de liaison
- Disponibilité accrue grâce à la possibilité de connexion à un serveur Telecontrol de secours
- Volume de données optimisé (liaison temporaire)  
Le CP peut être configuré sous STEP 7 avec une liaison temporaire au serveur Telecontrol au lieu d'une liaison permanente au serveur Telecontrol. Le cas échéant, la liaison au serveur Telecontrol est uniquement établie à la demande.
- Journalisation du volume de données  
Les volumes de données transmises sont enregistrés pour être exploités ultérieurement.

#### Configuration et échange de module

La configuration du module nécessite l'emploi de l'outil d'ingénierie suivant :

STEP 7 version V11.0 SP1 ou suivantes

Pour STEP 7 V11.0 SP1, vous avez en outre besoin du Support Package "CP 1242-7" (HSP0003001).

Pour transmettre des données de process via GPRS, utilisez les instructions de communication Telecontrol dans le programme utilisateur de la station.

Les données de configuration du CP 1242-7 sont enregistrées sur la CPU locale. Ceci permet d'échanger facilement le CP en cas de panne.

Vous pouvez embrocher jusqu'à trois modules de type CP 1242-7 sur le S7-1200. Ceci permet de réaliser par exemple des voies de communication redondantes.

#### Connexions électriques

- Alimentation du CP 1242-7  
Le CP est doté par ailleurs d'une connexion pour une alimentation DC 24 V externe.
- Interface radio pour le réseau GSM  
La communication GSM nécessite une antenne externe. Celle-ci se connecte sur le connecteur femelle SMA du CP.

### Informations complémentaires

Le manuel du CP 1242-7 contient des informations détaillées. Vous les trouverez sur Internet sur le site Siemens Industrial Automation Customer Support sous le numéro d'article suivant :

42330276 (<http://support.automation.siemens.com/WW/view/fr/42330276>)

## 12.6.4 Accessoires

### Antenne GSM/GPRS ANT794-4MR

Les antennes ci-après, destinées à être montées à l'intérieur ou à l'extérieur, sont disponibles pour une utilisation dans les réseaux GSM/GPRS :

- antenne quadribande ANT794-4MR



Figure 12-1 antenne GSM/GPRS ANT794-4MR

Désign. abrégée	N° de référence	Explication
ANT794-4MR	6NH9 860-1AA00	Antenne quadribande (900, 1800/1900 MHz, UMTS) ; pour intérieur et extérieur, insensible aux variations climatiques ; câble de connexion de 5 m relié à l'antenne ; connecteur SMA ; avec équerre de montage, vis, chevilles

- antenne plate ANT794-3M

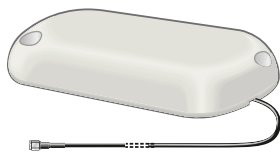


Figure 12-2 Antenne plate ANT794-3M

Désign. abrégée	N° de référence	Explication
ANT794-3M	6NH9 870-1AA00	Antenne plate (900, 1800/1900 MHz) ; pour intérieur et extérieur, insensible aux variations climatiques ; câble de connexion de 1,2 m relié à l'antenne ; connecteur SMA ; avec coussinet adhésif, fixation par vis possible

Les antennes doivent être commandées séparément.

### Informations complémentaires

Le manuel de l'appareil contient des informations détaillées. Vous les trouverez sur Internet sur le site du Siemens Industrial Automation Customer Support sous le numéro d'article :

23119005 (<http://support.automation.siemens.com/WW/view/fr/23119005>)

## 12.6.5 Exemples de configuration pour Telecontrol

Vous trouverez ci-après quelques exemples de configuration de stations dotées d'un CP 1242-7.

### Envoi de SMS

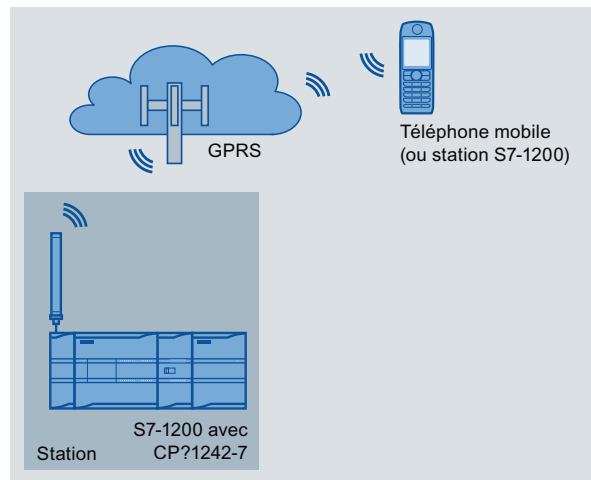


Figure 12-3 Envoi de SMS d'une station S7-1200

Un automate SIMATIC S7-1200 doté d'un CP 1242-7 peut envoyer des messages via SMS à un téléphone mobile ou à une station S7-1200 configurée.

### Telecontrol par un poste de commande central

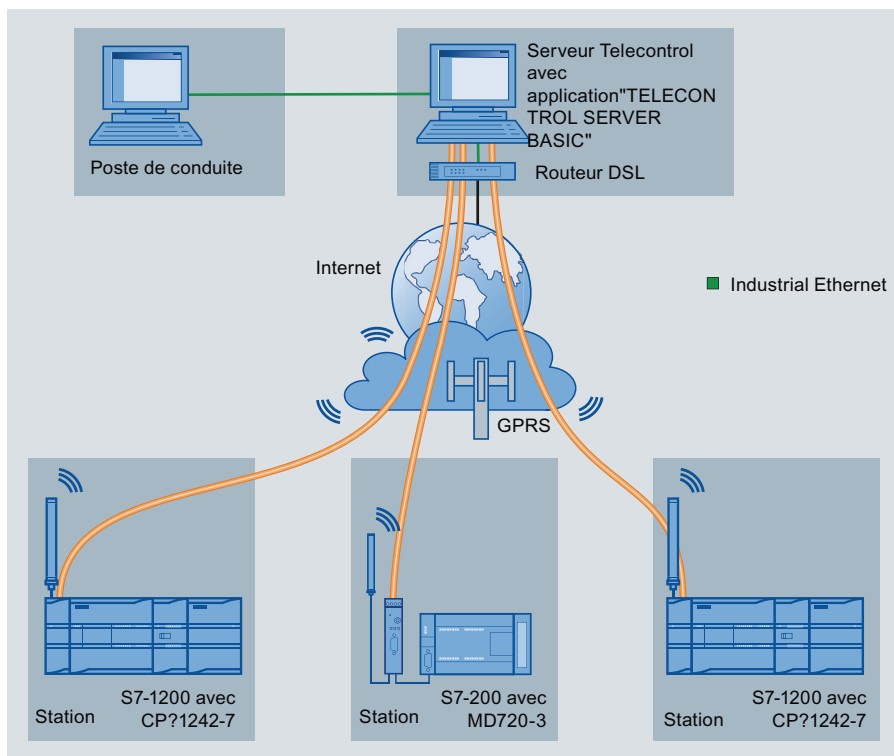


Figure 12-4 Communication de stations S7-1200 avec un poste de commande central

Dans le cas d'applications Telecontrol, les stations SIMATIC S7-1200 avec CP 1242-7 communiquent via le réseau GSM et Internet avec un poste de commande central. L'application "TELECONTROL SERVER BASIC" (TCSB) est installée sur le serveur Telecontrol du poste de commande central. Il en découle les possibilités d'utilisation suivantes :

- Communication Telecontrol entre station et poste de commande central

Dans ce cas, les données de terrain sont transmises au serveur Telecontrol du poste de commande central via le réseau GSM et Internet. Le serveur Telecontrol sert à superviser les stations distantes.

- Communication entre une station et un poste de commande central avec client OPC

Comme dans le premier cas, les stations communiquent avec le serveur Telecontrol. Le serveur Telecontrol échange des données avec le client OPC du poste de commande central grâce au serveur OPC intégré.

Le client OPC et le serveur Telecontrol peuvent également se trouver sur un même ordinateur, par exemple si TCSB est installé sur un ordinateur de contrôle-commande avec WinCC.

- Communication transversale entre stations via un poste de commande central

La communication transversale est possible avec les stations S7 également équipées d'un CP 1242-7.

Dans le cas de la communication transversale entre stations, le serveur Telecontrol retransmet les télégrammes de la station émettrice vers la station réceptrice.

## Communication directe entre stations

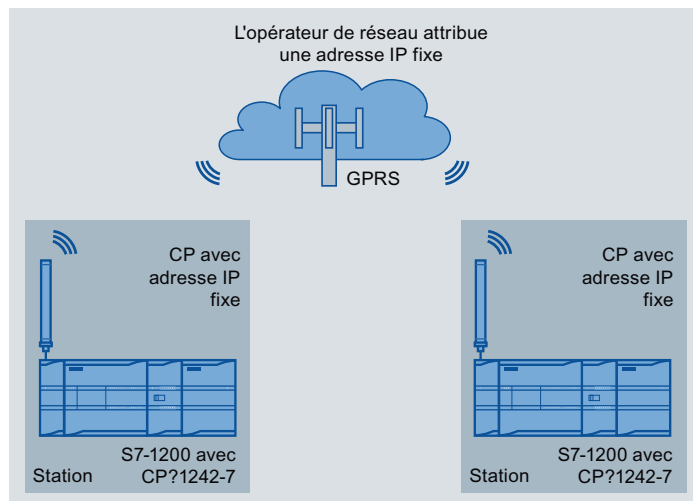


Figure 12-5 Communication directe entre deux stations S7-1200

Dans cette configuration, deux stations SIMATIC S7-1200 communiquent directement l'une avec l'autre, à l'aide du CP 1242-7, via le réseau GSM. Chaque CP 1242-7 possède une adresse IP fixe. Le service adéquat de l'opérateur de réseau GSM doit le permettre.

### TeleService via GPRS

En cas de TeleService via GPRS, une station d'ingénierie sur laquelle STEP 7 est installé, communique via le réseau GSM et Internet avec le CP 1242-7 du S7-1200.

Le pare-feu rejetant généralement les requêtes de liaison issues de l'extérieur, il faut prévoir une station mandataire entre la station distante et la station d'ingénierie. Cette station mandataire peut être un serveur Telecontrol ou, s'il n'existe pas de serveur Telecontrol dans la configuration, une passerelle TeleService.

### TeleService avec serveur Telecontrol

La liaison passe par le serveur Telecontrol.

- La station d'ingénierie et le serveur Telecontrol sont interconnectés via intranet (réseau local) ou Internet.
- Le serveur Telecontrol et la station distante sont interconnectés via Internet et le réseau GSM.

La station d'ingénierie et le serveur Telecontrol peuvent également se trouver sur le même ordinateur, en d'autres termes STEP 7 et TCBS sont installés sur le même ordinateur.

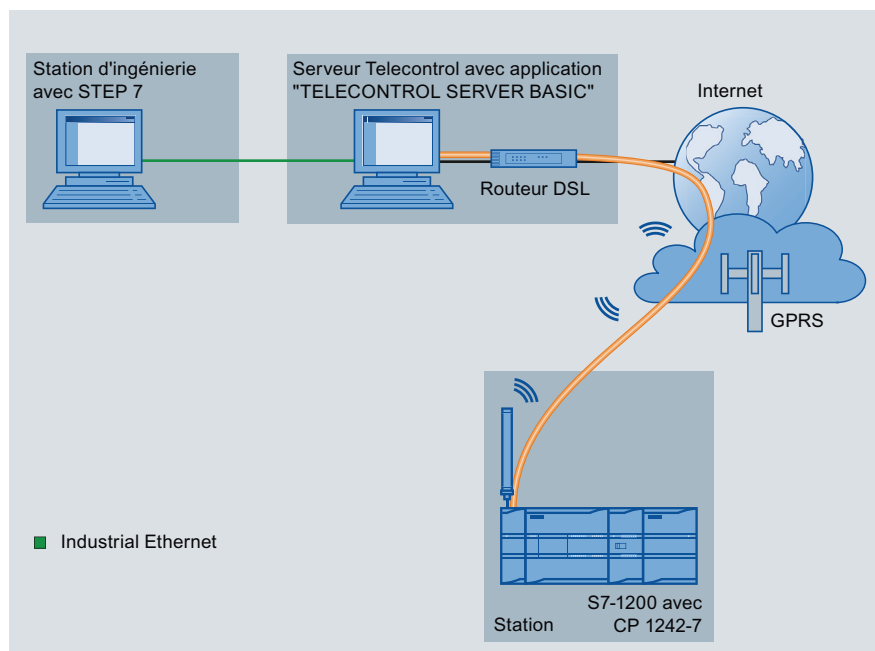


Figure 12-6 TeleService via GPRS dans une configuration avec serveur Telecontrol

### TeleService sans serveur Telecontrol

La liaison passe par la passerelle TeleService.

La liaison entre la station d'ingénierie et la passerelle TeleService peut passer soit par un réseau local soit par Internet.

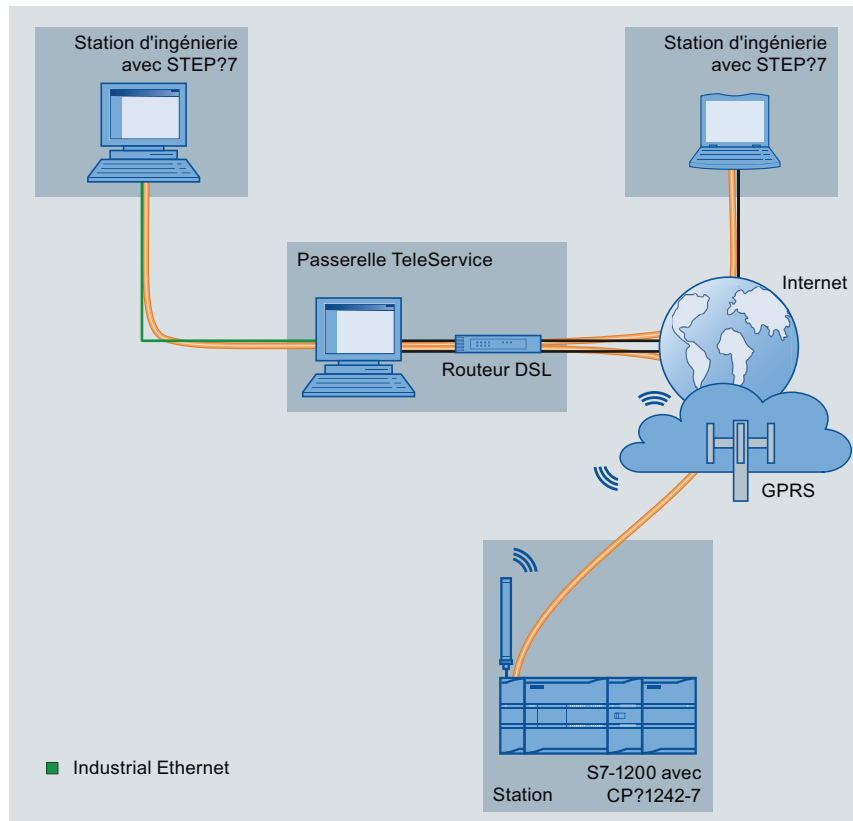


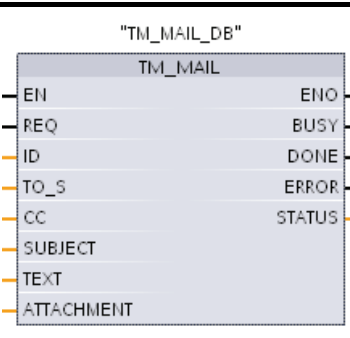
Figure 12-7 TeleService via GPRS dans une configuration avec passerelle TeleService





## 13.1 Instruction TM\_Mail (Transférer e-mail)

Tableau 13- 1 Instruction TM\_MAIL

CONT/LOG	SCL	Description
 <p>The diagram shows a block labeled "TM_MAIL" with several input and output parameters. On the left, inputs are listed: EN, REQ, ID, TO_S, CC, SUBJECT, TEXT, and ATTACHMENT. On the right, outputs are listed: ENO, BUSY, DONE, ERROR, and STATUS. Arrows indicate the direction of data flow between the block and these parameters.</p>	<pre>"TM_MAIL_DB" (     REQ:=_bool_in_,     ID:=_int_in_,     TO_S:=_string_in_,     CC:=_string_in_,     SUBJECT:=_string_in_,     TEXT:=_string_in_,     ATTACHMENT:=_variant_in_,     BUSY=&gt;_bool_out_,     DONE=&gt;_bool_out_,     ERROR=&gt;_bool_out_,     STATUS=&gt;_word_out_);</pre>	<p>L'instruction TM_MAIL envoie un courrier électronique à l'aide du protocole SMTP (Simple Mail Transfer Protocol) sur TCP/IP par le biais de la liaison CPU Industrial Ethernet. En l'absence de connectivité Internet basée sur Ethernet, on peut utiliser un adaptateur TeleService optionnel pour se connecter aux lignes téléphoniques terrestres. TM_MAIL s'exécute de manière asynchrone et la tâche s'étend sur plusieurs appels TM_MAIL. Vous devez affecter un DB d'instance lorsque vous appelez TM_MAIL. <b>Ce DB d'instance ne doit pas être défini comme rémanent.</b> Ainsi, le DB d'instance est initialisé lors du passage de la CPU de l'état ARRET à l'état MARCHE et une nouvelle opération TM_MAIL peut être déclenchée.</p>

<sup>1</sup> STEP 7 crée automatiquement le DB d'instance lorsque vous insérez l'instruction.

L'envoi du courriel démarre en présence d'un front montant (de 0 à 1) dans le paramètre d'entrée REQ. Le tableau suivant montre les relations entre BUSY, DONE et ERROR. Vous pouvez surveiller l'avancement de l'exécution de TM\_MAIL et détecter l'achèvement de l'instruction en évaluant ces paramètres lors d'appels successifs.

Les paramètres de sortie DONE, ERROR, STATUS et SFC\_STATUS ne sont valables que pendant un cycle lorsque l'état du paramètre de sortie BUSY passe de 1 à 0. La logique de votre programme doit sauvegarder les valeurs temporaires de l'état de sortie de manière à ce que vous puissiez détecter des changements d'état lors des cycles d'exécution suivants du programme.

Tableau 13- 2 Interactions entre les paramètres Done, Busy et Error

DONE	BUSY	ERROR	Description
Sans objet	1	Sans objet	La tâche est en cours.
1	0	0	La tâche a été achevée avec succès.
0	0	1	La tâche s'est terminée avec une erreur. Vous trouverez la cause de l'erreur dans le paramètre STATUS.
0	0	0	Il n'y a pas de tâche en cours.

13.1 Instruction TM\_Mail (Transférer e-mail)

Si la CPU passe à l'état ARRET alors que TM\_MAIL est actif, la liaison de communication au serveur de courriel est coupée. La liaison de communication au serveur de courriel est également interrompue si des problèmes surviennent dans la communication CPU sur le bus Industrial Ethernet. Dans ces cas, l'opération d'envoi est suspendue et le courriel n'atteint pas son destinataire.

**PRUDENCE**

**Modification des programmes utilisateur**

Ne modifiez les parties de votre programme utilisateur qui affectent directement les appels de TM\_MAIL que lorsque :

- la CPU est à l'état ARRET,
- aucun courriel n'est envoyé (REQ et BUSY = 0).

Cela vaut en particulier pour la suppression et le remplacement de blocs de programme, les appels de TM\_MAIL ou les appels des DB d'instance de TM\_MAIL.

Si vous ne parvenez pas à conserver le lien entre les blocs de programme, les fonctions de communication TCP/IP peuvent prendre un état indéfini. Vous devez exécuter un démarrage à chaud ou à froid de la CPU après avoir transféré un bloc de programme modifié.

**Cohérence des données**

Le paramètre d'entrée ADDR\_MAIL\_SERVER est lu au démarrage de l'opération. Une nouvelle valeur ne prend pas effet tant que l'opération en cours n'est pas achevée et qu'une nouvelle opération TM\_MAIL n'est pas déclenchée.

En revanche, les paramètres WATCH\_DOG\_TIME, TO\_S, CC, FROM, SUBJECT, TEXT, ATTACHMENT, USERNAME et PASSWORD sont lus pendant l'exécution de TM\_MAIL et ne peuvent être modifiés qu'une fois la tâche achevée (BUSY = 0).

**Connexion de numérotation : configuration des paramètres IE de l'adaptateur TS**

Vous devez configurer les paramètres IE de l'adaptateur TeleService pour les appels sortants afin d'établir la liaison au serveur de numérotation de votre fournisseur de services Internet. Si vous activez l'attribut d'appel sur demande, la liaison est établie uniquement lorsqu'un courriel est envoyé. En cas de connexion avec un modem analogique, il faut plus de temps pour établir la liaison (environ une minute en plus). Vous devez inclure ce temps supplémentaire dans la valeur WATCH\_DOG\_TIME.

Tableau 13- 3 Types de données pour les paramètres

Paramètre et type		Types de données	Description
REQ	IN	Bool	Un front montant déclenche l'opération.
ID	IN	Int	Identificateur de la liaison. Voir le paramètre ID des instructions TCON, TDISCON, TSEND et TRCV. Il faut utiliser un nombre qui n'est pas utilisé pour d'autres instances de cette instruction dans le programme utilisateur.

Paramètre et type		Types de données	Description
TO_S	IN	String	Adresses des destinataires. Données STRING de 240 caractères au maximum.
CC	IN	String	Adresses de copie conforme (facultatives). Données STRING de 240 caractères au maximum.
SUBJECT	IN	String	Objet du courriel. Données STRING de 240 caractères au maximum.
TEXT	IN	String	Texte du courriel (facultatif). Données STRING de 240 caractères au maximum. Si ce paramètre est une chaîne vide, le courriel sera envoyé sans texte.
ATTACHMENT	IN	Variant	Pointeur désignant les données des pièces jointes. Données octet, mot ou double mot de 65 534 octets au maximum. Si vous n'affectez pas de valeur à ce paramètre, le courriel sera envoyé sans pièce jointe.
DONE	OUT	Bool	<ul style="list-style-type: none"> <li>0 : Tâche pas encore commencée ou encore en cours d'exécution</li> <li>1 : Tâche exécutée sans erreur</li> </ul>
BUSY	OUT	Bool	<ul style="list-style-type: none"> <li>0 : Pas d'opération en cours</li> <li>1 : Opération en cours</li> </ul>
ERROR	OUT	Bool	Le bit ERROR est égal à 1 pendant un cycle lorsque la dernière demande s'est achevée avec une erreur. La valeur de code d'erreur dans la sortie STATUS ne vaut que pendant le cycle où ERROR est égal à 1.
STATUS	OUT	Word	Valeur ou information d'erreur renvoyée par l'instruction TM_MAIL
ADDR_MAIL_SERVER	<sup>1</sup> Static	DWord	Adresse IP du serveur de courriel. Vous devez définir chaque fragment d'adresse IP en tant qu'octet de deux caractères hexadécimaux de 4 bits. Si un fragment d'adresse est égal à la valeur décimale 10 qui vaut A en hexadécimal, vous devez entrer "0A" pour cet octet. Par exemple, pour l'adresse IP = 192.168.0.10 : ADDR_MAIL_SERVER = DW#16#C0A8000A, car : <ul style="list-style-type: none"> <li>192 = 16#C0,</li> <li>168 = 16#A8</li> <li>0 = 16#00</li> <li>10 = 16#0A</li> </ul>
WATCH_DOG_TIME	<sup>1</sup> Static	Time	Temps maximum autorisé pour que TM_MAIL établisse une liaison au serveur. Si ce temps est dépassé, l'exécution de TM_MAIL s'achève avec une erreur. Le temps de retard effectif jusqu'à ce que TM_MAIL s'achève et que l'erreur soit émise peut dépasser WATCH_DOG_TIME en raison du temps supplémentaire requis pour l'opération de déconnexion. Définissez le temps à 2 minutes pour commencer. Cette valeur peut être bien inférieure pour une liaison téléphonique RNIS.

13.1 Instruction TM\_Mail (Transférer e-mail)

Paramètre et type		Types de données	Description
USERNAME	<sup>1</sup> Static	String	Nom d'utilisateur du compte de messagerie. Données STRING de 180 caractères au maximum.
PASSWORD	<sup>1</sup> Static	String	Mot de passe du serveur de messagerie. Données STRING de 180 caractères au maximum.
FROM	<sup>1</sup> Static	String	Adresse de l'émetteur. Chaîne STRING avec 240 caractères au maximum.
SFC_STATUS	<sup>1</sup> Static	Word	Code de condition d'exécution des blocs de communication appelés

<sup>1</sup> Les valeurs de ces paramètres ne sont pas modifiées à chaque appel de TM\_MAIL. Elles sont définies dans le bloc de données d'instance de TM\_MAIL et ne sont référencées qu'une fois, à savoir lors du premier appel de TM\_MAIL.

### Authentification SMTP

TM\_MAIL prend en charge la méthode d'authentification SMTP AUTH LOGIN. Pour plus d'informations sur cette méthode d'authentification, consultez le manuel de votre serveur de courriel ou le site Web de votre fournisseur de services Internet.

La méthode d'authentification AUTH LOGIN utilise les paramètres USERNAME et PASSWORD de TM\_MAIL pour se connecter au serveur de courriel. Le nom d'utilisateur et le mot de passe doivent être préalablement configurés dans un compte de messagerie sur un serveur de courriel.

Si aucune valeur n'est affectée au paramètre USERNAME, la méthode d'authentification AUTH LOGIN n'est pas utilisée et le courriel est envoyé sans authentification.

### Paramètres TO\_S:, CC: et FROM:

Les paramètres TO\_S:, CC: et FROM: sont des chaînes, comme illustré dans les exemples suivants :

TO\_S: <wenna@mydomain.com>, <ruby@mydomain.com>,

CC: <admin@mydomain.com>, <judy@mydomain.com>,

FROM: <admin@mydomain.com>

Il faut respecter les règles suivantes lors de l'entrée de ces chaînes de caractères :

- Il faut entrer les caractères "TO\_S:", "CC:" et "FROM:" sans oublier le deux-points.
- Il faut insérer un espace et le signe "<" avant chaque adresse. Il doit, par exemple, y avoir un espace entre "TO\_S:" et <adresse électronique>.
- Il faut terminer chaque adresse par le signe ">".
- Il faut entrer une virgule "," après chaque adresse électronique pour les adresses TO\_S: et CC:. Par exemple, la virgule est obligatoire après l'unique adresse électronique dans "TO\_S: <email address>,".
- L'entrée FROM: ne doit contenir qu'une seule adresse électronique, sans virgule à la fin.

En raison du mode d'exécution et de l'utilisation de la mémoire, la syntaxe des données TO\_S:, CC: et FROM: de TM\_MAIL n'est pas vérifiée. Si les règles de format précédentes ne sont pas exactement respectées, la transaction avec le serveur de courriel SMTP échouera.

## Paramètres STATUS et SFC\_STATUS

Les codes d'erreur d'exécution renvoyés par TM\_MAIL peuvent être classés comme suit :

- W#16#0000 : L'instruction TM\_MAIL s'est achevée avec succès.
- W#16#7xxx : Etat de TM\_MAIL
- W#16#8xxx : Une erreur s'est produite lors d'un appel interne adressé à un appareil de communication ou au serveur de messagerie.

Le tableau suivant présente les codes d'erreur d'exécution de TM\_MAIL à l'exception des codes d'erreur des modules de communication appelés en interne.

---

### Remarque

#### Exigences concernant le serveur de courriel

TM\_MAIL ne peut communiquer qu'avec un serveur de courriel utilisant SMTP via le port 25. Le numéro de port affecté ne peut pas être modifié.

La plupart des services informatiques et des serveurs de courriel externes bloquent maintenant le port 25 pour empêcher un PC infecté par un virus de devenir un générateur de courriels indésirables.

Vous pouvez vous connecter à un serveur de courriel interne via SMTP et laisser le serveur interne gérer les améliorations de sécurité actuelles qui sont nécessaires pour transmettre le courriel à un serveur de courriel externe via Internet.

---

## Exemple de configuration de serveur de courriel interne

Si vous utilisez Microsoft Exchange comme serveur de courriel interne, vous pouvez configurer le serveur pour qu'il autorise l'accès SMTP depuis l'adresse IP affectée à l'API S7-1200. Configurez la console de gestion Exchange : Configuration du serveur > Transport Hub > Connecteurs de réception > Relais IP. L'onglet Réseau comprend une boîte intitulée "Recevoir du courrier à partir de serveurs distants ayant ces adresses IP". C'est là que vous ajouterez l'adresse IP de l'API qui exécute l'instruction TM\_MAIL. Aucune authentification n'est requise pour ce type de connexion avec un serveur Microsoft Exchange interne.

## Configuration du serveur de courriel

TM\_MAIL peut utiliser uniquement un serveur de courriel qui autorise la communication via le port 25, SMTP et l'authentification AUTH LOGIN (facultatif).

Configurez un compte de serveur de courriel compatible afin qu'il accepte l'ouverture de session SMTP à distance. Puis éditez le DB d'instance de TM\_MAIL pour y ajouter les chaînes de caractères USERNAME et PASSWORD de TM\_MAIL qui sont utilisées pour authentifier la connexion avec votre compte de messagerie.

13.1 Instruction TM\_Mail (Transférer e-mail)

Tableau 13- 4 Codes d'erreur

STATUS (W#16#...)	SFC_STATUS (W#16#...)	Description
0000	-	L'opération TM_MAIL s'est achevée sans erreur. Ce code STATUS nul ne garantit pas qu'un courriel a effectivement été envoyé (voir le premier point de la remarque suivant ce tableau).
7001	-	TM_MAIL est actif (BUSY = 1).
7002	7002	TM_MAIL est actif (BUSY = 1).
8xxx	xxxx	L'opération TM_MAIL s'est achevée avec une erreur dans les appels d'instruction de communication internes. Pour plus d'informations sur le paramètre SFC_STATUS, consultez les descriptions du paramètre STATUS des instructions de communication ouverte PROFINET sous-jacentes.
8010	xxxx	La connexion a échoué. Pour plus d'informations sur le paramètre SFC_STATUS, consultez la description du paramètre STATUS de l'instruction TCON.
8011	xxxx	Erreur lors de l'envoi des données. Pour plus d'informations sur le paramètre SFC_STATUS, consultez la description du paramètre STATUS de l'instruction TSEND.
8012	xxxx	Erreur lors de la réception des données. Pour plus d'informations sur le paramètre SFC_STATUS, consultez la description du paramètre STATUS de l'instruction TRCV.
8013	xxxx	La connexion a échoué. Pour plus d'informations sur l'évaluation du paramètre SFC_STATUS, consultez les descriptions du paramètre STATUS des instructions TCON et TDISCON.
8014	-	La connexion a échoué. Vous avez peut-être entré une adresse IP de serveur de courriel incorrecte (ADDR_MAIL_SERVER) ou indiqué trop peu de temps (WATCH_DOG_TIME) pour la connexion. Il est également possible que la CPU n'ait pas de liaison au réseau ou que la configuration de la CPU soit incorrecte.
82xx, 84xx, 85xx	-	Le message d'erreur provient du serveur de courriel et correspond au code d'erreur "8" du protocole SMTP. Voir le deuxième point de la remarque suivant ce tableau.
8450	-	Pas d'exécution de l'opération : boîte aux lettres électronique non disponible. Réessayez ultérieurement.
8451	-	Annulation de l'opération : erreur locale lors du traitement. Réessayez ultérieurement.
8500	-	Erreur de syntaxe de commande : peut-être le serveur de courriel n'accepte-t-il pas la procédure d'authentification LOGIN. Vérifiez les paramètres de TM_MAIL. Essayez d'envoyer un courriel sans authentification. Tentez de remplacer le paramètre USERNAME par une chaîne vide.
8501	-	Erreur de syntaxe : paramètre ou argument incorrect. Vous avez peut-être tapé une adresse incorrecte dans les paramètres TO_S ou CC.
8502	-	Commande inconnue ou non implémentée. Vérifiez vos entrées, en particulier le paramètre FROM. Il est peut-être incomplet ou vous avez omis les caractères "@" ou ".".
8535	-	Authentification SMTP inachevée. Vous avez peut-être entré un nom d'utilisateur ou un mot de passe incorrect.

STATUS (W#16#...)	SFC_STATUS (W#16#...)	Description
8550	-	Impossible d'atteindre le serveur de courriel ou vous n'avez pas de droits d'accès. Vous avez peut-être entré un nom d'utilisateur ou un mot de passe incorrect ou votre serveur de courriel n'accepte pas l'accès par ouverture de session. Cette erreur peut également être due à une entrée erronée du nom de domaine après le signe "@" dans les paramètres TO_S ou CC.
8552	-	Annulation de l'opération : dépassement de la taille de mémoire allouée. Réessayez ultérieurement.
8554	-	L'émission a échoué. Réessayez ultérieurement.

---

**Remarque****Possibilité d'erreurs d'envoi de courriel non signalées**

- L'entrée incorrecte d'une adresse de destinataire ne génère pas d'erreur STATUS pour TM\_MAIL. Dans ce cas, il n'y a aucune garantie que les autres destinataires (dont les adresses électroniques ont été correctement indiquées) recevront le courriel.
  - Vous trouverez plus d'informations sur les codes d'erreur SMTP sur Internet ou dans la documentation d'erreur du serveur de courriel. Vous pouvez également lire le dernier message d'erreur en provenance du serveur de courriel. Le message d'erreur est stocké dans le paramètre buffer1 du DB d'instance de TM\_MAIL.
-





## Outils en ligne et de diagnostic

### 14.1 DEL d'état

La CPU et les modules d'E/S utilisent des DEL pour fournir des informations sur l'état de fonctionnement du module ou des E/S.

#### DEL d'état sur une CPU

La CPU fournit les indicateurs d'état suivants :

- STOP/RUN
  - Jaune continu indique l'état ARRET.
  - Vert continu indique l'état MARCHE.
  - Vert et jaune clignotant en alternance indiquent que la CPU est à l'état MISE EN ROUTE.
- ERROR
  - Rouge clignotant indique une erreur, telle qu'une erreur interne dans la CPU, une erreur avec la carte mémoire ou une erreur de configuration (modules non concordants).
  - Rouge continu signale un matériel défectueux.
- La DEL MAINT (Maintenance) clignote dès que vous insérez une carte mémoire. La CPU passe alors à l'état ARRET. Une fois que la CPU est passée à l'état ARRET, exécutez l'une des actions suivantes pour déclencher l'évaluation de la carte mémoire :
  - Faites passer la CPU à l'état MARCHE.
  - Effectuez un effacement général (MRES).
  - Mettez la CPU hors tension puis sous tension.

Vous pouvez également utiliser l'instruction LED (Page 318) pour déterminer l'état des DEL.

Tableau 14- 1 DEL d'état pour une CPU

Description	STOP/RUN jaune / vert	ERROR rouge	MAINT jaune
Hors tension	Eteint	Eteint	Eteint
Démarrage, auto-test ou actualisation du firmware	Clignotant (jaune et vert en alternance)	-	Eteint
Etat ARRET	Allumé (jaune)	-	-
Etat MARCHE	Allumé (vert)	-	-
Enlever la carte mémoire	Allumé (jaune)	-	Clignotant
Erreur	Allumé (soit jaune soit vert)	Clignotant	-
Maintenance requise	Allumé (soit jaune soit vert)	-	Allumé

14.1 DEL d'état

Description	STOP/RUN jaune / vert	ERROR rouge	MAINT jaune
Matériel défectueux	Allumé (jaune)	Allumé	Eteint
Test des DEL ou firmware CPU défectueux	Clignotant (jaune et vert en alternance)	Clignotant	Clignotant

La CPU fournit également deux DEL qui indiquent l'état de la communication PROFINET. Ouvrez le cache du bornier inférieur pour voir les DEL PROFINET.

- Link (vert) s'allume pour signaler qu'une connexion a été établie avec succès.
- Rx/Tx (jaune) s'allume pour signaler une activité de transmission.

La CPU et chaque module d'entrées-sorties TOR (SM) fournissent une DEL I/O Channel pour chacune des entrées et sorties TOR. La DEL I/O Channel (verte) s'allume ou s'éteint pour indiquer l'état de l'entrée ou sortie correspondante.

**DEL d'état sur un SM**

En outre, chaque module SM TOR fournit une DEL DIAG qui indique l'état du module :

- Vert indique que le module est opérationnel.
- Rouge indique que le module est défectueux ou n'est pas opérationnel.

Chaque module d'entrées-sorties analogiques fournit une DEL I/O Channel pour chacune des entrées et sorties analogiques.

- Vert indique que la voie a été configurée et est active.
- Rouge signale une situation d'erreur de l'entrée ou sortie analogique correspondante.

En outre, chaque module SM analogique fournit une DEL DIAG qui indique l'état du module :

- Vert indique que le module est opérationnel.
- Rouge indique que le module est défectueux ou n'est pas opérationnel.

Le module d'entrées-sorties détecte la présence ou l'absence de courant vers le module (alimentation côté site, si requis).

Tableau 14- 2DEL d'état pour un module d'entrées-sorties (SM)

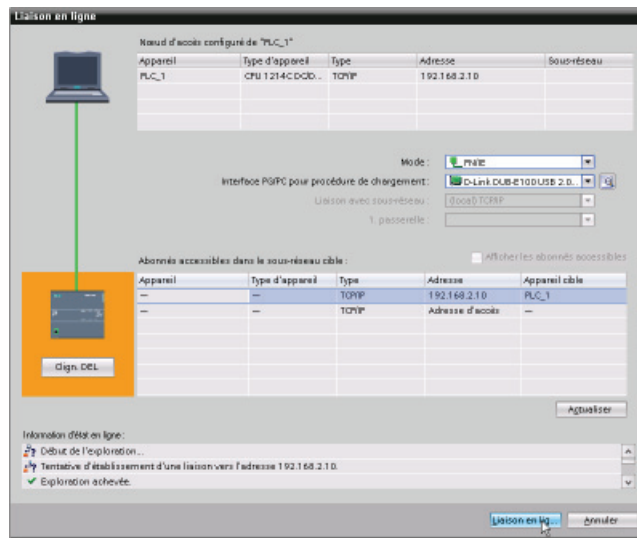
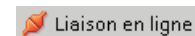
Description	DIAG (rouge / vert)	I/O Channel (rouge / vert)
L'alimentation côté site est coupée.	Rouge clignotant	Rouge clignotant
Non configuré ou mise à jour en cours	Vert clignotant	Eteint
Module configuré sans erreur	Allumé (vert)	Allumé (vert)
Situation d'erreur	Rouge clignotant	-
Erreur d'E/S (avec diagnostic activé)	-	Rouge clignotant
Erreur d'E/S (avec diagnostic désactivé)	-	Allumé (vert)

## 14.2 Passage en ligne et connexion à une CPU

Une liaison en ligne entre la console de programmation et la CPU est nécessaire pour charger des programmes et des données techniques de projet, ainsi que pour des activités telles que les suivantes :

- Test des programmes utilisateur
- Affichage et changement de l'état de fonctionnement de la CPU (Page 727)
- Affichage et réglage de la date et de l'heure de la CPU (Page 726)
- Affichage des informations sur les modules
- Comparaison et synchronisation (Page 729) hors ligne par rapport à des blocs de programme en ligne
- Chargement de blocs de programme depuis la CPU et dans la CPU
- Affichage du diagnostic et de la mémoire tampon de diagnostic (Page 728)
- Utilisation d'une table de visualisation (Page 733) pour tester le programme utilisateur par visualisation et forçage de valeurs
- Utilisation d'une table de forçage permanent pour forcer des valeurs de manière permanente dans la CPU (Page 736)

Pour établir une liaison en ligne à une CPU configurée, cliquez sur la CPU dans l'arborescence du projet et cliquez sur le bouton "Liaison en ligne" dans la vue du projet :



Si c'est la première fois que vous allez en ligne avec cette CPU, vous devez sélectionner le type d'interface PG/PC et l'interface PG/PC spécifique dans la boîte de dialogue Liaison en ligne avant d'établir une liaison en ligne à une CPU trouvée sur cette interface.

Votre console de programmation est maintenant connectée à la CPU. Les cadres de couleur orange correspondent à une liaison en ligne. Vous pouvez désormais utiliser les outils en ligne et de diagnostic dans l'arborescence du projet et dans la Task Card Outils en ligne.

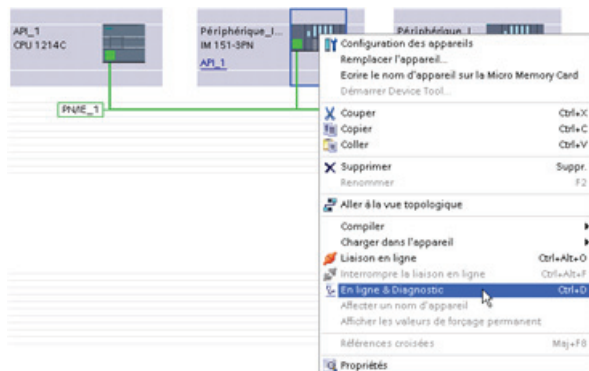
## 14.3 Affectation d'un nom à un périphérique PROFINET IO en ligne

Vous devez affecter un nom aux appareils sur votre réseau PROFINET pour pouvoir vous connecter à la CPU. Servez-vous de l'éditeur "Appareils & réseaux" pour affecter un nom à vos appareils PROFINET s'ils n'en ont pas déjà un ou s'il faut le modifier.

Vous devez définir, pour chaque périphérique PROFINET IO, le même nom dans le projet STEP 7 et, à l'aide de l'outil "En ligne & diagnostic", dans la mémoire de configuration de périphérique PROFINET IO (par exemple, mémoire de configuration de coupleur ET200S). Si un nom manque ou ne coïncide pas à l'un ou l'autre endroit, le mode d'échange de données PROFINET IO ne fonctionnera pas.

## 14.3 Affectation d'un nom à un périphérique PROFINET IO en ligne

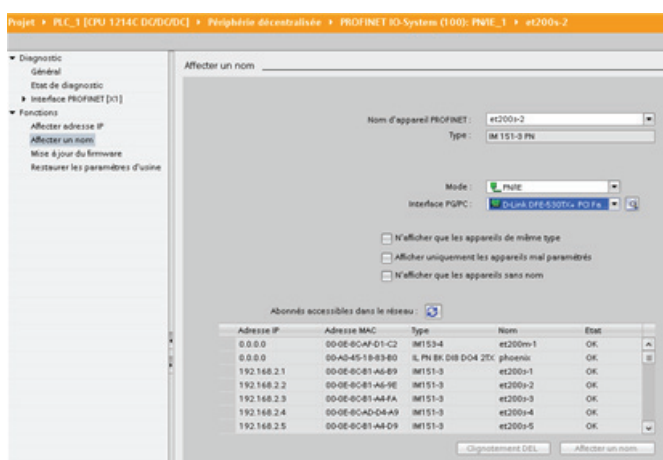
1. Dans l'éditeur "Appareils & réseaux", cliquez avec le bouton droit de la souris sur le périphérique PROFINET IO concerné et sélectionnez "En ligne & diagnostic".



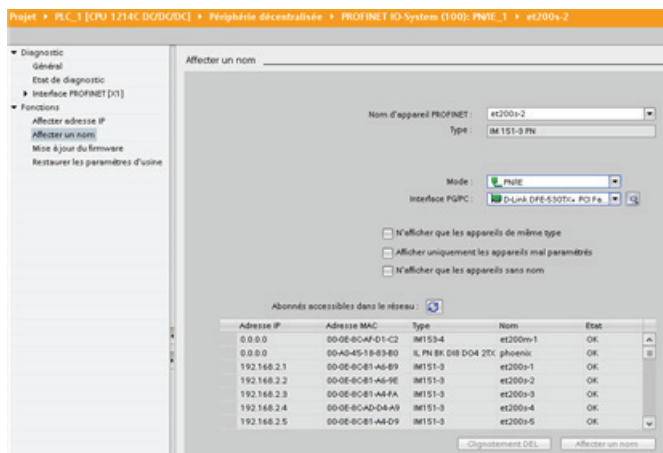
2. Faites les sélections de menu suivantes dans la boîte de dialogue "En ligne & Diagnostic" :

- "Fonctions"
- "Affecter un nom"

Cliquez sur l'icône "Abonnés accessibles dans le réseau" pour afficher tous les périphériques PROFINET IO dans le réseau.

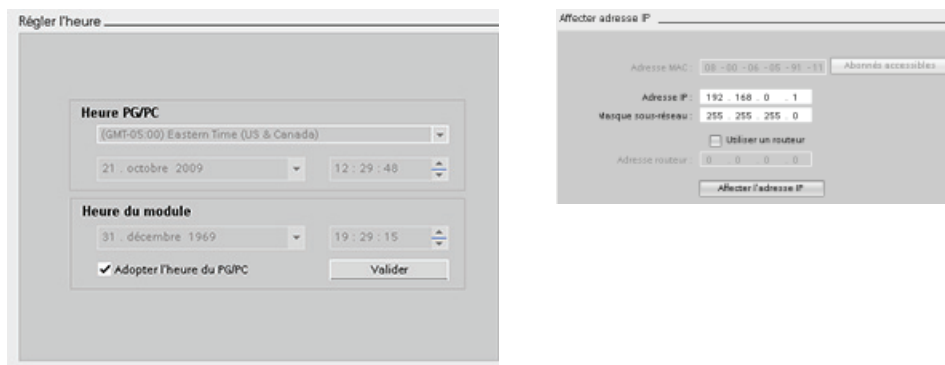


3. Dans la liste affichée, cliquez sur le périphérique PROFINET IO concerné et cliquez sur le bouton "Affecter un nom" pour écrire le nom dans la mémoire de configuration de périphérique PROFINET IO.



## 14.4 Réglage de l'adresse IP et de l'heure

Vous pouvez régler l'adresse IP (Page 144) et l'heure dans la CPU en ligne. Après avoir accédé à "En ligne et Diagnostics" depuis l'arborescence du Projet pour une CPU en ligne, vous pouvez afficher ou modifier l'adresse IP. Vous pouvez également afficher ou régler les paramètres de date et d'heure de la CPU en ligne.



---

### Remarque

Cette fonction est disponible uniquement pour une CPU qui soit dispose uniquement d'une adresse MAC (pas encore d'adresse IP attribuée), soit a été réinitialisée aux réglages d'usine.

---

## 14.5 Restauration des réglages d'usine

Vous pouvez restaurer les réglages d'usine d'origine d'un S7-1200 dans les conditions suivantes :

- Aucune carte mémoire n'est insérée dans la CPU.
- La CPU dispose d'une liaison en ligne.
- La CPU est à l'état ARRET.

---

### Remarque

Si la CPU est à l'état MARCHE et que vous lancez l'opération de restauration, vous pourrez la faire passer à l'état ARRET après acquittement d'un message de confirmation.

---

## Marche à suivre

Procédez comme suit pour restaurer les réglages d'usine d'une CPU :

1. Ouvrez la vue En ligne & Diagnostic de la CPU.
2. Sélectionnez "Restaurer les réglages d'usine" dans le dossier "Fonctions".
3. Cochez la case "Conserver l'adresse IP" si vous voulez conserver l'adresse IP ou "Réinitialiser l'adresse IP" si vous voulez effacer l'adresse IP.
4. Cliquez sur le bouton "Restaurer".
5. Acquitez le message de confirmation en cliquant sur "OK".

## Résultat

Le module passe à l'état ARRET si nécessaire et les réglages d'usine sont restaurés :

- La mémoire de travail, la mémoire de chargement interne et toutes les zones d'opérandes sont effacées.
- Tous les paramètres reprennent leur valeur par défaut.
- La mémoire tampon de diagnostic est effacée.
- L'heure est réinitialisée.
- L'adresse IP est conservée ou effacée selon votre sélection (l'adresse MAC est fixe et n'est jamais modifiée).

## 14.6 Panneau de commande CPU de la CPU en ligne



Le "Panneau de commande CPU" affiche l'état de fonctionnement (ARRET ou MARCHÉ) de la CPU en ligne. Il montre également si la CPU a rencontré une erreur ou si des valeurs sont en cours de forçage.

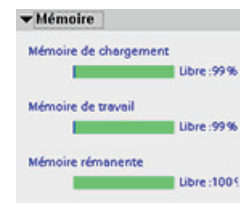
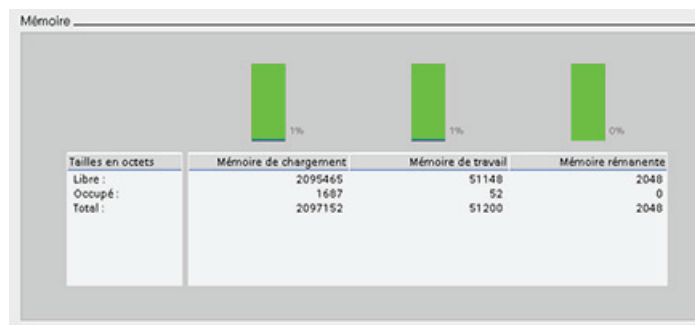
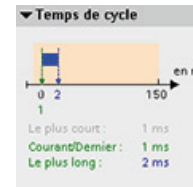
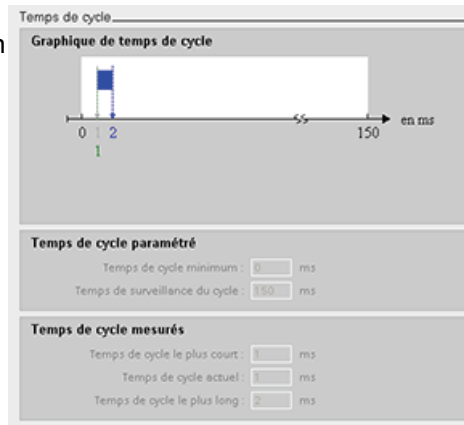
Utilisez le panneau de commande CPU de la Task Card Outils en ligne pour changer l'état de fonctionnement d'une CPU en ligne. La Task Card Outils en ligne est accessible dès que la CPU est en ligne.

## 14.7 Surveillance du temps de cycle et de l'utilisation de la mémoire

Vous pouvez surveiller le temps de cycle et l'utilisation de la mémoire d'une CPU en ligne.

Après connexion à la CPU en ligne, ouvrez la Task Card Outils en ligne pour visualiser les mesures suivantes :

- Temps de cycle
- Utilisation de la mémoire

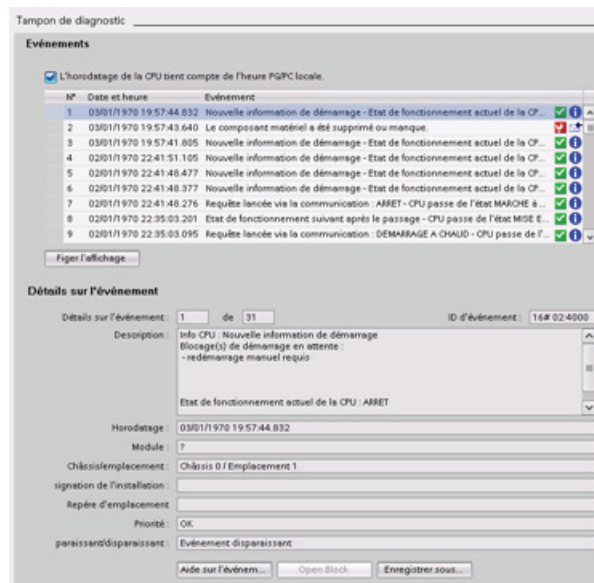


## 14.8 Affichage des événements de diagnostic dans la CPU

Servez-vous de la mémoire tampon de diagnostic pour visualiser les activités récentes dans la CPU. La mémoire tampon de diagnostic est accessible à partir de "En ligne & diagnostic" pour une CPU en ligne dans l'arborescence du projet. Elle contient les entrées suivantes :

- Evénements de diagnostic
- Changements d'état de fonctionnement de la CPU (passages à l'état ARRET ou MARCHE)





La première entrée contient l'événement le plus récent. Chaque entrée dans la mémoire tampon de diagnostic contient la date et l'heure auxquelles l'événement a été consigné, ainsi qu'une description.

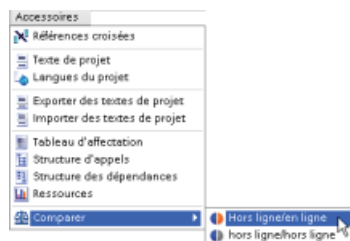
Le nombre maximum d'entrées dépend de la CPU. 50 entrées au maximum sont acceptées.

Seuls les dix événements les plus récents dans la mémoire tampon de diagnostic sont stockés de façon permanente. La restauration des réglages d'usine dans la CPU réinitialise la mémoire tampon de diagnostic en y effaçant les entrées.

Vous pouvez également utiliser l'instruction GET\_DIAG (Page 322) pour collecter les informations de diagnostic.

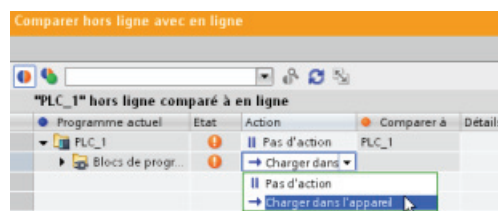
## 14.9 Comparaison de CPU hors ligne et en ligne

Vous pouvez comparer les blocs de code dans une CPU en ligne avec les blocs de code de votre projet. Si les blocs de code de votre projet ne correspondent pas à ceux de la CPU en ligne, l'éditeur de comparaison vous permet de synchroniser votre projet avec la CPU en ligne soit en chargeant les blocs de code de votre projet dans la CPU, soit en effaçant les blocs du projet qui n'existent pas dans la CPU en ligne.



Sélectionnez la CPU dans votre projet.

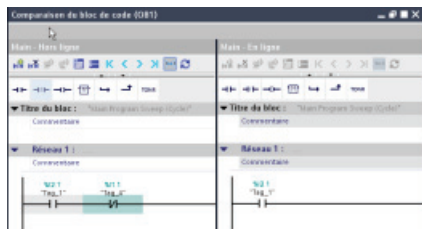
Utilisez la commande "Comparer hors ligne/en ligne" pour ouvrir l'éditeur "Compare" (accédez à la commande soit depuis le menu "Outils", soit en cliquant avec le bouton droit de la souris sur la CPU dans votre projet).



Cliquez dans la colonne "Action" d'un objet et choisissez de supprimer l'objet, de ne rien faire ou de charger l'objet dans l'appareil.

Cliquez sur le bouton "Synchroniser" pour charger les blocs de code.

14.10 Visualisation et forçage de valeurs dans la CPU



Cliquez avec le bouton droit de la souris sur un objet dans la colonne "Comparer à" et cliquez sur le bouton "Démarrer la comparaison détaillée" pour afficher les blocs de code côte à côte.

La comparaison détaillée met en évidence les différences entre les blocs de code de la CPU en ligne et les blocs de code de la CPU dans votre projet.

## 14.10 Visualisation et forçage de valeurs dans la CPU

STEP 7 fournit des outils en ligne pour la surveillance de la CPU :

- Vous pouvez visualiser les valeurs en cours des variables. La fonction de visualisation ne modifie pas la séquence du programme. Elle vous donne des informations sur la séquence du programme et les données du programme dans la CPU.
- Vous pouvez également utiliser d'autres fonctions pour commander la séquence et les données du programme utilisateur :
  - Vous pouvez forcer la valeur des variables dans la CPU en ligne pour voir comment le programme utilisateur réagit.
  - Vous pouvez forcer de manière permanente une sortie de périphérie (Q0.1:P ou "Demarr":P, par exemple) à une valeur spécifique.
  - Vous pouvez débloquer les sorties à l'état ARRET.

---

### Remarque

Soyez toujours prudent lorsque vous utilisez des fonctions de forçage. Ces fonctions peuvent avoir une influence grave sur l'exécution du programme utilisateur/système.

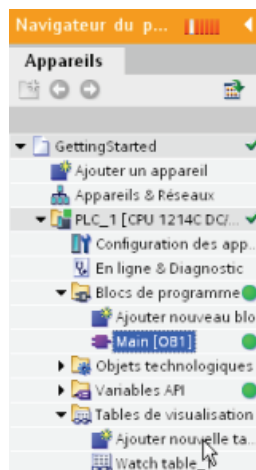
---

Tableau 14- 3 Fonctionnalités en ligne des éditeurs STEP 7

Editeur	Visualisation	Forçage	Forçage permanent
Table de visualisation	Oui	Oui	Non
Table de forçage permanent	Oui	Non	Oui
Editeur de programmes	Oui	Oui	Non
Table des variables	Oui	Non	Non
Editeur de DB	Oui	Non	Non

### 14.10.1 Passage en ligne pour visualiser les valeurs dans la CPU

Pour visualiser les variables, une liaison en ligne doit être établie à la CPU. Il vous suffit de cliquer sur le bouton "Liaison en ligne" de la barre d'outils.



Une fois que vous êtes connecté à la CPU, STEP 7 affiche les entêtes des zones de travail en orange.

L'arborescence du projet affiche une comparaison du projet hors ligne et de la CPU en ligne. Un cercle vert signifie que la CPU et le projet sont synchronisés, c'est-à-dire que tous deux ont la même configuration et le même programme utilisateur.

Les tables de variables montrent les variables. Les tables de visualisation peuvent également montrer les variables, ainsi que les adresses directes.

	Nom	Adresse	Format d'affichage	Valeur visualisati...	Valeur de forçage
1	*On*	%I0.0	Bool		
2	*Off*	%I0.1	Bool		
3	*Run*	%Q0.0	Bool		

Pour visualiser l'exécution du programme utilisateur et afficher les valeurs des variables, cliquez sur le bouton "Visualiser tout" de la barre d'outils.

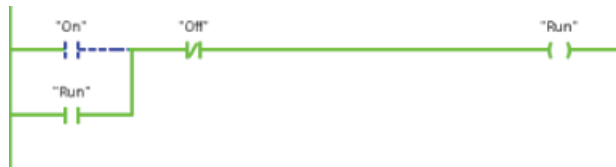
	Nom	Adresse	Format d'affichage	Valeur visualisati...	Valeur de forçage
1	*On*	%I0.0	Bool	<input type="checkbox"/> FALSE	
2	*Off*	%I0.1	Bool	<input type="checkbox"/> FALSE	
3	*Run*	%Q0.0	Bool	<input type="checkbox"/> FALSE	

Le champ "Valeur de visualisation" indique la valeur de chaque variable.

### 14.10.2 Affichage de l'état dans l'éditeur de programme

Vous pouvez visualiser l'état des variables dans les éditeurs de programme CONT et LOG. Utilisez la barre des éditeurs pour afficher l'éditeur CONT. La barre des éditeurs vous permet de basculer entre les éditeurs ouverts sans avoir à les ouvrir ou les fermer.

Dans la barre d'outils de l'éditeur de programme, cliquez sur le bouton "Marche/arrêt visualisation" pour afficher le statut de votre programme utilisateur.



Le réseau dans l'éditeur de programme affiche le flux de courant en vert.

Vous pouvez également faire un clic droit sur l'opération ou le paramètre pour modifier la valeur pour l'opération.

### 14.10.3 Acquisition des valeurs en ligne d'un DB pour redéfinir les valeurs initiales

Vous pouvez capturer les valeurs en cours visualisées dans une CPU en ligne pour en faire les valeurs initiales d'un DB global.

- Une liaison en ligne doit être établie à la CPU.
- La CPU doit être à l'état MARCHE.
- Le DB doit être ouvert dans STEP 7.



Utilisez le bouton "Afficher un instantané des valeurs de visualisation" pour capturer les valeurs en cours des variables sélectionnées dans le DB. Vous pouvez alors copier ces valeurs dans la colonne "Valeur initiale" du DB.

1. Dans l'éditeur de DB, cliquez sur le bouton "Visualiser tout". La colonne "Valeur de visualisation" affiche les valeurs de données en cours.
2. Cliquez sur le bouton "Afficher un instantané des valeurs de visualisation" pour afficher les valeurs en cours dans la colonne "Instantané".
3. Cliquez sur le bouton "Visualiser tout" pour arrêter la visualisation des données dans la CPU.
4. Copiez une valeur de la colonne "Instantané" pour une variable.
  - Sélectionnez la valeur à copier.
  - Cliquez avec le bouton droit de la souris sur la valeur sélectionnée afin d'afficher le menu contextuel.
  - Sélectionnez la commande "Copier".
5. Collez la valeur copiée dans la colonne "Valeur initiale" correspondante de la variable (cliquez avec le bouton droit de la souris sur la cellule et sélectionnez "Coller" dans le menu contextuel).

6. Enregistrez le projet pour configurer les valeurs copiées en tant que nouvelles valeurs initiales pour le DB.
7. Compilez et chargez le DB dans la CPU Le DB utilisera les nouvelles valeurs initiales après le passage de la CPU à l'état MARCHE.

---

**Remarque**

Les valeurs visibles dans la colonne "Valeur de visualisation" sont toujours copiées de la CPU. STEP 7 ne vérifie pas si toutes les valeurs proviennent du même cycle de la CPU.

---

#### 14.10.4 Utilisation d'une table de visualisation pour visualiser et forcer des valeurs dans la CPU

Une table de visualisation vous permet d'exécuter des fonctions de visualisation et de forçage sur des données pendant que la CPU exécute votre programme. Il peut s'agir de données de la mémoire image (I ou Q), de mémentos, de DB ou d'entrées physiques (I\_:P) en fonction de la fonction de visualisation ou de forçage. Vous ne pouvez pas visualiser les sorties physiques (Q\_:P) avec précision, car la fonction de visualisation peut uniquement afficher la dernière valeur écrite depuis la mémoire Q et ne lit pas la valeur réelle dans les sorties physiques.

La fonction de visualisation ne modifie pas la séquence du programme. Elle vous donne des informations sur la séquence du programme et les données du programme dans la CPU.

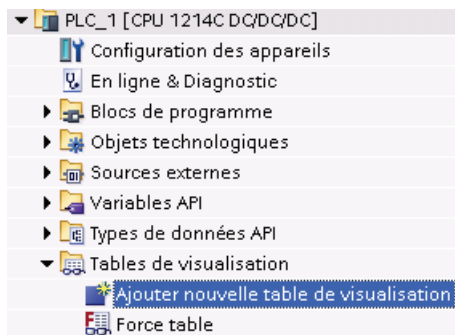
Les fonctions de forçage permettent à l'utilisateur de gérer la séquence et les données du programme. Il faut être prudent lors de l'utilisation des fonctions de forçage. Ces fonctions peuvent avoir une influence grave sur l'exécution du programme utilisateur/système. Les trois fonctions de forçage sont Forçage, Forçage permanent et Débloquer les sorties en STOP.

Vous pouvez exécuter les fonctions en ligne suivantes avec la table de visualisation :

- Visualisation de l'état des variables
- Forçage des valeurs de variables individuelles

Vous pouvez choisir quand vous voulez visualiser ou forcer la variable :

- Au début du cycle : Lecture ou écriture de la valeur au début du cycle
- En fin de cycle : Lecture ou écriture de la valeur à la fin du cycle
- Au passage à l'arrêt



Pour créer une table de visualisation :

1. Double-cliquez sur "Ajouter nouvelle table de visualisation" pour ouvrir une nouvelle table de visualisation.
2. Entrez un nom de variable pour ajouter une variable à la table de visualisation.

Les options suivantes sont disponibles pour la visualisation des variables :

- Visualiser tout : Cette commande lance la visualisation des variables visibles dans la table de visualisation active.
- Visualiser immédiatement : Cette commande lance la visualisation des variables visibles dans la table de visualisation active. La table de visualisation visualise les variables immédiatement et une seule fois.

Les options suivantes sont disponibles pour le forçage des variables :

- "Forcer à 0" met la valeur d'une adresse sélectionnée à "0".
- "Forcer à 1" met la valeur d'une adresse sélectionnée à "1".
- "Forçage immédiat" modifie immédiatement la valeur des adresses sélectionnées pour un cycle.
- "Forcer avec déclenchement" modifie les valeurs des adresses sélectionnées.

Cette fonction ne fournit pas de retour pour indiquer que les adresses sélectionnées ont été effectivement forcées. Si vous avez besoin d'un retour, utilisez la fonction "Forçage immédiat".

- "Débloquer sorties de périphérie" désactive le verrouillage des sorties et est disponible uniquement lorsque la CPU est à l'état ARRET.

Pour visualiser les variables, une liaison en ligne doit être établie à la CPU.

	Nom	Adresse	Format d'affich...	Valeur visualisation	Visualisation avec décl...	Forçage avec décl...	Valeur de forçage
1	"Start"	%I0.0	Bool		Permanent	Permanent	<input type="checkbox"/>
2	"Stop"	%I0.1	Bool		Permanent	Permanent	<input type="checkbox"/>
3	"Running"	%M0.0	Bool		Permanent	Permanent	<input type="checkbox"/>

Vous pouvez sélectionner les diverses fonctions à l'aide des boutons en haut de la table de visualisation.

Entrez le nom de la variable à visualiser et sélectionnez un format d'affichage dans la liste de sélection déroulante. Lorsqu'une liaison en ligne est établie avec la CPU, cliquer sur le bouton "Visualiser" affiche la valeur en cours de la donnée dans le champ "Valeur de visualisation".

### 14.10.4.1 Utilisation d'un déclenchement lors de la visualisation ou du forçage de variables API

Le déclenchement détermine à quel point du cycle l'adresse sélectionnée doit être visualisée ou forcée.

Tableau 14- 4 Types de déclenchement

Déclenchement	Description
Permanent	Collecte les données en continu.
Début du cycle	Permanent : Collecte les données en continu au début du cycle après que la CPU a lu les entrées.
	Unique : Collecte les données une fois au début du cycle après que la CPU a lu les entrées.
Fin du cycle	Permanent : Collecte les données en continu à la fin du cycle avant que la CPU n'écrive dans les sorties.
	Unique : Collecte les données une fois à la fin du cycle avant que la CPU n'écrive dans les sorties.
Commutation en STOP	Permanent : Collecte les données en continu lorsque la CPU passe à l'état ARRET.
	Unique : Collecte les données une fois après que la CPU est passée à l'état ARRET.

Pour forcer une variable API lors d'un événement de déclenchement donné, sélectionnez soit le début soit la fin du cycle.

- Forçage d'une sortie : Le meilleur événement de déclenchement pour forcer une sortie est la fin du cycle, juste avant que la CPU n'écrive dans les sorties.

Visualisez la valeur des sorties au début du cycle pour déterminer quelle valeur est écrite dans les sorties physiques. Visualisez également les sorties avant que la CPU n'écrive les valeurs dans les sorties physiques pour contrôler la logique du programme et effectuer une comparaison avec le comportement réel des E/S.

- Forçage d'une entrée : Le meilleur événement de déclenchement pour forcer une entrée est le début du cycle, juste après que la CPU a lu les entrées et avant que le programme utilisateur n'utilise les valeurs d'entrée.

Si vous forcez des entrées au début du cycle, visualisez également la valeur des entrées à la fin du cycle pour vous assurer que la valeur de l'entrée à la fin du cycle n'a pas changé depuis le début du cycle. S'il y a une différence dans les valeurs, c'est peut-être que votre programme utilisateur écrit par erreur dans une entrée.

Pour diagnostiquer pourquoi la CPU est passée à l'état ARRET, utilisez le déclenchement "Commutation en STOP" pour capturer les dernières valeurs du processus.

#### 14.10.4.2 Débloquage des sorties à l'état ARRET

La table de visualisation vous permet d'écrire dans les sorties lorsque la CPU est à l'état ARRET. Cette fonctionnalité vous permet de contrôler le câblage des sorties et de vérifier que le câble connecté à une broche de sortie déclenche un signal haut ou bas vers la borne de l'appareil de processus auquel il est connecté.



Même si la CPU est à l'état ARRET, débloquer une sortie physique peut activer le point du processus auquel elle est connectée.

Vous pouvez changer l'état des sorties à l'état ARRET lorsque les sorties sont débloquées. Si les sorties sont verrouillées, vous ne pouvez pas les modifier à l'état ARRET.

- Pour permettre la modification des sorties à l'état ARRET, sélectionnez l'option "Débloquer sorties de périphérie" de la commande "Forcer" du menu "En ligne" ou cliquez avec le bouton droit de la souris sur la ligne de la table de visualisation.

Vous ne pouvez pas activer les sorties à l'état Arrêt si vous avez configuré des E/S décentralisées. Une erreur est retournée si vous tentez de le faire.

- Le passage de la CPU à l'état MARCHE désactive l'option "Débloquer sorties de périphérie".
- Si des entrées ou sorties sont forcées de manière permanente, la CPU n'est pas autorisée à débloquer des sorties alors qu'elle est à l'état ARRET. Le forçage permanent doit d'abord être annulé.

### 14.10.5 Forçage permanent de valeurs dans la CPU

#### 14.10.5.1 Utilisation de la table de forçage permanent

Une table de forçage permanent fournit une fonction de "forçage permanent" qui écrase la valeur d'une entrée ou d'une sortie avec une valeur spécifiée pour l'adresse de périphérie d'entrée ou de sortie. La CPU applique cette valeur forcée en permanence à la mémoire image des entrées avant l'exécution du programme utilisateur et à la mémoire image des sorties avant l'écriture des sorties dans les modules.

---

#### Remarque

Les valeurs de forçage permanent sont stockées dans la CPU et non dans la table de forçage permanent.

Vous ne pouvez pas forcer une entrée (adresse I) ou une sortie (adresse Q), mais vous pouvez forcer une entrée ou une sortie de périphérie. La table de forçage permanent ajoute automatiquement ":P" à l'adresse (par exemple, "On":P ou "Run":P).

---



	Nom	Adresse	Format d'affichage	Valeur visualisation	Valeur de forçage permanent	F
1	"On":P	%I0.0:P	Bool		TRUE	<input checked="" type="checkbox"/>
2	"Off":P	%I0.1:P	Bool			<input type="checkbox"/>
3	"Run":P	%Q0.1:P	Bool			<input type="checkbox"/>

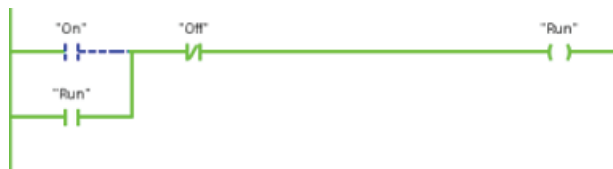
Entrez la valeur pour l'entrée ou la sortie à forcer dans la cellule "Valeur de forçage permanent". Vous pouvez alors utiliser la case à cocher dans la colonne "Forçage permanent" pour activer le forçage permanent de l'entrée ou de la sortie.



Utilisez le bouton "Démarrer ou remplacer forçage permanent" pour forcer la valeur des variables de manière permanente dans la table de forçage permanent. Cliquez sur le bouton "Arrêter forçage permanent" pour arrêter le forçage des variables.

Dans la table de forçage permanent, vous pouvez visualiser l'état de la valeur forcée pour une entrée, mais vous ne pouvez pas visualiser la valeur forcée d'une sortie.

Vous pouvez également visualiser l'état des variables forcées dans l'éditeur de programmes.



#### IMPORTANT

Lors du forçage permanent d'une entrée ou d'une sortie dans une table de forçage permanent, les actions de forçage permanent deviennent partie intégrante de la configuration du projet. Si vous fermez STEP 7, les éléments forcés de manière permanente restent actifs dans le programme de la CPU, et ce jusqu'à ce qu'ils soient effacés. Pour annuler le forçage permanent de ces éléments, vous devez utiliser STEP 7 pour vous connecter à la CPU en ligne, puis utiliser la table de forçage permanent pour désactiver ou arrêter la fonction de forçage permanent pour ces éléments.

### 14.10.5.2 Fonctionnement de la fonction de forçage permanent

Vous pouvez effectuer un forçage permanent d'entrées et de sorties en indiquant l'adresse d'entrée ou de sortie physique (I\_:P ou Q\_:P) dans la table de forçage permanent puis en lançant la fonction de forçage permanent.

Dans le programme, les entrées physiques lues sont remplacées par les valeurs de forçage permanent. Le programme utilise les valeurs de forçage permanent pendant le traitement. Lorsque le programme écrit dans une sortie physique, la valeur de sortie est remplacée par la valeur de forçage permanent. La valeur de forçage permanent apparaît au niveau de la sortie physique et est utilisée par le processus.

14.10 Visualisation et forçage de valeurs dans la CPU

Lors du forçage permanent d'une entrée ou d'une sortie dans la table de forçage permanent, les actions de forçage deviennent partie intégrante du programme utilisateur. Même si vous fermez le logiciel de programmation, les sélections de forçage permanent restent actives dans le programme CPU qui fonctionne, jusqu'à ce que vous les annuliez en commutant en ligne avec le logiciel de programmation et en arrêtant la fonction de forçage permanent. Les programmes à entrées ou sorties forcées qui sont chargés dans une autre CPU à partir d'une carte mémoire continuent à appliquer le forçage permanent aux entrées et sorties sélectionnées dans le programme.

Si la CPU exécute le programme utilisateur à partir d'une carte mémoire protégée en écriture, vous ne pouvez pas lancer ou modifier le forçage permanent d'E/S à partir d'une table de visualisation, car il n'est pas possible d'écraser les valeurs dans le programme utilisateur protégé en écriture. Toute tentative de forçage permanent de valeurs protégées en écriture génère une erreur. Si vous utilisez une carte mémoire pour transférer un programme utilisateur, tout élément forcé en permanence sur cette carte mémoire sera transféré dans la CPU.

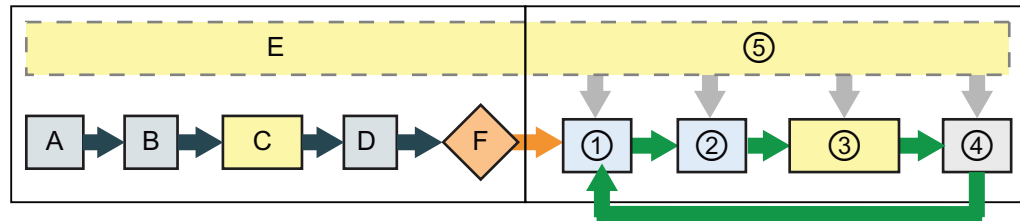
---

**Remarque**

**Les E/S TOR affectées à HSC, PWM et PTO ne peuvent pas être forcées.**

Les E/S TOR utilisées par le compteur rapide (HSC), la modulation de largeur d'impulsion (PWM) et la sortie de trains d'impulsions (PTO) sont affectées pendant la configuration des appareils. Lorsque des adresses d'E/S TOR sont affectées à ces appareils, les valeurs dans les adresses d'E/S affectées ne peuvent pas être forcées par la fonction de forçage permanent de la table de visualisation.

---

**Mise en route**

- A L'effacement de la zone de mémoire I n'est pas affecté par la fonction Forçage permanent.
- B L'initialisation des valeurs de sortie n'est pas affectée par la fonction Forçage permanent.
- C Pendant l'exécution des OB de démarrage, la CPU applique la valeur de forçage permanent lorsque le programme utilisateur accède à l'entrée physique.
- D L'enregistrement d'événements d'alarme dans la file d'attente n'est pas affecté.
- E La validation de l'écriture dans les sorties n'est pas affectée.

**MARCHÉ**

- ① Pendant l'écriture de la mémoire Q dans les sorties physiques, la CPU applique la valeur de forçage permanent lors de la mise à jour des sorties.
- ② Lors de la lecture des entrées physiques, la CPU applique les valeurs de forçage permanent juste avant de copier les entrées en mémoire I.
- ③ Pendant l'exécution du programme utilisateur (OB de cycle de programme), la CPU applique la valeur de forçage permanent lorsque le programme utilisateur accède à l'entrée physique ou écrit dans la sortie physique.
- ④ Le traitement des demandes de communications et le test d'auto-diagnostic ne sont pas affectés par la fonction Forçage permanent.
- ⑤ Le traitement des alarmes à n'importe quel moment du cycle n'est pas affecté.

## 14.11 Chargement dans la CPU à l'état MARCHE

La CPU prend en charge la fonction "Chargement dans la CPU à l'état MARCHE". Cette fonction doit vous permettre d'apporter de petites modifications au programme utilisateur en entraînant le moins de gêne possible pour le processus commandé par le programme. La réalisation de cette fonction permet toutefois des modifications importantes du programme, pouvant provoquer des perturbations ou même s'avérer dangereuses.

### ATTENTION

Lorsque vous chargez des modifications dans la CPU à l'état MARCHE, ces modifications affectent immédiatement le fonctionnement du processus. La modification du programme à l'état MARCHE peut entraîner un fonctionnement inattendu du système, pouvant causer la mort, des blessures graves ou des dégâts matériels importants.

Aussi, seules des personnes autorisées, conscientes des conséquences des modifications à l'état MARCHE sur le fonctionnement du système, sont-elles habilitées à effectuer un chargement à l'état MARCHE.

### 14.11 Chargement dans la CPU à l'état MARCHE

La fonction "Chargement à l'état MARCHE" permet de modifier un programme et de charger les modifications dans votre CPU sans faire passer cette dernière à l'état ARRET :

- Vous pouvez apporter des modifications mineures au processus en cours sans avoir à l'arrêter (par exemple, modifier la valeur d'un paramètre).
- Vous pouvez déboguer un programme plus rapidement grâce à cette fonction (par exemple, inverser la logique d'un contact normalement ouvert ou fermé).

Vous pouvez apporter les modifications suivantes aux variables et blocs de programme et les charger dans la CPU à l'état MARCHE :

- Créer, remplacer et effacer des fonctions (FC), blocs fonctionnels (FB) et tables de variables.
- Créer et effacer des blocs de données (DB). Toutefois, les modifications de structures de DB ne peuvent pas être écrasées. Les valeurs initiales des DB peuvent être écrasées. Vous ne pouvez pas charger un DB du serveur Web (commande ou fragment) à l'état MARCHE.
- Remplacer des blocs d'organisation (OB). Vous ne pouvez toutefois ni créer d'OB, ni en supprimer.

Dix blocs au maximum peuvent être chargés simultanément à l'état MARCHE. Pour charger plus de dix blocs, il faut mettre la CPU à l'état ARRET.

Si vous chargez des modifications dans un processus réel (et non dans un processus simulé comme cela pourrait être le cas au cours du débogage d'un programme), il est vital de considérer toutes les conséquences en termes de sécurité sur les machines et les opérateurs des machines avant le chargement.

---

#### Remarque

Si la CPU est à l'état MARCHE et que des modifications ont été apportées au programme, STEP 7 tentera toujours, dans un premier temps, d'effectuer un chargement dans la CPU à l'état MARCHE. Si ce n'est pas ce que vous désirez, vous devez faire passer la CPU à l'état ARRET.

Si les modifications effectuées ne sont pas acceptées en chargement à l'état MARCHE, STEP 7 demandera à l'utilisateur de mettre la CPU à l'état ARRET.

---

#### 14.11.1 Conditions requises pour un chargement dans la CPU à l'état MARCHE

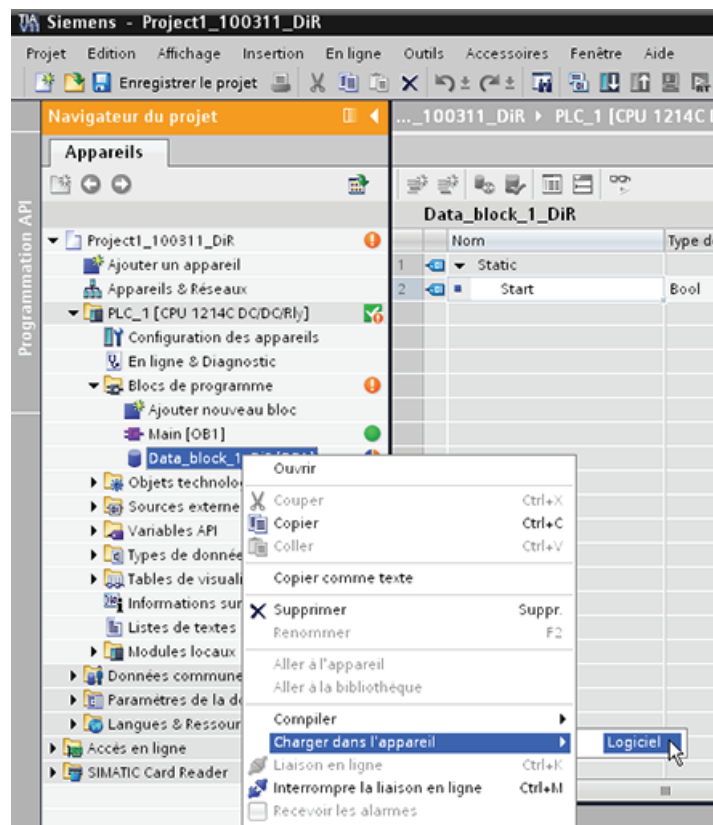
Vous ne pouvez pas charger des modifications de votre programme dans une CPU à l'état MARCHE si les conditions suivantes ne sont pas satisfaites :

- La compilation de votre programme ne doit pas présenter d'erreurs.
- Vous devez avoir établi une communication entre la console de programmation où s'exécute STEP 7 et la CPU.
- Dans la version V3.0 ou plus, le firmware de la CPU cible doit prendre en charge la fonction "Chargement dans la CPU à l'état MARCHE".

## 14.11.2 Modification du programme à l'état "Marché"

Pour modifier le programme à l'état MARCHÉ, vous devez d'abord vous assurer que la CPU accepte le "chargement à l'état MARCHÉ" et qu'elle est à l'état MARCHÉ.

1. Sélectionnez l'une des méthodes suivantes pour charger votre programme dans la CPU à l'état MARCHÉ :
  - Commande "Charger dans l'appareil" du menu "En ligne"
  - Bouton "Charger dans l'appareil" de la barre d'outils
  - Clic droit sur "Blocs de programme" dans le navigateur du projet et sélection de la commande "Charger dans l'appareil > Logiciel"

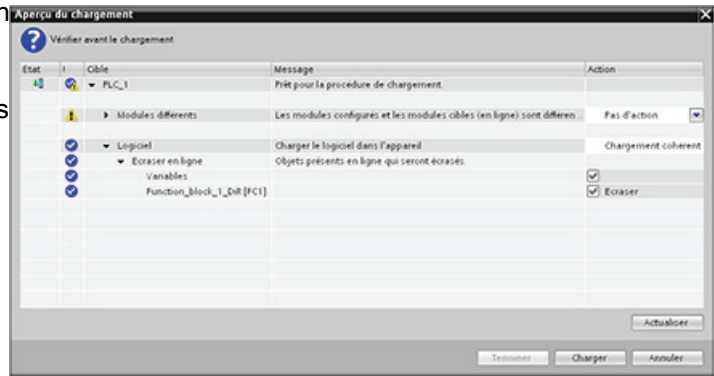


2. Si la compilation du programme a lieu sans erreurs, STEP 7 charge le programme dans la CPU.
3. STEP 7 vous demande de charger votre programme ou d'annuler l'opération.
4. Si vous cliquez sur "Charger", STEP 7 charge le programme dans la CPU.

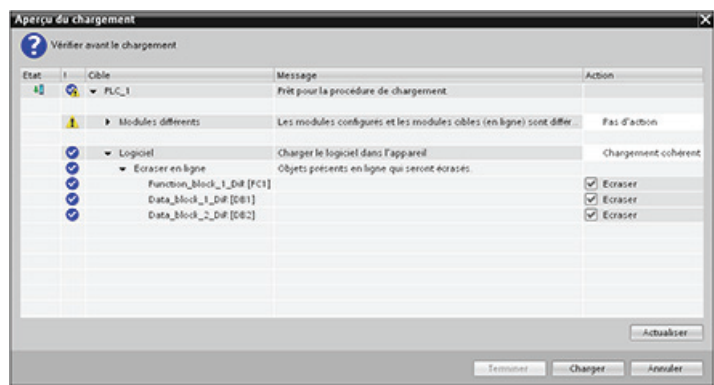
### 14.11.3 Chargement de blocs sélectionnés

On considère le dossier Blocs de programme, une sélection de blocs ou un bloc individuel.

1. Si l'utilisateur sélectionne un bloc individuel à charger de l'éditeur de bloc dans la CPU, la seule option disponible dans la colonne "Action" est "Chargement cohérent". L'utilisateur peut afficher le détail de la ligne de catégorie pour s'assurer des blocs à charger. Dans cet exemple, une petite modification a été apportée au bloc hors ligne et aucun autre bloc n'a besoin d'être chargé.



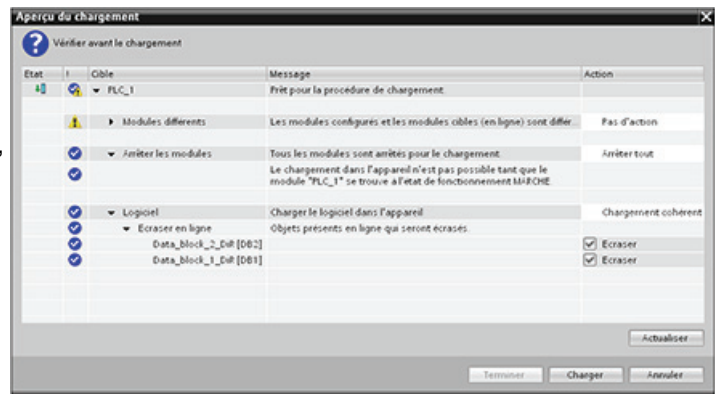
2. Dans cet exemple, il faut charger plus d'un bloc dans la CPU.



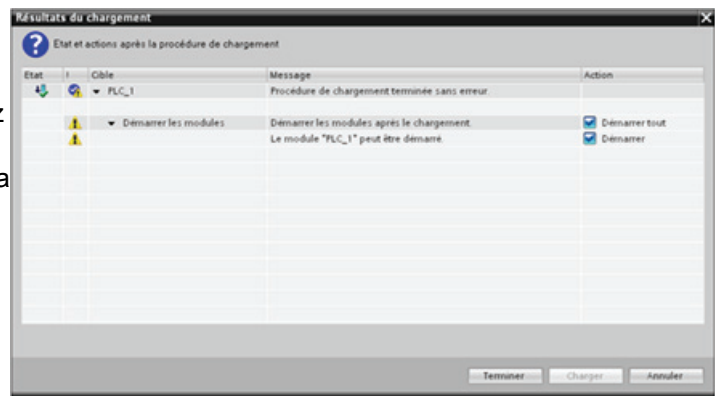
#### Remarque

Dix blocs au maximum peuvent être chargés simultanément à l'état MARCHE. Pour charger plus de dix blocs, il faut mettre la CPU à l'état ARRÊT.

3. Si l'utilisateur tente un chargement à l'état MARCHÉ mais que le système constate que ce n'est pas possible avant le chargement en cours, la ligne de catégorie Arrêt modules apparaît dans la boîte de dialogue.

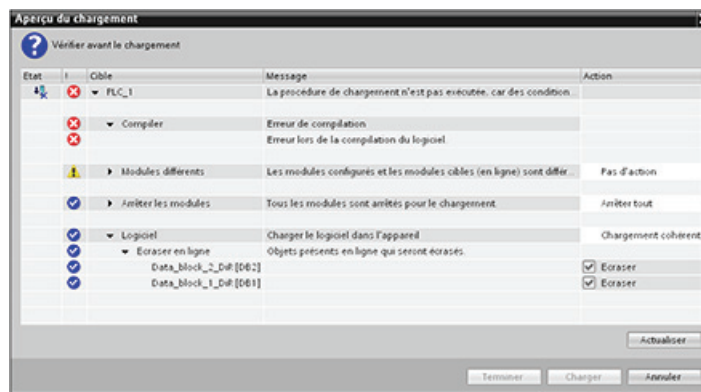


4. Cliquez sur le bouton "Charger" ; la boîte de dialogue "Résultats du chargement" s'affiche. Cliquez sur le bouton "Terminer" pour achever le chargement dans la CPU.



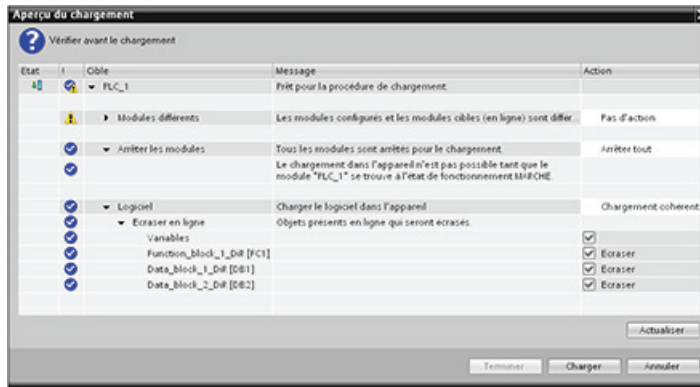
#### 14.11.4 Chargement d'un bloc individuel sélectionné alors qu'un autre bloc présente une erreur de compilation

Si l'utilisateur tente un chargement cohérent alors qu'une erreur de compilation existe dans un autre bloc, la boîte de dialogue signale une erreur et le bouton de chargement est désactivé.



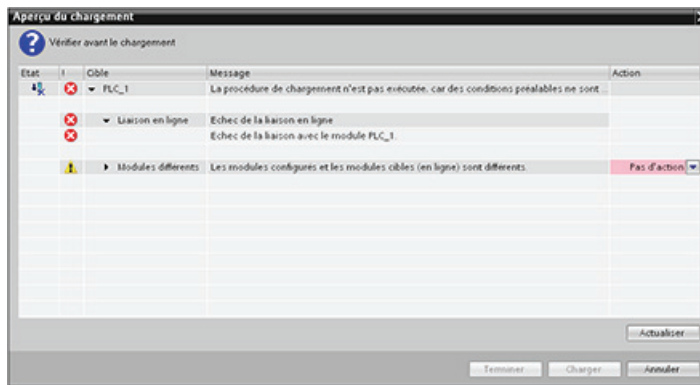
14.11 Chargement dans la CPU à l'état MARCHE

L'utilisateur doit corriger l'erreur de compilation dans l'autre bloc. Le bouton "Charger" devient alors actif.



14.11.5 Réaction du système en cas d'échec de l'opération de chargement

En cas de défaillance de la liaison réseau pendant le chargement initial à l'état MARCHE, la boîte de dialogue "Aperçu du chargement" s'affiche comme illustré dans la figure ci-dessous.





### 14.11.6 Chargement du programme dans la CPU à l'état MARCHE

Avant de charger le programme dans la CPU à l'état MARCHE, tenez compte des conséquences d'une modification à l'état MARCHE sur le fonctionnement de la CPU dans les situations suivantes :

- Si vous avez effacé la logique de commande d'une sortie, la CPU maintient le dernier état de cette sortie jusqu'à la mise hors tension puis sous tension suivante ou jusqu'au passage suivant à l'état ARRET.
- Si vous avez effacé un compteur rapide ou des fonctions de sortie d'impulsions alors qu'ils s'exécutaient, le compteur rapide ou la sortie d'impulsions continue à s'exécuter jusqu'à la mise hors tension puis sous tension suivante ou jusqu'au passage suivant à l'état ARRET.
- Toute logique dépendant de l'état du memento "Premier cycle" n'est pas exécutée avant la mise hors tension puis sous tension suivante ou le passage suivant de l'état ARRET à l'état MARCHE. Le memento "Premier cycle" est mis à 1 uniquement par le passage à l'état MARCHE et n'est pas affecté par un chargement à l'état MARCHE.
- Les valeurs en cours des blocs de données et/ou variables ne seront pas écrasées.

---

#### Remarque

Pour que vous puissiez charger votre programme à l'état MARCHE, la CPU doit accepter les modifications à l'état MARCHE, le programme doit être compilé sans erreurs et la communication entre STEP 7 et la CPU doit fonctionner correctement.

Vous pouvez apporter les modifications suivantes aux variables et blocs de programme et les charger dans la CPU à l'état MARCHE :

- Créer, remplacer et effacer des fonctions (FC), blocs fonctionnels (FB) et tables de variables.
- Créer et effacer des blocs de données (DB). Toutefois, les modifications de structures de DB ne peuvent pas être écrasées. Les valeurs initiales des DB peuvent être écrasées. Vous ne pouvez pas charger un DB du serveur Web (commande ou fragment) à l'état MARCHE.
- Remplacer des blocs d'organisation (OB). Vous ne pouvez toutefois ni créer d'OB, ni en supprimer.

Dix blocs au maximum peuvent être chargés simultanément à l'état MARCHE. Pour charger plus de dix blocs, il faut mettre la CPU à l'état ARRET.

Une fois qu'un chargement a été déclenché, vous ne pouvez pas exécuter d'autres tâches dans STEP 7 tant que ce chargement n'est pas achevé.

---

### Instructions pouvant échouer en raison d'un chargement à l'état MARCHE

Les instructions suivantes peuvent présenter une erreur temporaire lorsque des modifications chargées à l'état MARCHE sont activées dans la CPU. L'erreur se produit lorsque l'instruction est déclenchée alors que la CPU se prépare à activer les modifications chargées. Pendant ce temps, la CPU suspend toute demande d'accès du programme utilisateur à la mémoire de chargement alors qu'elle achève l'accès en cours du programme utilisateur à la mémoire de chargement. Ceci afin que les modifications chargées puissent être activées de manière cohérente.

Instruction	Réponse pendant que l'activation est en attente
DataLogCreate	STATUS = W#16#80C0, ERROR = TRUE
DataLogOpen	STATUS = W#16#80C0, ERROR = TRUE
DataLogWrite	STATUS = W#16#80C0, ERROR = TRUE
DataLogClose	STATUS = W#16#80C0, ERROR = TRUE
DataLogNewFile	STATUS = W#16#80C0, ERROR = TRUE
READ_DBL	RET_VAL = W#16#82C0
WRIT_DBL	RET_VAL = W#16#82C0
RTM	RET_VAL = 0x80C0

Dans tous les cas, la sortie RLO de l'instruction a la valeur FAUX lorsque l'erreur se produit. Cette erreur est temporaire. Si elle se produit, il faut retenter l'instruction plus tard.

#### Remarque

Il ne faut pas faire de nouvelle tentative pendant l'exécution en cours de l'OB.

# Caractéristiques techniques

# A

## A.1 Caractéristiques techniques d'ordre général

### Conformité aux normes

La conception du système d'automatisation S7-1200 est conforme aux normes et spécifications d'essai suivantes. Les critères de test pour le système d'automatisation S7-1200 se fondent sur ces normes et spécifications d'essai.

Notez que les modèles S7-1200 ne sont peut-être pas tous certifiés pour ces normes et que l'état d'homologation peut changer sans avis. Il est de la responsabilité de l'utilisateur de déterminer les homologations applicables en se reportant aux marquages figurant sur le produit. Veuillez contacter votre agence Siemens si vous avez besoin d'informations supplémentaires concernant la dernière liste d'homologations exactes par numéro de référence.

### Homologation CE



Le système d'automatisation S7-1200 satisfait aux exigences et objectifs en matière de sécurité des directives CE énumérées ci-dessous et est conforme aux normes européennes harmonisées (EN) pour les automates programmables énumérées dans les Journaux officiels de l'Union Européenne.

- Directive CE 2006/95/CE (Basse tension) "Matériel électrique destiné à être employé dans certaines limites de tension"
  - EN 61131-2:2007 Automates programmables Spécifications et essais des équipements
- Directive CE 2004/108/CE (directive CEM) "Compatibilité électromagnétique"
  - Norme sur les émissions  
EN 61000-6-4:2007 : environnement industriel
  - Norme sur l'immunité  
EN 61000-6-2:2005 : environnement industriel
- Directive CE 94/9/CE (ATEX) "Appareils et systèmes de protection destinés à être utilisés en atmosphères explosibles"
  - EN 60079-15:2005 : Classe de protection "n"

La déclaration de conformité CE est tenue à disposition des autorités compétentes auprès de :

Siemens AG  
IA AS RD ST PLC Amberg  
Werner-von-Siemens-Str. 50  
D92224 Amberg  
Allemagne

### Homologation cULus



Underwriters Laboratories Inc. conformément à :

- Underwriters Laboratories, Inc. : listé UL 508 (Industrial Control Equipment)
- Canadian Standards Association : CSA C22.2 numéro 142 (Process Control Equipment)

#### IMPORTANT

La gamme SIMATIC S7-1200 satisfait à la norme CSA.

La marque cULus indique que le S71200 a été examiné et certifié par les laboratoires Underwriters Laboratories (UL) comme étant conforme aux normes UL 508 et CSA 22.2 numéro 142.

### Homologation FM



Factory Mutual Research (FM)

Norme d'homologation classe numéro 3600 et 3611

Homologué pour utilisation dans :

Classe I, Division 2, Groupe gaz A, B, C, D, Classe de température T3C Ta = 60 °C

Classe I, Zone 2, IIC, Classe de température T3 Ta = 60 °C

Classe canadienne I, Installation zone 2 par CEC 18-150

EXCEPTION IMPORTANTE : Voir les caractéristiques techniques pour le nombre d'entrées ou de sorties pouvant être à 1 simultanément. Certains modèles sont déclassés pour Ta = 60 °C.

#### ! ATTENTION

Le remplacement de composants peut remettre en cause la conformité à la classe I, division 2 et zone 2.

Les réparations ne doivent être effectuées que par un centre de service Siemens agréé.

## Homologation ATEX



L'homologation ATEX ne s'applique qu'aux modèles à courant continu. Elle ne s'applique pas aux modèles à courant alternatif et relais.

EN 60079-0:2006 : Atmosphères explosives - Règles générales

EN 60079-15:2005 : Matériel électrique pour atmosphères explosives ;

Type de protection 'nA'

II 3 G Ex nA II T3

**EXCEPTION IMPORTANTE** : Voir les caractéristiques techniques pour le nombre d'entrées ou de sorties pouvant être à 1 simultanément. Certains modèles sont déclassés pour Ta = 60 °C.

Installez les modules dans une enceinte appropriée fournissant un degré de protection IP54 minimum selon EN 60529 et tenez compte des conditions ambiantes dans lesquelles les appareils seront utilisés.

Lorsque la température dans les conditions nominales dépasse 70 °C au point d'entrée du câble ou 80 °C au point de branchement des conducteurs, la spécification de température du câble sélectionné doit être conforme à la température mesurée réelle.

Il faut prendre des mesures pour empêcher la tension nominale d'être dépassée par des perturbations transitoires de plus de 40%.

## Homologation C-Tick



Le système d'automatisation S7-1200 satisfait aux exigences de la norme AS/NZS 2064 (classe A).

## Certification coréenne



Le système d'automatisation S7-1200 satisfait aux exigences de l'homologation coréenne (marque KC). Il a été défini comme équipement de classe A et est conçu pour des applications industrielles et non pour un usage domestique.

### Homologation pour le domaine maritime

Les produits S7-1200 sont régulièrement soumis à des homologations d'agences spéciales pour des marchés et des applications spécifiques. Veuillez contacter votre agence Siemens si vous avez besoin d'informations supplémentaires concernant la dernière liste d'homologations exactes par numéro de référence.

Sociétés de classification :

- ABS (American Bureau of Shipping)
- BV (Bureau Veritas)
- DNV (Det Norske Veritas)
- GL (Germanischer Lloyd)
- LRS (Lloyds Register of Shipping)
- Class NK (Nippon Kaiji Kyokai)

### Environnements industriels

Le système d'automatisation S7-1200 est conçu pour être utilisé dans des environnements industriels.

Tableau A- 1 Environnements industriels

Champ d'application	Exigences concernant l'émission de bruits	Exigences concernant l'immunité aux bruits
Industriel	EN 61000-6-4:2007	EN 61000-6-2:2005

### Compatibilité électromagnétique

La compatibilité électromagnétique (CEM) est la capacité d'un appareil électrique à fonctionner comme prévu dans un environnement électromagnétique et à fonctionner sans émettre des niveaux d'interférence électromagnétique pouvant perturber d'autres appareils électriques à proximité.

Tableau A- 2 Immunité selon EN 61000-6-2

Compatibilité électromagnétique - Immunité selon EN 61000-6-2	
EN 61000-4-2 Décharge électrostatique	8 kV : décharge dans l'air vers toutes les surfaces 6 kV : décharge au contact vers les surfaces conductrices exposées
EN 61000-4-3 Essai d'immunité aux champs électromagnétiques rayonnés aux fréquences radioélectriques	80 à 1000 MHz, 10 V/m, 80% de modulation d'amplitude à 1 kHz 1,4 à 2,0 GHz, 3 V/m, 80% de modulation d'amplitude à 1 kHz 2,0 à 2,7 GHz, 1 V/m, 80% de modulation d'amplitude à 1 kHz
EN 61000-4-4 Salves transitoires rapides	2 kV, 5 kHz avec réseau de couplage vers système CC et CA 2 kV, 5 kHz avec blocage de couplage vers E/S
EN 6100-4-5 Immunité aux pointes de tension	Systèmes CA : 2 kV mode commun, 1 kV mode différentiel Systèmes CC : 2 kV mode commun, 1 kV mode différentiel Une protection externe est requise pour les systèmes CC (signaux E/S, systèmes d'alimentation CC).

<b>Compatibilité électromagnétique - Immunité selon EN 61000-6-2</b>	
EN 61000-4-6 Perturbations par conduction	150 kHz à 80 MHz, 10 V eff., 80 % de modulation d'amplitude à 1 kHz
EN 61000-4-11 Baisses de tension	Systèmes CA 0% pour 1 cycle, 40% pour 12 cycles et 70% pour 30 cycles à 60 Hz

Tableau A- 3 Emissions conduites et rayonnées selon EN 61000-6-4

<b>Compatibilité électromagnétique - Emissions conduites et rayonnées selon EN 61000-6-4</b>		
Emissions conduites EN 55011, classe A, groupe 1	0,15 MHz à 0,5 MHz	<79dB (µV) quasi-pointe ; <66 dB (µV) moyenne
	0,5 MHz à 5 MHz	<73dB (µV) quasi-pointe ; <60 dB (µV) moyenne
	5 MHz à 30 MHz	<73dB (µV) quasi-pointe ; <60 dB (µV) moyenne
Emissions rayonnées EN 55011, classe A, groupe 1	30 MHz à 230 MHz	<40dB (µV/m) quasi-pointe, mesuré à 10 m
	230 MHz à 1 GHz	<47dB (µV/m) quasi-pointe, mesuré à 10 m

## Conditions ambiantes

Tableau A- 4 Transport et stockage

<b>Conditions ambiantes - Transport et stockage</b>	
EN 6006822, test Bb, chaleur sèche EN 6006821, test Ab, froid	-40 °C à +70 °C
EN 60068-2-30, test Db, chaleur humide saturée	25 °C à 55 °C, 95 % d'humidité
EN 60068-2-14, test Na, choc de température	40 °C à +70 °C, temps de maintien 3 heures, 5 cycles
EN 60068-2-32, chute libre	0,3 m, 5 fois, emballage du produit
Pression atmosphérique	1080 à 660 hPa (correspond à une altitude de -1000 à 3500 m)

Tableau A- 5 Conditions de service

<b>Conditions ambiantes - Fonctionnement</b>	
Plage de température ambiante (admission d'air 25 mm en dessous de l'unité)	-20 °C à 60 °C en montage horizontal -20 °C à 50 °C en montage vertical 95% d'humidité sans condensation Sauf indication contraire
Pression atmosphérique	1080 à 795 hPa (correspond à une altitude de -1000 à 2000 m)
Concentration de contaminants	SO <sub>2</sub> : < 0,5 ppm ; H <sub>2</sub> S : < 0,1 ppm ; RH < 60 % sans condensation
EN 60068-2-14, test Nb, changement de température	5 °C à 55 °C, 3 °C/minute
EN 60068-2-27 Choc mécanique	15 G, impulsion de 11 ms, 6 chocs dans chacun des 3 axes
EN 60068-2-6 Vibrations sinusoïdales	Montage sur profilé support : 3,5 mm de 5 à 9 Hz, 1G de 9 à 150 Hz Montage sur panneau : 7,0 mm de 5 à 9 Hz, 2G de 9 à 150 Hz 10 balayages par axe, 1 octave/minute

**IMPORTANT**

Pour les systèmes devant démarrer dans la plage de -20 °C à 0 °C, il faut que le programme utilisateur retarde la mise sous tension des sorties de 10 secondes après le démarrage.

Tableau A- 6 Essai d'isolation à potentiel élevé

Essai d'isolation à potentiel élevé	
Circuits 24 V/5 V nominal	520 V- (essai de type de barrières d'isolation optique)
Circuits 115/230 V à la terre	1500 V~
Circuits 115/230 V à circuits 115/230 V	1500 V~
Circuits 115 V/230 V à circuits 24 V/5 V	1500 V~ (3000 V~ / 4242 V- essai de type)
Port Ethernet à circuits 24 V/5 V et terre <sup>1</sup>	1500 V~ (essai de type uniquement)

<sup>1</sup> L'isolation du port Ethernet est conçue pour limiter les risques pendant les pannes de réseau de courte durée avec tensions dangereuses. Elle ne satisfait pas aux exigences en matière de sécurité pour l'isolation des lignes CA de routine.

**Classe de protection**

- Classe de protection II selon EN 61131-2 (un conducteur de protection n'est pas requis)

**Degré de protection**

- IP20 Protection mécanique, EN 60529
- Protection contre le contact des doigts avec la haute tension comme testé par sonde standard. Une protection externe est nécessaire contre la poussière, la saleté, l'eau et les objets étrangers de diamètre inférieur à 12,5 mm.

**Tensions nominales**

Tableau A- 7 Tensions nominales

Tension nominale	Tolérance
24 V-	20,4 V- à 28,8 V- 22,0 V- à 28,8 V à une température ambiante de -20 °C à 0 °C
120/230 V~	85 V~ à 264 V~, 47 à 63 Hz



**IMPORTANT**

Lorsqu'un contact mécanique active un courant de sortie vers la CPU S7-1200 ou un module d'entrées-sorties TOR quelconque, il envoie un signal "1" aux sorties TOR pendant environ 50 microsecondes. Cela peut provoquer un fonctionnement inattendu des machines ou du processus, pouvant entraîner la mort, des blessures graves et des dommages matériels importants. Vous devez tenir compte de ce fait, notamment si vous utilisez des unités qui réagissent à des impulsions de courte durée.

**Protection contre la tension inverse**

Un circuit de protection contre la tension inverse est fourni sur chaque paire de bornes de l'alimentation +24 V- ou de l'alimentation d'entrée utilisateur pour les CPU, les modules d'entrées-sorties (SM) et les Signal Boards (SB). Endommager le système reste possible si l'on câble des paires de bornes différentes dans des polarités inverses.

Certains ports d'entrée d'alimentation 24 V- dans le système S7-1200 sont interconnectés, avec un circuit logique commun connectant plusieurs bornes M. Par exemple, les circuits suivants sont interconnectés lorsqu'ils sont signalés comme "non isolés" dans les fiches techniques : l'alimentation 24 V- de la CPU, l'entrée d'alimentation pour la bobine de relais d'un SM ou l'alimentation pour une entrée analogique non isolée. Toutes les bornes M non isolées doivent être connectées au même potentiel de référence externe.

 **ATTENTION**

Connecter des bornes M non isolées à des potentiels de référence différents provoque des flux de courant indésirables qui peuvent être à l'origine de dégâts ou d'un fonctionnement imprévisible dans l'automate et tout équipement connecté.

Le non-respect de ces conseils peut être à l'origine de dégâts ou d'un fonctionnement imprévisible pouvant entraîner la mort, des blessures graves et des dommages matériels importants.

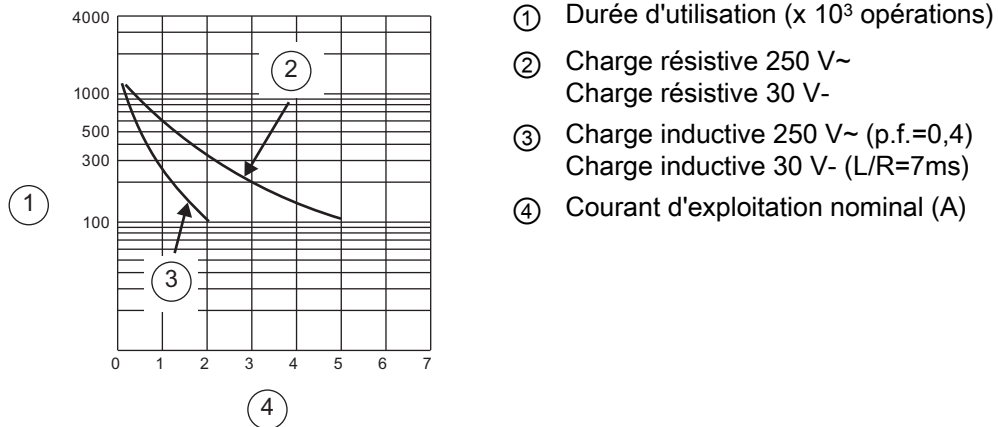
Vérifiez toujours que toutes les bornes M non isolées dans un système S7-1200 sont connectées au même potentiel de référence.

**Sorties CC**

Aucun circuit de protection contre les courts-circuits n'est fourni pour les sorties CC des CPU, modules d'entrées-sorties (SM) et Signal Boards (SB).

**Durée d'utilisation électrique des relais**

La figure ci-dessous présente les performances typiques données par les fournisseurs de relais. Les performances effectives peuvent varier selon votre application spécifique. Un circuit de protection externe adapté à la charge allongera la durée d'utilisation des contacts.



**A.2 CPU 1211C**

**A.2.1 Caractéristiques et fonctions générales**

Tableau A- 8 Caractéristiques générales

Caractéristiques techniques	CPU 1211C AC/DC/Relais	CPU 1211C DC/DC/Relais	CPU 1211C DC/DC/DC
Numéro de référence	6ES7 211-1BE31-0XB0	6ES7 211-1HE31-0XB0	6ES7 211-1AE31-0XB0
Dimensions L x H x P (mm)	90 x 100 x 75	90 x 100 x 75	90 x 100 x 75
Poids avec emballage	420 grammes	380 grammes	370 grammes
Dissipation de courant	10 W	8 W	8 W
Courant disponible (bus CM)	750 mA max. (5 V-)	750 mA max. (5 V-)	750 mA max. (5 V-)
Courant disponible (24 V-)	300 mA max. (alimentation de capteur)	300 mA max. (alimentation de capteur)	300 mA max. (alimentation de capteur)
Consommation de courant entrées TOR (24 V-)	4 mA / entrée utilisée	4 mA / entrée utilisée	4 mA / entrée utilisée

Tableau A- 9 Fonctions de la CPU

Caractéristiques techniques		Description
Mémoire utilisateur <sup>1</sup>	de travail	30 Ko
	de chargement	1 Mo interne, extensible jusqu'à la taille de la carte SD
	rémanente	10 Ko
E/S TOR intégrées		6 entrées/4 sorties
E/S analogiques intégrées		2 entrées
Taille de la mémoire image		1024 octets d'entrées (I) / 1024 octets de sorties (Q)
Mémentos (M)		4096 octets
Mémoire temporaire (locale)		<ul style="list-style-type: none"> <li>• 16 Ko pour le démarrage et le cycle du programme (FB et FC associés inclus)</li> <li>• 4 Ko pour les événements d'alarme standard, FB et FC inclus</li> <li>• 4 Ko pour les événements d'alarme d'erreur, FB et FC inclus</li> </ul>
Modules d'entrées-sorties pour extension		aucun
SB, CB, BB pour extension		1 max.
Modules de communication pour extension		3 CM max.
Compteurs rapides		3 intégrés, 5 avec Signal Board, voir le tableau Affectations des entrées HSC pour la CPU 1211C (Page 361) <ul style="list-style-type: none"> <li>• Monophasé : 3 à 100 kHz, SB : 2 à 30 kHz</li> <li>• Quadrature de phase : 3 à 80 kHz, SB : 2 à 20 kHz</li> </ul>
Sorties d'impulsions <sup>2</sup>		4
Entrées de capture d'impulsions		6
Alarmes temporisées / cycliques		4 au total avec résolution de 1 ms
Alarmes sur front		6 sur front montant et 6 sur front descendant (10 et 10 avec Signal Board optionnel)
Carte mémoire		Carte mémoire SIMATIC (facultative)
Précision de l'horloge temps réel		+/- 60 secondes/mois
Durée de conservation de l'horloge temps réel		20 jours typ./12 jours min. à 40°C (supercondensateur sans maintenance)

<sup>1</sup> La taille du programme utilisateur, des données et de la configuration est limitée par la mémoire de chargement et la mémoire de travail disponibles dans la CPU. Il n'y a pas de limite spécifique pour le nombre de blocs OB, FC, FB et DB pris en charge ni pour la taille d'un bloc particulier. La seule limitation est la taille de mémoire globale.

<sup>2</sup> Pour les modèles de CPU avec sorties relais, vous devez installer un Signal Board (SB) TOR pour utiliser les sorties d'impulsions.

Tableau A- 10 Performances

Type d'instruction	Vitesse d'exécution
Logique booléenne	0,08 µs/instruction
Copier mot	1,7 µs/instruction
Mathématiques sur réels	2,3 µs/instruction

## A.2.2 Temporisations, compteurs et blocs de code pris en charge par la CPU 1211C

Tableau A- 11Blocs, temporisations et compteurs pris en charge par la CPU 1211C

Élément	Description	
Blocs	Type	OB, FB, FC, DB
	Taille	30 Ko
	Quantité	Jusqu'à 1024 blocs au total (OB + FB + FC + DB)
	Plage d'adresses pour les FB, FC et DB	1 à 65535 (FB 1 à FB 65535, par exemple)
	Profondeur d'imbrication	16 à partir de l'OB de cycle de programme ou de démarrage ; 4 à partir de l'OB d'alarme temporisée, d'alarme horaire, d'alarme cyclique, d'alarme de processus, d'erreur de temps ou d'alarme de diagnostic
Visualisation	Il est possible de visualiser simultanément l'état de 2 blocs de code.	
OB	Cycle de programme	Plusieurs : OB 1, OB 200 à OB 65535
	Mise en route	Plusieurs : OB 100, OB 200 à OB 65535
	Alarmes temporisées et alarmes cycliques	4 <sup>1</sup> (1 par événement) : OB 200 à OB 65535
	Alarmes de processus (fronts et HSC)	50 (1 par événement) : OB 200 à OB 65535
	Alarmes d'erreur de temps	1: OB 80
Alarmes de diagnostic	1: OB 82	
Temporisations	Type	CEI
	Quantité	Limitée uniquement par la taille de la mémoire
	Stockage	Structure dans un DB, 16 octets par temporisation
Compteurs	Type	CEI
	Quantité	Limitée uniquement par la taille de la mémoire
	Stockage	Structure dans un DB, la taille dépend du type de compteur <ul style="list-style-type: none"> <li>• SInt, USInt : 3 octets</li> <li>• Int, UInt : 6 octets</li> <li>• DInt, UDInt : 12 octets</li> </ul>

<sup>1</sup> Les alarmes temporisées et cycliques utilisent les mêmes ressources dans la CPU. Vous pouvez avoir uniquement 4 de ces alarmes au total (temporisées et cycliques). Vous ne pouvez pas avoir 4 alarmes temporisées et 4 alarmes cycliques.

Tableau A- 12Communication

Caractéristiques techniques	Description
Nombre de ports	1
Type	Ethernet
Appareil IHM <sup>1</sup>	3
Console de programmation (PG)	1

Caractéristiques techniques	Description
Liaisons	<ul style="list-style-type: none"> <li>• 8 pour la communication ouverte (Open User Communication) active ou passive : TSEND_C, TRCV_C, TCON, TDISCON, TSEND et TRCV</li> <li>• 3 pour la communication S7 (CPU à CPU) GET/PUT serveur</li> <li>• 8 pour la communication S7 (CPU à CPU) GET/PUT client</li> </ul>
Débits	10/100 Mb/s
Isolation (signal externe à logique API)	Isolé par transformateur, 1500 V-, pour une protection contre les événements de courte durée uniquement
Type de câble	CAT5e blindé

- <sup>1</sup> La CPU fournit des liaisons IHM spécialisées pour la prise en charge de 3 appareils IHM au maximum (vous pouvez avoir 2 SIMATIC Comfort Panels au maximum). Le nombre total de stations IHM dépend des types de pupitres IHM dans votre configuration. Vous pouvez, par exemple, avoir trois SIMATIC Basic Panels au plus connectés à votre CPU ou bien deux SIMATIC Comfort Panels au plus avec un Basic Panel additionnel.

Tableau A- 13Alimentation

Caractéristiques techniques		CPU 1211C AC/DC/Relais	CPU 1211C DC/DC/Relais	CPU 1211C DC/DC/DC
Plage de tension		85 à 264 V~	20,4 V- à 28,8 V- 22,0 V- à 28,8 V- à une température ambiante de -20 °C à 0 °C	
Fréquence de ligne		47 à 63 Hz	--	--
Courant d'entrée	CPU seulement à la charge max.	60 mA à 120 V~ 30 mA à 240 V~	300 mA à 24 V-	300 mA à 24 V-
	CPU avec tous les accessoires d'extension à la charge max.	180 mA à 120 V~ 90 mA à 240 V~	900 mA à 24 V-	900 mA à 24 V-
Appel de courant (max.)		20 A à 264 V~	12 A à 28,8 V-	12 A à 28,8 V-
Isolation (alimentation d'entrée à logique)		1500 V~	Non isolée	Non isolée
Fuite à la terre, ligne CA à la terre fonctionnelle		0,5 mA max.	--	--
Temps de retard (perte d'alimentation)		20 ms à 120 V~ 80 ms à 240 V~	10 ms à 24 V-	10 ms à 24 V-
Fusible interne, non remplaçable par l'utilisateur		3 A, 250 V, action retardée	3 A, 250 V, action retardée	3 A, 250 V, action retardée

Tableau A- 14Alimentation de capteur

Caractéristiques techniques	CPU 1211C AC/DC/Relais	CPU 1211C DC/DC/Relais	CPU 1211C DC/DC/DC
Plage de tension	20,4 à 28,8 V-	L+ moins 4 V- min. L+ moins 5 V- min. à une température ambiante de -20 °C à 0 °C	
Courant de sortie nominal (max.)	300 mA (protégé contre les courts-circuits)	300 mA (protégé contre les courts-circuits)	300 mA (protégé contre les courts-circuits)
Bruit d'ondulation maximum (<10 MHz)	< 1 V crête à crête	Comme la ligne d'entrée	Comme la ligne d'entrée
Isolation (logique CPU à alimentation de capteur)	Non isolée	Non isolée	Non isolée

### A.2.3 Entrées et sorties TOR

Tableau A- 15Entrées TOR

Caractéristiques techniques	CPU 1211C AC/DC/Relais, CPU 1211C DC/DC/Relais et CPU 1211C DC/DC/DC
Nombre d'entrées	6
Type	P/N (CEI type 1 en mode P)
Tension nominale	24 V- à 4 mA, nominal
Tension continue admise	30 V-, max.
Tension de choc	35 V- pour 0,5 s
Signal 1 logique (min.)	15 V- à 2,5 mA
Signal 0 logique (max.)	5 V- à 1 mA
Isolation (site à logique)	500 V~ pour 1 minute
Groupes d'isolation	1
Temps de filtre	0,2, 0,4, 0,8, 1,6, 3,2, 6,4 et 12,8 ms (sélectionnables par groupes de 4)
Fréquences d'entrée d'horloge HSC (max.) (niveau 1 logique = 15 à 26 V-)	Monophasé : 100 kHz Quadrature de phase : 80 kHz
Nombre d'entrées simultanément à 1	6 à 60 °C horizontal, 50 °C vertical
Longueur de câble (mètres)	500 m blindé, 300 m non blindé, 50 m blindé pour entrées HSC

Tableau A- 16Sorties TOR

Caractéristiques techniques	CPU 1211C AC/DC/Relais et CPU 1211C DC/DC/Relais	CPU 1211C DC/DC/DC
Nombre de sorties	4	4
Type	Relais, contact sec	Transistor à effet de champ MOS (mode N)
Plage de tension	5 à 30 V- ou 5 à 250 V~	20,4 à 28,8 V-
Signal 1 logique à courant max.	--	20 V- min.
Signal 0 logique avec charge 10 kΩ	--	0,1 V- max.
Courant (max.)	2,0 A	0,5 A
Charge de lampe	30 W CC / 200 W CA	5 W
Résistance état activé	0,2 Ω max. lorsque neuf	0,6 Ω max.
Courant de fuite par sortie	--	10 µA max.
Courant de choc	7 A avec contacts fermés	8 A pour 100 ms max.
Protection contre la surcharge	Non	Non
Isolation (site à logique)	1500 V~ pour 1 minute (bobine à contact) Aucune (bobine à logique)	500 V~ pour 1 minute
Résistance d'isolation	100 MΩ min. lorsque neuf	--
Isolation entre contacts ouverts	750 V~ pour 1 minute	--
Groupes d'isolation	1	1
Tension de blocage inductive	--	L+ moins 48 V-, dissipation 1 W
Fréquence de commutation maximum des relais	1 Hz	--
Retard de commutation (Qa.0 à Qa.3)	10 ms max.	1,0 µs max., de 0 à 1 3,0 µs max., de 1 à 0
Fréquence de sortie de trains d'impulsions (Qa.0 et Qa.2)	Non recommandé <sup>1</sup>	100 kHz max., 2 Hz min. <sup>2</sup>
Durée d'utilisation mécanique (sans charge)	10 000 000 cycles ouvert/fermé	--
Durée d'utilisation des contacts à la charge nominale	100 000 cycles ouvert/fermé	--
Comportement au passage de MARCHE à ARRÊT	Dernière valeur ou valeur de remplacement (valeur par défaut 0)	Dernière valeur ou valeur de remplacement (valeur par défaut 0)
Nombre de sorties simultanément à 1	4 à 60 °C horizontal, 50 °C vertical	
Longueur de câble (mètres)	500 m blindé, 150 m non blindé	500 m blindé, 150 m non blindé

<sup>1</sup> Pour les modèles de CPU avec sorties relais, vous devez installer un Signal Board (SB) TOR pour utiliser les sorties d'impulsions.

<sup>2</sup> Selon vos câble et récepteur d'impulsions, une résistance de charge supplémentaire (au moins 10% du courant nominal) peut améliorer la qualité du signal d'impulsion et l'immunité aux bruits.

## A.2.4 Entrées analogiques

Tableau A- 17 Entrées analogiques

Caractéristiques techniques	Description
Nombre d'entrées	2
Type	Tension (mode simple)
Plage pleine échelle	0 à 10 V
Plage pleine échelle (mot de données)	0 à 27648
Plage de dépassement haut	10,001 à 11,759 V
Plage de dépassement (mot de données)	27 649 à 32 511
Plage de débordement haut	11,760 à 11,852 V
Plage de débordement haut (mot de données)	32 512 à 32 767
Résolution	10 bits
Tension de tenue maximum	35 V-
Lissage	Aucun, faible, moyen ou fort Voir le tableau pour la réponse indicielle (ms) des entrées analogiques de la CPU (Page 760).
Réjection des bruits	10, 50 ou 60 Hz
Impédance	≥100 kΩ
Isolation (site à logique)	Aucune
Précision (25°C / -20 à 60°C)	3,0% / 3,5% de la pleine échelle
Longueur de câble (mètres)	100 m, paire torsadée blindée

### A.2.4.1 Réponse indicielle des entrées analogiques intégrées de la CPU

Tableau A- 18 Réponse indicielle (ms), 0 V à 10 V mesuré à 95%

Sélection de lissage (moyennage d'échantillon)	Fréquence de réjection (temps d'intégration)		
	60 Hz	50 Hz	10 Hz
Aucun (1 cycle) : Pas de moyennage	50 ms	50 ms	100 ms
Faible (4 cycles) : 4 échantillons	60 ms	70 ms	200 ms
Moyen (16 cycles) : 16 échantillons	200 ms	240 ms	1150 ms
Fort (32 cycles) : 32 échantillons	400 ms	480 ms	2300 ms
<b>Temps d'échantillonnage</b>	<b>4,17 ms</b>	<b>5 ms</b>	<b>25 ms</b>



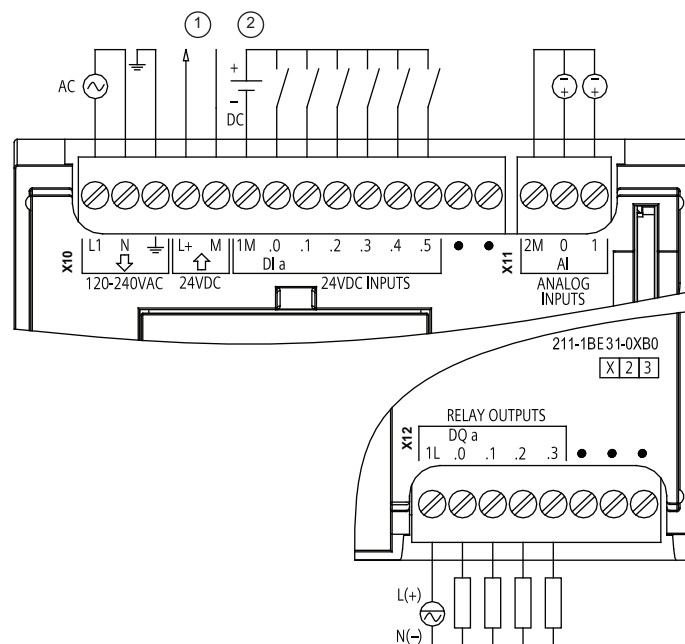
### A.2.4.2 Temps d'échantillonnage pour les ports analogiques intégrés de la CPU

Tableau A- 19 Temps d'échantillonnage pour les entrées analogiques intégrées de la CPU

Fréquence de réjection (sélection de temps d'intégration)	Temps d'échantillonnage
60 Hz (16,6 ms)	4,17 ms
50 Hz (20 ms)	5 ms
10 Hz (100 ms)	25 ms

### A.2.5 Schémas de câblage de la CPU 1211C

Tableau A- 20 CPU 1211C AC/DC/Relais (6ES7 211-1BE31-0XB0)



- ① Sortie alim. capteur 24 V-  
Pour plus d'immunité aux bruits, connectez "M" à la terre du châssis même si vous n'utilisez pas l'alimentation de capteur.
- ② Pour les entrées de type P, connectez "-" à "M" (comme illustré).  
Pour les entrées de type N, connectez "+" à "M".

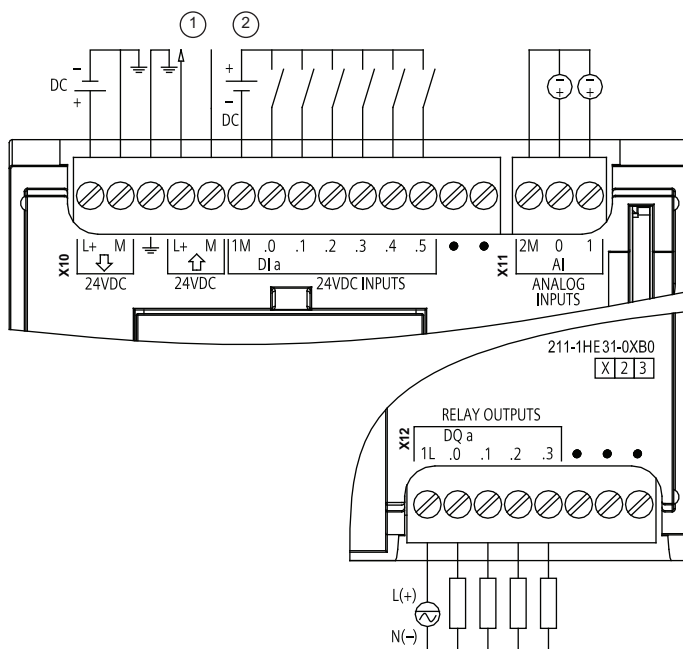
Remarque : Les connecteurs X11 doivent être en or. Vous trouverez les numéros de référence à l'annexe C, "Pièces de rechange".

Tableau A- 21 Brochage pour la CPU 1211C AC/DC/Relais (6ES7 211-1BE31-0XB0)

Broche	X10	X11 (or)	X12
1	L1 / 120-240 V~	2 M	1L
2	N / 120-240 V~	AI 0	DQ a.0
3	Terre fonctionnelle	AI 1	DQ a.1
4	L+ / Sortie capteur 24 V-	--	DQ a.2
5	M / Sortie capteur 24 V-	--	DQ a.3
6	1M	--	Pas de connexion
7	DI a.0	--	Pas de connexion

Broche	X10	X11 (or)	X12
8	DI a.1	--	Pas de connexion
9	DI a.2	--	--
10	DI a.3	--	--
11	DI a.4	--	--
12	DI a.5	--	--
13	Pas de connexion	--	--
14	Pas de connexion	--	--

Tableau A- 22CPU 1211C DC/DC/Relais (6ES7 211-1HE31-0XB0)



- ① Sortie alim. capteur 24 V-  
Pour plus d'immunité aux bruits, connectez "M" à la terre du châssis même si vous n'utilisez pas l'alimentation de capteur.
- ② Pour les entrées de type P, connectez "-" à "M" (comme illustré).  
Pour les entrées de type N, connectez "+" à "M".

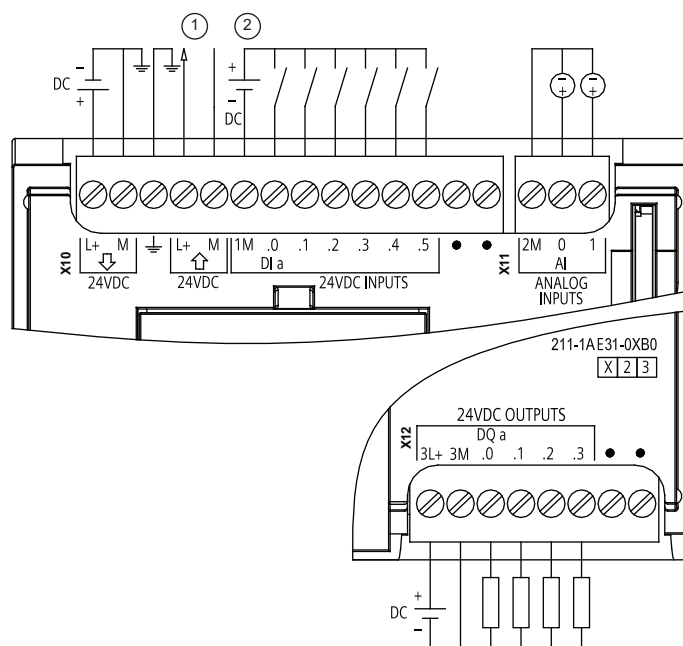
Remarque : Les connecteurs X11 doivent être en or. Vous trouverez les numéros de référence à l'annexe C, "Pièces de rechange".

Tableau A- 23Brochage pour la CPU 1211C DC/DC/Relais (6ES7 211-1HE31-0XB0)

Broche	X10	X11 (or)	X12
1	L+ / 24 V-	2 M	1L
2	M / 24 V-	AI 0	DQ a.0
3	Terre fonctionnelle	AI 1	DQ a.1
4	L+ / Sortie capteur 24 V-	--	DQ a.2
5	M / Sortie capteur 24 V-	--	DQ a.3
6	1M	--	Pas de connexion
7	DI a.0	--	Pas de connexion
8	DI a.1	--	Pas de connexion

Broche	X10	X11 (or)	X12
9	DI a.2	--	--
10	DI a.3	--	--
11	DI a.4	--	--
12	DI a.5	--	--
13	Pas de connexion	--	--
14	Pas de connexion	--	--

Tableau A- 24CPU 1211C DC/DC/DC (6ES7 211-1AE31-0XB0)



- ① Sortie alim. capteur 24 V-  
Pour plus d'immunité aux bruits, connectez "M" à la terre du châssis même si vous n'utilisez pas l'alimentation de capteur.
- ② Pour les entrées de type P, connectez "-" à "M" (comme illustré).  
Pour les entrées de type N, connectez "+" à "M".

Remarque : Les connecteurs X11 doivent être en or. Vous trouverez les numéros de référence à l'annexe C, "Pièces de rechange".

Tableau A- 25Brochage pour la CPU 1211C DC/DC/DC (6ES7 211-1AE31-0XB0)

Broche	X10	X11 (or)	X12
1	L+ / 24 V-	2 M	3L+
2	M / 24 V-	AI 0	3M
3	Terre fonctionnelle	AI 1	DQ a.0
4	L+ / Sortie capteur 24 V-	--	DQ a.1
5	M / Sortie capteur 24 V-	--	DQ a.2
6	1M	--	DQ a.3
7	DI a.0	--	Pas de connexion
8	DI a.1	--	Pas de connexion
9	DI a.2	--	--

A.3 CPU 1212C

Broche	X10	X11 (or)	X12
10	DI a.3	--	--
11	DI a.4	--	--
12	DI a.5	--	--
13	Pas de connexion	--	--
14	Pas de connexion	--	--

**Remarque**

Les entrées analogiques inutilisées doivent être court-circuitées.

## A.3 CPU 1212C

### A.3.1 Caractéristiques et fonctions générales

Tableau A- 26 Généralités

Caractéristiques techniques	CPU 1212C AC/DC/Relais	CPU 1212C DC/DC/Relais	CPU 1212C DC/DC/DC
Numéro de référence	6ES7 212-1BE31-0XB0	6ES7 212-1HE31-0XB0	6ES7 212-1AE31-0XB0
Dimensions L x H x P (mm)	90 x 100 x 75	90 x 100 x 75	90 x 100 x 75
Poids avec emballage	425 grammes	385 grammes	370 grammes
Dissipation de courant	11 W	9 W	9 W
Courant disponible (bus SM et CM)	1000 mA max. (5 V-)	1000 mA max. (5 V-)	1000 mA max. (5 V-)
Courant disponible (24 V-)	300 mA max. (alimentation de capteur)	300 mA max. (alimentation de capteur)	300 mA max. (alimentation de capteur)
Consommation de courant entrées TOR (24 V-)	4 mA / entrée utilisée	4 mA / entrée utilisée	4 mA / entrée utilisée

Tableau A- 27 Fonctions de la CPU

Caractéristiques techniques	Description	
Mémoire utilisateur <sup>1</sup>	de travail	50 Ko
	de chargement	1 Mo interne, extensible jusqu'à la taille de la carte SD
	rémanente	10 Ko
E/S TOR intégrées	8 entrées/6 sorties	
E/S analogiques intégrées	2 entrées	
Taille de la mémoire image	1024 octets d'entrées (I) / 1024 octets de sorties (Q)	

Caractéristiques techniques	Description
Mémentos (M)	4096 octets
Mémoire temporaire (locale)	<ul style="list-style-type: none"> <li>• 16 Ko pour le démarrage et le cycle du programme (FB et FC associés inclus)</li> <li>• 4 Ko pour les événements d'alarme standard, FB et FC inclus</li> <li>• 4 Ko pour les événements d'alarme d'erreur, FB et FC inclus</li> </ul>
Modules d'entrées-sorties pour extension	2 SM max.
SB, CB, BB pour extension	1 max.
Modules de communication pour extension	3 CM max.
Compteurs rapides	5 intégrés, 6 avec Signal Board, voir le tableau Affectations des entrées HSC pour la CPU 1212C (Page 361) <ul style="list-style-type: none"> <li>• Monophasé : 3 à 100 kHz et 1 à 30 kHz de fréquence d'horloge, SB : 2 à 30 kHz</li> <li>• Quadrature de phase : 3 à 80 kHz et 1 à 20 kHz de fréquence d'horloge, SB : 2 à 20 kHz</li> </ul>
Sorties d'impulsions <sup>2</sup>	4
Entrées de capture d'impulsions	8
Alarmes temporisées / cycliques	4 au total avec résolution de 1 ms
Alarmes sur front	8 sur front montant et 8 sur front descendant (12 et 12 avec Signal Board optionnel)
Carte mémoire	Carte mémoire SIMATIC (facultative)
Précision de l'horloge temps réel	+/- 60 secondes/mois
Durée de conservation de l'horloge temps réel	20 jours typ./12 jours min. à 40°C (supercondensateur sans maintenance)

<sup>1</sup> La taille du programme utilisateur, des données et de la configuration est limitée par la mémoire de chargement et la mémoire de travail disponibles dans la CPU. Il n'y a pas de limite spécifique pour le nombre de blocs OB, FC, FB et DB pris en charge ni pour la taille d'un bloc particulier. La seule limitation est la taille de mémoire globale.

<sup>2</sup> Pour les modèles de CPU avec sorties relais, vous devez installer un Signal Board (SB) TOR pour utiliser les sorties d'impulsions.

Tableau A- 28 Performances

Type d'instruction	Vitesse d'exécution
Logique booléenne	0,08 µs/instruction
Copier mot	1,7 µs/instruction
Mathématiques sur réels	2,3 µs/instruction

### A.3.2 Temporisations, compteurs et blocs de code pris en charge par la CPU 1212C

Tableau A- 29Blocs, temporisations et compteurs pris en charge par la CPU 1212C

Élément	Description	
Blocs	Type	OB, FB, FC, DB
	Taille	50 Ko
	Quantité	Jusqu'à 1024 blocs au total (OB + FB + FC + DB)
	Plage d'adresses pour les FB, FC et DB	1 à 65535 (FB 1 à FB 65535, par exemple)
	Profondeur d'imbrication	16 à partir de l'OB de cycle de programme ou de démarrage ; 4 à partir de l'OB d'alarme temporisée, d'alarme horaire, d'alarme cyclique, d'alarme de processus, d'erreur de temps ou d'alarme de diagnostic
	Visualisation	Il est possible de visualiser simultanément l'état de 2 blocs de code.
OB	Cycle de programme	Plusieurs : OB 1, OB 200 à OB 65535
	Mise en route	Plusieurs : OB 100, OB 200 à OB 65535
	Alarmes temporisées et alarmes cycliques	4 <sup>1</sup> (1 par événement) : OB 200 à OB 65535
	Alarmes de processus (fronts et HSC)	50 (1 par événement) : OB 200 à OB 65535
	Alarmes d'erreur de temps	1: OB 80
	Alarmes de diagnostic	1: OB 82
Temporisations	Type	CEI
	Quantité	Limitée uniquement par la taille de la mémoire
	Stockage	Structure dans un DB, 16 octets par temporisation
Compteurs	Type	CEI
	Quantité	Limitée uniquement par la taille de la mémoire
	Stockage	Structure dans un DB, la taille dépend du type de compteur <ul style="list-style-type: none"> <li>• SInt, USInt : 3 octets</li> <li>• Int, UInt : 6 octets</li> <li>• DInt, UDInt : 12 octets</li> </ul>

<sup>1</sup> Les alarmes temporisées et cycliques utilisent les mêmes ressources dans la CPU. Vous pouvez avoir uniquement 4 de ces alarmes au total (temporisées et cycliques). Vous ne pouvez pas avoir 4 alarmes temporisées et 4 alarmes cycliques.

Tableau A- 30Communication

Caractéristiques techniques	Description
Nombre de ports	1
Type	Ethernet
Appareil IHM <sup>1</sup>	3
Console de programmation (PG)	1

Caractéristiques techniques	Description
Liaisons	<ul style="list-style-type: none"> <li>8 pour la communication ouverte (Open User Communication) active ou passive : TSEND_C, TRCV_C, TCON, TDISCON, TSEND et TRCV</li> <li>3 pour la communication S7 (CPU à CPU) GET/PUT serveur</li> <li>8 pour la communication S7 (CPU à CPU) GET/PUT client</li> </ul>
Débits	10/100 Mb/s
Isolation (signal externe à logique API)	Isolé par transformateur, 1500 V-, pour une protection contre les événements de courte durée uniquement
Type de câble	CAT5e blindé

- <sup>1</sup> La CPU fournit des liaisons IHM spécialisées pour la prise en charge de 3 appareils IHM au maximum (vous pouvez avoir 2 SIMATIC Comfort Panels au maximum). Le nombre total de stations IHM dépend des types de pupitres IHM dans votre configuration. Vous pouvez, par exemple, avoir trois SIMATIC Basic Panels au plus connectés à votre CPU ou bien deux SIMATIC Comfort Panels au plus avec un Basic Panel additionnel.

Tableau A- 31Alimentation

Caractéristiques techniques	CPU 1212C AC/DC/Relais	CPU 1212C DC/DC/Relais	CPU 1212C DC/DC/DC
Plage de tension	85 à 264 V~	20,4 V- à 28,8 V- 22,0 V- à 28,8 V- à une température ambiante de -20 °C à 0 °C	
Fréquence de ligne	47 à 63 Hz	--	--
Courant d'entrée (charge max.)	CPU uniquement	80 mA à 120 V~ 40 mA à 240 V~	400 mA à 24 V-
	CPU avec tous les accessoires d'extension	240 mA à 120 V~ 120 mA à 240 V~	1200 mA à 24 V- 1200 mA à 24 V-
Appel de courant (max.)	20 A à 264 V~	12 A à 28,8 V-	12 A à 28,8 V-
Isolation (alimentation d'entrée à logique)	1500 V~	Non isolée	Non isolée
Fuite à la terre, ligne CA à la terre fonctionnelle	0,5 mA max.	--	--
Temps de retard (perte d'alimentation)	20 ms à 120 V~ 80 ms à 240 V~	10 ms à 24 V-	10 ms à 24 V-
Fusible interne, non remplaçable par l'utilisateur	3 A, 250 V, action retardée	3 A, 250 V, action retardée	3 A, 250 V, action retardée

Tableau A- 32Alimentation de capteur

Caractéristiques techniques	CPU 1212C AC/DC/Relais	CPU 1212C DC/DC/Relais	CPU 1212C DC/DC/DC
Plage de tension	20,4 à 28,8 V-	L+ moins 4 V- min. L+ moins 5 V- min. à une température ambiante de -20 °C à 0 °C	
Courant de sortie nominal (max.)	300 mA (protégé contre les courts-circuits)	300 mA (protégé contre les courts-circuits)	300 mA (protégé contre les courts-circuits)

A.3 CPU 1212C

Caractéristiques techniques	CPU 1212C AC/DC/Relais	CPU 1212C DC/DC/Relais	CPU 1212C DC/DC/DC
Bruit d'ondulation maximum (<10 MHz)	< 1 V crête à crête	Comme la ligne d'entrée	Comme la ligne d'entrée
Isolation (logique CPU à alimentation de capteur)	Non isolée	Non isolée	Non isolée

### A.3.3 Entrées et sorties TOR

Tableau A- 33 Entrées TOR

Caractéristiques techniques	CPU 1212C AC/DC/Relais, DC/DC/Relais et DC/DC/DC
Nombre d'entrées	8
Type	P/N (CEI type 1 en mode P)
Tension nominale	24 V- à 4 mA, nominal
Tension continue admise	30 V-, max.
Tension de choc	35 V- pour 0,5 s
Signal 1 logique (min.)	15 V- à 2,5 mA
Signal 0 logique (max.)	5 V- à 1 mA
Isolation (site à logique)	500 V~ pour 1 minute
Groupes d'isolation	1
Temps de filtre	0,2, 0,4, 0,8, 1,6, 3,2, 6,4 et 12,8 ms (sélectionnables par groupes de 4)
Fréquences d'entrée d'horloge HSC (max.) (niveau 1 logique = 15 à 26 V-)	Monophasé : 100 kHz (Ia.0 à Ia.5) et 30 kHz (Ia.6 à Ia.7) Quadrature de phase : 80 kHz (Ia.0 à Ia.5) et 20 kHz (Ia.6 à Ia.7)
Nombre d'entrées simultanément à 1	8 à 60 °C horizontal, 50 °C vertical
Longueur de câble (mètres)	500 m blindé, 300 m non blindé, 50 m blindé pour entrées HSC

Tableau A- 34 Sorties TOR

Caractéristiques techniques	CPU 1212C AC/DC/Relais et DC/DC/Relais	CPU 1212C DC/DC/DC
Nombre de sorties	6	6
Type	Relais, contact sec	Transistor à effet de champ MOS (mode N)
Plage de tension	5 à 30 V- ou 5 à 250 V~	20,4 à 28,8 V-
Signal 1 logique à courant max.	--	20 V- min.
Signal 0 logique avec charge 10 kΩ	--	0,1 V- max.
Courant (max.)	2,0 A	0,5 A
Charge de lampe	30 W CC / 200 W CA	5 W
Résistance état activé	0,2 Ω max. lorsque neuf	0,6 Ω max.
Courant de fuite par sortie	--	10 µA max.



Caractéristiques techniques	CPU 1212C AC/DC/Relais et DC/DC/Relais	CPU 1212C DC/DC/DC
Courant de choc	7 A avec contacts fermés	8 A pour 100 ms max.
Protection contre la surcharge	Non	Non
Isolation (site à logique)	1500 V~ pour 1 minute (bobine à contact) Aucune (bobine à logique)	500 V~ pour 1 minute
Résistance d'isolation	100 MΩ min. lorsque neuf	--
Isolation entre contacts ouverts	750 V~ pour 1 minute	--
Groupes d'isolation	2	1
Tension de blocage inductive	--	L+ moins 48 V-, dissipation 1 W
Retard de commutation (Qa.0 à Qa.3)	10 ms max.	1,0 μs max., de 0 à 1 3,0 μs max., de 1 à 0
Retard de commutation (Qa.4 à Qa.5)	10 ms max.	50 μs max., de 0 à 1 200 μs max., de 1 à 0
Fréquence de commutation maximum des relais	1 Hz	--
Fréquence de sortie de trains d'impulsions (Qa.0 et Qa.2)	Non recommandé <sup>1</sup>	100 kHz max., 2 Hz min. <sup>2</sup>
Durée d'utilisation mécanique (sans charge)	10 000 000 cycles ouvert/fermé	--
Durée d'utilisation des contacts à la charge nominale	100 000 cycles ouvert/fermé	--
Comportement au passage de MARCHE à ARRÊT	Dernière valeur ou valeur de remplacement (valeur par défaut 0)	Dernière valeur ou valeur de remplacement (valeur par défaut 0)
Nombre de sorties simultanément à 1	6 jusqu'à 60 °C horizontal, 50 °C vertical	
Longueur de câble (mètres)	500 m blindé, 150 m non blindé	500 m blindé, 150 m non blindé

<sup>1</sup> Pour les modèles de CPU avec sorties relais, vous devez installer un Signal Board (SB) TOR pour utiliser les sorties d'impulsions.

<sup>2</sup> Selon vos câble et récepteur d'impulsions, une résistance de charge supplémentaire (au moins 10% du courant nominal) peut améliorer la qualité du signal d'impulsion et l'immunité aux bruits.

### A.3.4 Entrées analogiques

Tableau A- 35 Entrées analogiques

Caractéristiques techniques	Description
Nombre d'entrées	2
Type	Tension (mode simple)
Plage pleine échelle	0 à 10 V
Plage pleine échelle (mot de données)	0 à 27648
Plage de dépassement haut	10,001 à 11,759 V

Caractéristiques techniques	Description
Plage de dépassement (mot de données)	27 649 à 32 511
Plage de débordement haut	11,760 à 11,852 V
Plage de débordement haut (mot de données)	32 512 à 32 767
Résolution	10 bits
Tension de tenue maximum	35 V-
Lissage	Aucun, faible, moyen ou fort Voir le tableau pour la réponse indicielle (ms) des entrées analogiques de la CPU (Page 770).
Réjection des bruits	10, 50 ou 60 Hz
Impédance	≥100 kΩ
Isolation (site à logique)	Aucune
Précision (25°C / -20 à 60°C)	3,0% / 3,5% de la pleine échelle
Longueur de câble (mètres)	100 m, paire torsadée blindée

#### A.3.4.1 Réponse indicielle des entrées analogiques intégrées de la CPU

Tableau A- 36 Réponse indicielle (ms), 0 V à 10 V mesuré à 95%

Sélection de lissage (moyennage d'échantillon)	Fréquence de réjection (temps d'intégration)		
	60 Hz	50 Hz	10 Hz
Aucun (1 cycle) : Pas de moyennage	50 ms	50 ms	100 ms
Faible (4 cycles) : 4 échantillons	60 ms	70 ms	200 ms
Moyen (16 cycles) : 16 échantillons	200 ms	240 ms	1150 ms
Fort (32 cycles) : 32 échantillons	400 ms	480 ms	2300 ms
<b>Temps d'échantillonnage</b>	<b>4,17 ms</b>	<b>5 ms</b>	<b>25 ms</b>

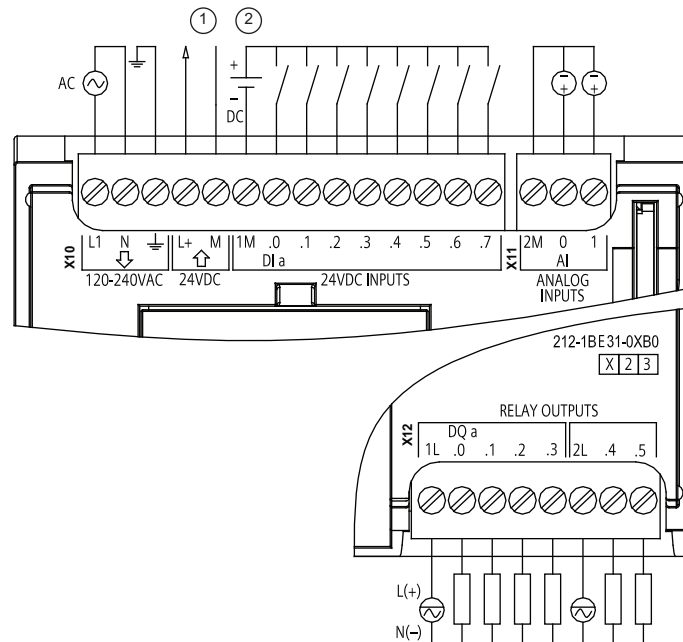
#### A.3.4.2 Temps d'échantillonnage pour les ports analogiques intégrés de la CPU

Tableau A- 37 Temps d'échantillonnage pour les entrées analogiques intégrées de la CPU

Fréquence de réjection (sélection de temps d'intégration)	Temps d'échantillonnage
60 Hz (16,6 ms)	4,17 ms
50 Hz (20 ms)	5 ms
10 Hz (100 ms)	25 ms

### A.3.5 Schémas de câblage de la CPU 1212C

Tableau A- 38CPU 1212C AC/DC/Relais (6ES7 212-1BE31-0XB0)



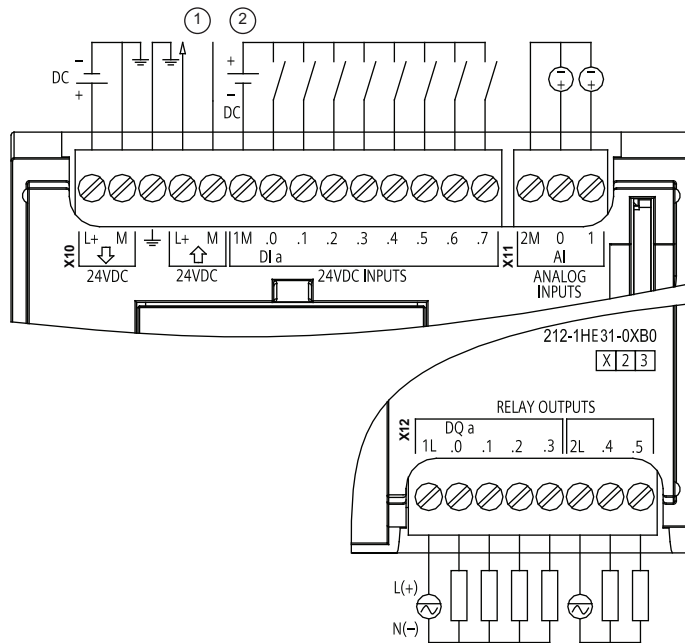
- ① Sortie alim. capteur 24 V-  
Pour plus d'immunité aux bruits, connectez "M" à la terre du châssis même si vous n'utilisez pas l'alimentation de capteur.
- ② Pour les entrées de type P, connectez "-" à "M" (comme illustré).  
Pour les entrées de type N, connectez "+" à "M".

Remarque : Les connecteurs X11 doivent être en or. Vous trouverez les numéros de référence à l'annexe C, "Pièces de rechange".

Tableau A- 39Brochage pour la CPU 1212C AC/DC/Relais (6ES7 212-1BE31-0XB0)

Broche	X10	X11 (or)	X12
1	L1 / 120-240 V~	2 M	1L
2	N / 120-240 V~	AI 0	DQ a.0
3	Terre fonctionnelle	AI 1	DQ a.1
4	L+ / Sortie capteur 24 V-	--	DQ a.2
5	M / Sortie capteur 24 V-	--	DQ a.3
6	1M	--	2L
7	DI a.0	--	DQ a.4
8	DI a.1	--	DQ a.5
9	DI a.2	--	--
10	DI a.3	--	--
11	DI a.4	--	--
12	DI a.5	--	--
13	DI a.6	--	--
14	DI a.7	--	--

Tableau A- 40CPU 1212C DC/DC/Relais (6ES7 212-1HE31-0XB0)



① Sortie alim. capteur 24 V-

Pour plus d'immunité aux bruits, connectez "M" à la terre du châssis même si vous n'utilisez pas l'alimentation de capteur.

② Pour les entrées de type P, connectez "-" à "M" (comme illustré).

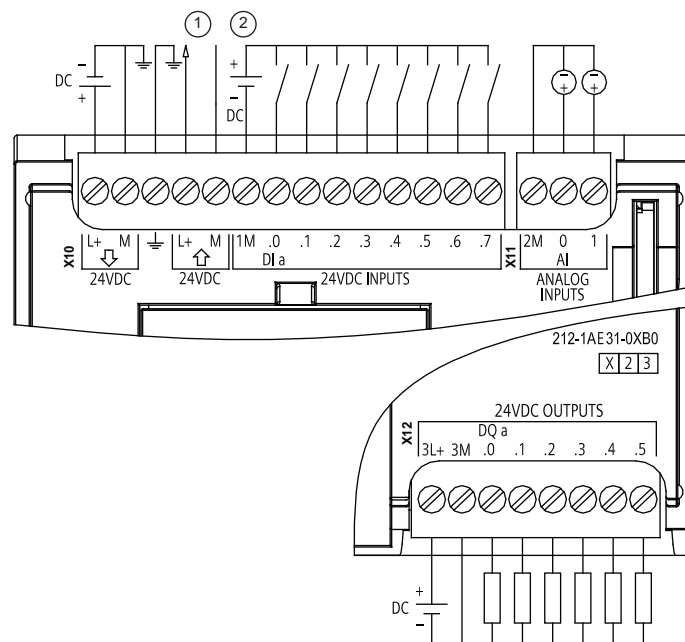
Pour les entrées de type N, connectez "+" à "M".

Remarque : Les connecteurs X11 doivent être en or. Vous trouverez les numéros de référence à l'annexe C, "Pièces de rechange".

Tableau A- 41Brochage pour la CPU 1212C DC/DC/Relais (6ES7 212-1HE31-0XB0)

Broche	X10	X11 (or)	X12
1	L+ / 24 V-	2 M	1L
2	M / 24 V-	AI 0	DQ a.0
3	Terre fonctionnelle	AI 1	DQ a.1
4	L+ / Sortie capteur 24 V-	--	DQ a.2
5	M / Sortie capteur 24 V-	--	DQ a.3
6	1M	--	2L
7	DI a.0	--	DQ a.4
8	DI a.1	--	DQ a.5
9	DI a.2	--	--
10	DI a.3	--	--
11	DI a.4	--	--
12	DI a.5	--	--
13	DI a.6	--	--
14	DI a.7	--	--

Tableau A- 42CPU 1212C DC/DC/DC (6ES7-212-1AE31-0XB0)



- ① Sortie alim. capteur 24 V-  
Pour plus d'immunité aux bruits, connectez "M" à la terre du châssis même si vous n'utilisez pas l'alimentation de capteur.
- ② Pour les entrées de type P, connectez "-" à "M" (comme illustré).  
Pour les entrées de type N, connectez "+" à "M".

Remarque : Les connecteurs X11 doivent être en or. Vous trouverez les numéros de référence à l'annexe C, "Pièces de rechange".

Tableau A- 43Brochage pour la CPU 1212C DC/DC/DC (6ES7 212-1AE31-0XB0)

Broche	X10	X11 (or)	X12
1	L+ / 24 V-	2 M	3L+
2	M / 24 V-	AI 0	3M
3	Terre fonctionnelle	AI 1	DQ a.0
4	L+ / Sortie capteur 24 V-	--	DQ a.1
5	M / Sortie capteur 24 V-	--	DQ a.2
6	1M	--	DQ a.3
7	DI a.0	--	DQ a.4
8	DI a.1	--	DQ a.5
9	DI a.2	--	--
10	DI a.3	--	--
11	DI a.4	--	--
12	DI a.5	--	--
13	DI a.6	--	--
14	DI a.7	--	--

**Remarque**

Les entrées analogiques inutilisées doivent être court-circuitées.

## A.4 CPU 1214C

### A.4.1 Caractéristiques et fonctions générales

Tableau A- 44 Généralités

Caractéristiques techniques	CPU 1214C AC/DC/Relais	CPU 1214C DC/DC/Relais	CPU 1214C DC/DC/DC
Numéro de référence	6ES7 214-1BG31-0XB0	6ES7 214-1HG31-0XB0	6ES7 214-1AG31-0XB0
Dimensions L x H x P (mm)	110 x 100 x 75	110 x 100 x 75	110 x 100 x 75
Poids avec emballage	475 grammes	435 grammes	415 grammes
Dissipation de courant	14 W	12 W	12 W
Courant disponible (bus SM et CM)	1600 mA max. (5 V-)	1600 mA max. (5 V-)	1600 mA max. (5 V-)
Courant disponible (24 V-)	400 mA max. (alimentation de capteur)	400 mA max. (alimentation de capteur)	400 mA max. (alimentation de capteur)
Consommation de courant entrées TOR (24 V-)	4 mA / entrée utilisée	4 mA / entrée utilisée	4 mA / entrée utilisée

Tableau A- 45 Fonctions de la CPU

Caractéristiques techniques	Description	
Mémoire utilisateur <sup>1</sup>	de travail	75 Ko
	de chargement	4 Mo interne, extensible jusqu'à la taille de la carte SD
	rémanente	10 Ko
E/S TOR intégrées	14 entrées/10 sorties	
E/S analogiques intégrées	2 entrées	
Taille de la mémoire image	1024 octets d'entrées (I) / 1024 octets de sorties (Q)	
Mémentos (M)	8192 octets	
Mémoire temporaire (locale)	<ul style="list-style-type: none"> <li>• 16 Ko pour le démarrage et le cycle du programme (FB et FC associés inclus)</li> <li>• 4 Ko pour les événements d'alarme standard, FB et FC inclus</li> <li>• 4 Ko pour les événements d'alarme d'erreur, FB et FC inclus</li> </ul>	

Caractéristiques techniques	Description
Modules d'entrées-sorties pour extension	8 SM max.
SB, CB, BB pour extension	1 max.
Modules de communication pour extension	3 CM max.
Compteurs rapides	6 au total, voir le tableau Fonctionnement du compteur rapide (Page 361) <ul style="list-style-type: none"> <li>• Monophasé : 3 à 100 kHz et 3 à 30 kHz de fréquence d'horloge</li> <li>• Quadrature de phase : 3 à 80 kHz et 3 à 20 kHz de fréquence d'horloge</li> </ul>
Sorties d'impulsions <sup>2</sup>	4
Entrées de capture d'impulsions	14
Alarmes temporisées / cycliques	4 au total avec résolution de 1 ms
Alarmes sur front	12 sur front montant et 12 sur front descendant (14 et 14 avec Signal Board optionnel)
Carte mémoire	Carte mémoire SIMATIC (facultative)
Précision de l'horloge temps réel	+/- 60 secondes/mois
Durée de conservation de l'horloge temps réel	20 jours typ./12 jours min. à 40°C (supercondensateur sans maintenance)

<sup>1</sup> La taille du programme utilisateur, des données et de la configuration est limitée par la mémoire de chargement et la mémoire de travail disponibles dans la CPU. Il n'y a pas de limite spécifique pour le nombre de blocs OB, FC, FB et DB pris en charge ni pour la taille d'un bloc particulier. La seule limitation est la taille de mémoire globale.

<sup>2</sup> Pour les modèles de CPU avec sorties relais, vous devez installer un Signal Board (SB) TOR pour utiliser les sorties d'impulsions.

Tableau A- 46 Performances

Type d'instruction	Vitesse d'exécution
Logique booléenne	0,08 µs/instruction
Copier mot	1,7 µs/instruction
Mathématiques sur réels	2,3 µs/instruction

## A.4.2 Temporisations, compteurs et blocs de code pris en charge par la CPU 1214C

Tableau A- 47 Blocs, temporisations et compteurs pris en charge par la CPU 1214C

Elément	Description	
Blocs	Type	OB, FB, FC, DB
	Taille	64 Ko
	Quantité	Jusqu'à 1024 blocs au total (OB + FB + FC + DB)
	Plage d'adresses pour les FB, FC et DB	1 à 65535 (FB 1 à FB 65535, par exemple)

Elément	Description
	Profondeur d'imbrication 16 à partir de l'OB de cycle de programme ou de démarrage ; 4 à partir de l'OB d'alarme temporisée, d'alarme horaire, d'alarme cyclique, d'alarme de processus, d'alarme d'erreur de temps ou d'alarme de diagnostic
	Visualisation Il est possible de visualiser simultanément l'état de 2 blocs de code.
OB	Cycle de programme Plusieurs : OB 1, OB 200 à OB 65535
	Mise en route Plusieurs : OB 100, OB 200 à OB 65535
	Alarmes temporisées et alarmes cycliques 4 <sup>1</sup> (1 par événement) : OB 200 à OB 65535
	Alarmes de processus (fronts et HSC) 50 (1 par événement) : OB 200 à OB 65535
	Alarmes d'erreur de temps 1: OB 80
	Alarmes de diagnostic 1: OB 82
Temporisations	Type CEI
	Quantité Limitée uniquement par la taille de la mémoire
	Stockage Structure dans un DB, 16 octets par temporisation
Compteurs	Type CEI
	Quantité Limitée uniquement par la taille de la mémoire
	Stockage Structure dans un DB, la taille dépend du type de compteur <ul style="list-style-type: none"> <li>• SInt, USInt : 3 octets</li> <li>• Int, UInt : 6 octets</li> <li>• DInt, UDInt : 12 octets</li> </ul>

<sup>1</sup> Les alarmes temporisées et cycliques utilisent les mêmes ressources dans la CPU. Vous pouvez avoir uniquement 4 de ces alarmes au total (temporisées et cycliques). Vous ne pouvez pas avoir 4 alarmes temporisées et 4 alarmes cycliques.

Tableau A- 48Communication

Caractéristiques techniques	Description
Nombre de ports	1
Type	Ethernet
Appareil IHM <sup>1</sup>	3
Console de programmation (PG)	1
Liaisons	<ul style="list-style-type: none"> <li>• 8 pour la communication ouverte (Open User Communication) active ou passive : TSEND_C, TRCV_C, TCON, TDISCON, TSEND et TRCV</li> <li>• 3 pour la communication S7 (CPU à CPU) GET/PUT serveur</li> <li>• 8 pour la communication S7 (CPU à CPU) GET/PUT client</li> </ul>
Débits	10/100 Mb/s



Caractéristiques techniques	Description
Isolation (signal externe à logique API)	Isolé par transformateur, 1500 V-, pour une protection contre les événements de courte durée uniquement
Type de câble	CAT5e blindé

<sup>1</sup> La CPU fournit des liaisons IHM spécialisées pour la prise en charge de 3 appareils IHM au maximum (vous pouvez avoir 2 SIMATIC Comfort Panels au maximum). Le nombre total de stations IHM dépend des types de pupitres IHM dans votre configuration. Vous pouvez, par exemple, avoir trois SIMATIC Basic Panels au plus connectés à votre CPU ou bien deux SIMATIC Comfort Panels au plus avec un Basic Panel additionnel.

Tableau A- 49Alimentation

Caractéristiques techniques		CPU 1214C AC/DC/Relais	CPU 1214C DC/DC/Relais	CPU 1214C DC/DC/DC
Plage de tension		85 à 264 V~	20,4 V- à 28,8 V- 22,0 V- à 28,8 V- à une température ambiante de -20 °C à 0 °C	
Fréquence de ligne		47 à 63 Hz	--	
Courant d'entrée (charge max.)	CPU uniquement	100 mA à 120 V~ 50 mA à 240 V~	500 mA à 24 V-	
	CPU avec tous les accessoires d'extension	300 mA à 120 V~ 150 mA à 240 V~	1500 mA à 24 V-	
Appel de courant (max.)		20 A à 264 V~	12 A à 28,8 V-	
Isolation (alimentation d'entrée à logique)		1500 V~	Non isolée	
Fuite à la terre, ligne CA à la terre fonctionnelle		0,5 mA max.	-	
Temps de retard (perte d'alimentation)		20 ms à 120 V~ 80 ms à 240 V~	10 ms à 24 V-	
Fusible interne, non remplaçable par l'utilisateur		3 A, 250 V, action retardée		

Tableau A- 50Alimentation de capteur

Caractéristiques techniques		CPU 1214C AC/DC/Relais	CPU 1214C DC/DC/Relais	CPU 1214C DC/DC/DC
Plage de tension		20,4 à 28,8 V-	L+ moins 4 V- min. L+ moins 5 V- min. à une température ambiante de -20 °C à 0 °C	
Courant de sortie nominal (max.)		400 mA (protégé contre les courts-circuits)		
Bruit d'ondulation maximum (<10 MHz)		< 1 V crête à crête	Comme la ligne d'entrée	
Isolation (logique CPU à alimentation de capteur)		Non isolée		

### A.4.3 Entrées et sorties TOR

Tableau A- 51 Entrées TOR

Caractéristiques techniques	CPU 1214C AC/DC/Relais	CPU 1214C DC/DC/Relais	CPU 1214C DC/DC/DC
Nombre d'entrées	14		
Type	P/N (CEI type 1 en mode P)		
Tension nominale	24 V- à 4 mA, nominal		
Tension continue admise	30 V- max.		
Tension de choc	35 V- pour 0,5 s		
Signal 1 logique (min.)	15 V- à 2,5 mA		
Signal 0 logique (max.)	5 V- à 1 mA		
Isolation (site à logique)	500 V~ pour 1 minute		
Groupes d'isolation	1		
Temps de filtre	0,2, 0,4, 0,8, 1,6, 3,2, 6,4 et 12,8 ms (sélectionnables par groupes de 4)		
Fréquences d'entrée d'horloge HSC (max.) (niveau 1 logique = 15 à 26 V-)	Monophasé : 100 KHz (Ia.0 à Ia.5) et 30 KHz (Ia.6 à Ib.5) Quadrature de phase : 80 KHz (Ia.0 à Ia.5) et 20 KHz (Ia.6 à Ib.5)		
Nombre d'entrées simultanément à 1	<ul style="list-style-type: none"> <li>• 7 (pas d'E/S adjacentes) à 60 °C horizontal ou 50 °C vertical</li> <li>• 14 à 55 °C horizontal ou 45 °C vertical</li> </ul>		
Longueur de câble (mètres)	500 m blindé, 300 m non blindé, 50 m blindé pour entrées HSC		

Tableau A- 52 Sorties TOR

Caractéristiques techniques	CPU 1214C AC/DC/Relais et DC/DC/Relais	CPU 1214C DC/DC/DC
Nombre de sorties	10	10
Type	Relais, contact sec	Transistor à effet de champ MOS (mode P)
Plage de tension	5 à 30 V- ou 5 à 250 V~	20,4 à 28,8 V-
Signal 1 logique à courant max.	--	20 V- min.
Signal 0 logique avec charge 10 kΩ	--	0,1 V- max.
Courant (max.)	2,0 A	0,5 A
Charge de lampe	30 W CC / 200 W CA	5 W
Résistance état activé	0,2 Ω max. lorsque neuf	0,6 Ω max.
Courant de fuite par sortie	--	10 µA max.
Courant de choc	7 A avec contacts fermés	8 A pour 100 ms max.
Protection contre la surcharge	Non	Non
Isolation (site à logique)	1500 V~ pour 1 minute (bobine à contact) Aucune (bobine à logique)	500 V~ pendant 1 minute
Résistance d'isolation	100 MΩ min. lorsque neuf	--
Isolation entre contacts ouverts	750 V~ pour 1 minute	--

Caractéristiques techniques	CPU 1214C AC/DC/Relais et DC/DC/Relais	CPU 1214C DC/DC/DC
Groupes d'isolation	2	1
Tension de blocage inductive	--	L+ moins 48 V-, dissipation 1 W
Retard de commutation (Qa.0 à Qa.3)	10 ms max.	1,0 µs max., de 0 à 1 3,0 µs max., de 1 à 0
Retard de commutation (Qa.4 à Qb.1)	10 ms max.	50 µs max., de 0 à 1 200 µs max., de 1 à 0
Fréquence de commutation maximum des relais	1 Hz	--
Fréquence de sortie de trains d'impulsions (Qa.0 et Qa.2)	Non recommandé <sup>1</sup>	100 kHz max., 2 Hz min. <sup>2</sup>
Durée d'utilisation mécanique (sans charge)	10 000 000 cycles ouvert/fermé	--
Durée d'utilisation des contacts à la charge nominale	100 000 cycles ouvert/fermé	--
Comportement au passage de MARCHE à ARRET	Dernière valeur ou valeur de remplacement (valeur par défaut 0)	
Nombre de sorties simultanément à 1	<ul style="list-style-type: none"> <li>• 5 (pas d'E/S adjacentes) à 60 °C horizontal ou 50 °C vertical</li> <li>• 10 à 55 °C horizontal ou 45 °C vertical</li> </ul>	
Longueur de câble (mètres)	500 m blindé, 150 m non blindé	

<sup>1</sup> Pour les modèles de CPU avec sorties relais, vous devez installer un Signal Board (SB) TOR pour utiliser les sorties d'impulsions.

<sup>2</sup> Selon vos câble et récepteur d'impulsions, une résistance de charge supplémentaire (au moins 10% du courant nominal) peut améliorer la qualité du signal d'impulsion et l'immunité aux bruits.

#### A.4.4 Entrées analogiques

Tableau A- 53 Entrées analogiques

Caractéristiques techniques	Description
Nombre d'entrées	2
Type	Tension (mode simple)
Plage pleine échelle	0 à 10 V
Plage pleine échelle (mot de données)	0 à 27648
Plage de dépassement haut	10,001 à 11,759 V
Plage de dépassement (mot de données)	27 649 à 32 511
Plage de débordement haut	11,760 à 11,852 V
Plage de débordement haut (mot de données)	32 512 à 32 767
Résolution	10 bits
Tension de tenue maximum	35 V-

Caractéristiques techniques	Description
Lissage	Aucun, faible, moyen ou fort Voir le tableau pour la réponse indicielle (ms) des entrées analogiques de la CPU (Page 780).
Réjection des bruits	10, 50 ou 60 Hz
Impédance	≥100 kΩ
Isolation (site à logique)	Aucune
Précision (25°C / -20 à 60°C)	3,0% / 3,5% de la pleine échelle
Longueur de câble (mètres)	100 m, paire torsadée blindée

#### A.4.4.1 Réponse indicielle des entrées analogiques intégrées de la CPU

Tableau A- 54 Réponse indicielle (ms), 0 V à 10 V mesuré à 95%

Sélection de lissage (moyennage d'échantillon)	Fréquence de réjection (temps d'intégration)		
	60 Hz	50 Hz	10 Hz
Aucun (1 cycle) : Pas de moyennage	50 ms	50 ms	100 ms
Faible (4 cycles) : 4 échantillons	60 ms	70 ms	200 ms
Moyen (16 cycles) : 16 échantillons	200 ms	240 ms	1150 ms
Fort (32 cycles) : 32 échantillons	400 ms	480 ms	2300 ms
<b>Temps d'échantillonnage</b>	<b>4,17 ms</b>	<b>5 ms</b>	<b>25 ms</b>

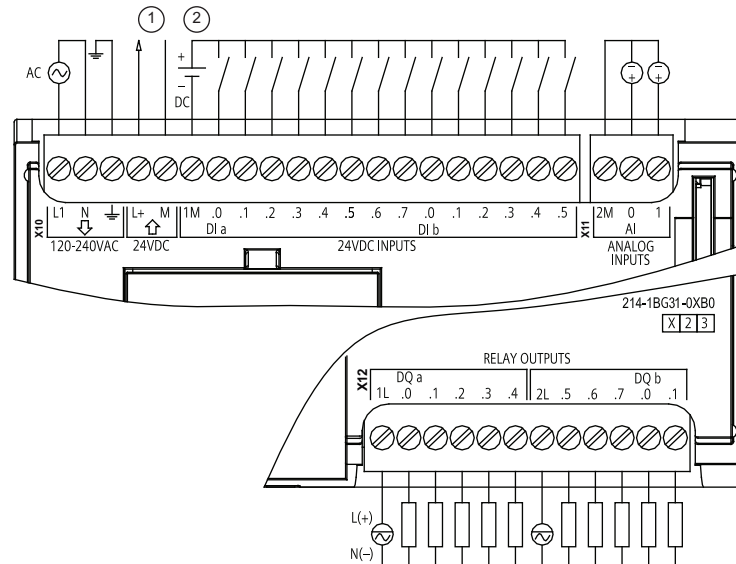
#### A.4.4.2 Temps d'échantillonnage pour les ports analogiques intégrés de la CPU

Tableau A- 55 Temps d'échantillonnage pour les entrées analogiques intégrées de la CPU

Fréquence de réjection (sélection de temps d'intégration)	Temps d'échantillonnage
60 Hz (16,6 ms)	4,17 ms
50 Hz (20 ms)	5 ms
10 Hz (100 ms)	25 ms

## A.4.5 Schémas de câblage de la CPU 1214C

Tableau A- 56CPU 1214C AC/DC/Relais (6ES7 214-1BG31-0XB0)



- ① Sortie alim. capteur 24 V-  
Pour plus d'immunité aux bruits, connectez "M" à la terre du châssis même si vous n'utilisez pas l'alimentation de capteur.
- ② Pour les entrées de type P, connectez "-" à "M" (comme illustré).  
Pour les entrées de type N, connectez "+" à "M".

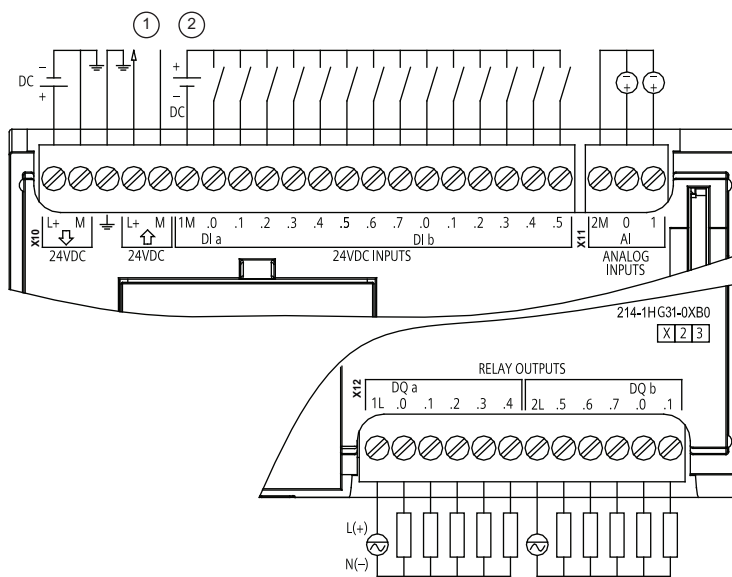
Remarque : Les connecteurs X11 doivent être en or. Vous trouverez les numéros de référence à l'annexe C, "Pièces de rechange".

Tableau A- 57Brochage pour la CPU 1214C AC/DC/Relais (6ES7 214-1BG31-0XB0)

Broche	X10	X11 (or)	X12
1	L1 / 120-240 V~	2 M	1L
2	N / 120-240 V~	AI 0	DQ a.0
3	Terre fonctionnelle	AI 1	DQ a.1
4	L+ / Sortie capteur 24 V-	--	DQ a.2
5	M / Sortie capteur 24 V-	--	DQ a.3
6	1M	--	DQ a.4
7	DI a.0	--	2L
8	DI a.1	--	DQ a.5
9	DI a.2	--	DQ a.6
10	DI a.3	--	DQ a.7
11	DI a.4	--	DQ b.0
12	DI a.5	--	DQ b.1
13	DI a.6	--	--
14	DI a.7	--	--
15	DI b.0	--	--
16	DI b.1	--	--

Broche	X10	X11 (or)	X12
17	DI b.2	--	--
18	DI b.3	--	--
19	DI b.4	--	--
20	DI b.5	--	--

Tableau A- 58CPU 1214C DC/DC/Relais (6ES7 214-1HG31-0XB0)



- ① Sortie alim. capteur 24 V-  
Pour plus d'immunité aux bruits, connectez "M" à la terre du châssis même si vous n'utilisez pas l'alimentation de capteur.
- ② Pour les entrées de type P, connectez "-" à "M" (comme illustré).  
Pour les entrées de type N, connectez "+" à "M".

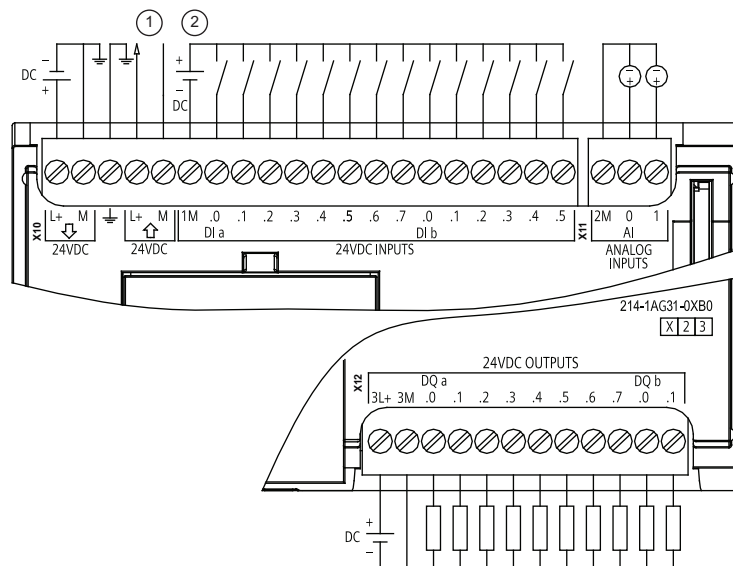
Remarque : Les connecteurs X11 doivent être en or. Vous trouverez les numéros de référence à l'annexe C, "Pièces de rechange".

Tableau A- 59Brochage pour la CPU 1214C DC/DC/Relais (6ES7 214-1HG31-0XB0)

Broche	X10	X11 (or)	X12
1	L+ / 24 V-	2 M	1L
2	M / 24 V-	AI 0	DQ a.0
3	Terre fonctionnelle	AI 1	DQ a.1
4	L+ / Sortie capteur 24 V-	--	DQ a.2
5	M / Sortie capteur 24 V-	--	DQ a.3
6	1M	--	DQ a.4
7	DI a.0	--	2L
8	DI a.1	--	DQ a.5
9	DI a.2	--	DQ a.6
10	DI a.3	--	DQ a.7
11	DI a.4	--	DQ b.0

Broche	X10	X11 (or)	X12
12	DI a.5	--	DQ b.1
13	DI a.6	--	--
14	DI a.7	--	--
15	DI b.0	--	--
16	DI b.1	--	--
17	DI b.2	--	--
18	DI b.3	--	--
19	DI b.4	--	--
20	DI b.5	--	--

Tableau A- 60CPU 1214C DC/DC/DC (6ES7 214-1AG31-0XB0)



- ① Sortie alim. capteur 24 V-  
Pour plus d'immunité aux bruits, connectez "M" à la terre du châssis même si vous n'utilisez pas l'alimentation de capteur.
- ② Pour les entrées de type P, connectez "-" à "M" (comme illustré).  
Pour les entrées de type N, connectez "+" à "M".

Remarque : Les connecteurs X11 doivent être en or. Vous trouverez les numéros de référence à l'annexe C, "Pièces de rechange".

Tableau A- 61Brochage pour la CPU 1214C DC/DC/DC (6ES7 214-1AG31-0XB0)

Broche	X10	X11 (or)	X12
1	L+ / 24 V-	2 M	3L+
2	M / 24 V-	AI 0	3M
3	Terre fonctionnelle	AI 1	DQ a.0
4	L+ / Sortie capteur 24 V-	--	DQ a.1
5	M / Sortie capteur 24 V-	--	DQ a.2
6	1M	--	DQ a.3

Broche	X10	X11 (or)	X12
7	DI a.0	--	DQ a.4
8	DI a.1	--	DQ a.5
9	DI a.2	--	DQ a.6
10	DI a.3	--	DQ a.7
11	DI a.4	--	DQ b.0
12	DI a.5	--	DQ b.1
13	DI a.6	--	--
14	DI a.7	--	-
15	DI b.0	--	--
16	DI b.1	--	--
17	DI b.2	--	--
18	DI b.3	--	--
19	DI b.4	--	--
20	DI b.5	--	--

**Remarque**

Les entrées analogiques inutilisées doivent être court-circuitées.

## A.5 CPU 1215C

### A.5.1 Caractéristiques et fonctions générales

Tableau A- 62 Généralités

Caractéristiques techniques	CPU 1215C AC/DC/Relais	CPU 1215C DC/DC/Relais	CPU 1215C DC/DC/DC
Numéro de référence	6ES7 215-1BG31-0XB0	6ES7 215-1HG31-0XB0	6ES7 215-1AG31-0XB0
Dimensions L x H x P (mm)	130 x 100 x 75	130 x 100 x 75	130 x 100 x 75
Poids avec emballage	550 grammes	585 grammes	520 grammes
Dissipation de courant	14 W	12 W	12 W
Courant disponible (bus SM et CM)	1600 mA max. (5 V-)	1600 mA max. (5 V-)	1600 mA max. (5 V-)
Courant disponible (24 V-)	400 mA max. (alimentation de capteur)	400 mA max. (alimentation de capteur)	400 mA max. (alimentation de capteur)
Consommation de courant entrées TOR (24 V-)	4 mA / entrée utilisée	4 mA / entrée utilisée	4 mA / entrée utilisée



Tableau A- 63 Fonctions de la CPU

Caractéristiques techniques		Description
Mémoire utilisateur <sup>1</sup>	de travail	100 Ko
	de chargement	4 Mo interne, extensible jusqu'à la taille de la carte SD
	rémanente	10 Ko
E/S TOR intégrées		14 entrées/10 sorties
E/S analogiques intégrées		2 entrées/2 sorties
Taille de la mémoire image		1024 octets d'entrées (I) / 1024 octets de sorties (Q)
Mémentos (M)		8192 octets
Mémoire temporaire (locale)		<ul style="list-style-type: none"> <li>• 16 Ko pour le démarrage et le cycle du programme (FB et FC associés inclus)</li> <li>• 4 Ko pour les événements d'alarme standard, FB et FC inclus</li> <li>• 4 Ko pour les événements d'alarme d'erreur, FB et FC inclus</li> </ul>
Modules d'entrées-sorties pour extension		8 SM max.
SB, CB, BB pour extension		1 max.
Modules de communication pour extension		3 CM max.
Compteurs rapides		6 au total, voir le tableau Affectations des entrées HSC pour la CPU 1215C <ul style="list-style-type: none"> <li>• Monophasé : 3 à 100 kHz et 3 à 30 kHz de fréquence d'horloge</li> <li>• Quadrature de phase : 3 à 80 kHz et 3 à 20 kHz de fréquence d'horloge</li> </ul>
Sorties d'impulsions <sup>2</sup>		4
Entrées de capture d'impulsions		14
Alarmes temporisées / cycliques		4 au total avec résolution de 1 ms
Alarmes sur front		12 sur front montant et 12 sur front descendant (14 et 14 avec Signal Board optionnel)
Carte mémoire		Carte mémoire SIMATIC (facultative)
Précision de l'horloge temps réel		+/- 60 secondes/mois
Durée de conservation de l'horloge temps réel		20 jours typ./12 jours min. à 40°C (supercondensateur sans maintenance)

<sup>1</sup> La taille du programme utilisateur, des données et de la configuration est limitée par la mémoire de chargement et la mémoire de travail disponibles dans la CPU. Il n'y a pas de limite spécifique pour le nombre de blocs OB, FC, FB et DB pris en charge ni pour la taille d'un bloc particulier. La seule limitation est la taille de mémoire globale.

<sup>2</sup> Pour les modèles de CPU avec sorties relais, vous devez installer un Signal Board (SB) TOR pour utiliser les sorties d'impulsions.

Tableau A- 64 Performances

Type d'instruction	Vitesse d'exécution
Logique booléenne	0,08 µs/instruction
Copier mot	1,7 µs/instruction
Mathématiques sur réels	2,3 µs/instruction

## A.5.2 Temporisations, compteurs et blocs de code pris en charge par la CPU 1215C

Tableau A- 65Blocs, temporisations et compteurs pris en charge par la CPU 1215C

Élément	Description	
Blocs	Type	OB, FB, FC, DB
	Taille	64 Ko
	Quantité	Jusqu'à 1024 blocs au total (OB + FB + FC + DB)
	Plage d'adresses pour les FB, FC et DB	1 à 65535 (FB 1 à FB 65535, par exemple)
	Profondeur d'imbrication	16 à partir de l'OB de cycle de programme ou de démarrage ; 4 à partir de l'OB d'alarme temporisée, d'alarme horaire, d'alarme cyclique, d'alarme de processus, d'erreur de temps ou d'alarme de diagnostic
	Visualisation	Il est possible de visualiser simultanément l'état de 2 blocs de code.
OB	Cycle de programme	Plusieurs : OB 1, OB 200 à OB 65535
	Mise en route	Plusieurs : OB 100, OB 200 à OB 65535
	Alarmes temporisées et alarmes cycliques	4 <sup>1</sup> (1 par événement) : OB 200 à OB 65535
	Alarmes de processus (fronts et HSC)	50 (1 par événement) : OB 200 à OB 65535
	Alarmes d'erreur de temps	1: OB 80
	Alarmes de diagnostic	1: OB 82
Temporisations	Type	CEI
	Quantité	Limitée uniquement par la taille de la mémoire
	Stockage	Structure dans un DB, 16 octets par temporisation
Compteurs	Type	CEI
	Quantité	Limitée uniquement par la taille de la mémoire
	Stockage	Structure dans un DB, la taille dépend du type de compteur <ul style="list-style-type: none"> <li>• SInt, USInt : 3 octets</li> <li>• Int, UInt : 6 octets</li> <li>• DInt, UDInt : 12 octets</li> </ul>

<sup>1</sup> Les alarmes temporisées et cycliques utilisent les mêmes ressources dans la CPU. Vous pouvez avoir uniquement 4 de ces alarmes au total (temporisées et cycliques). Vous ne pouvez pas avoir 4 alarmes temporisées et 4 alarmes cycliques.

Tableau A- 66Communication

Caractéristiques techniques	Description
Nombre de ports	2
Type	Ethernet
Appareil IHM <sup>1</sup>	3
Console de programmation (PG)	1

Caractéristiques techniques	Description
Liaisons	<ul style="list-style-type: none"> <li>• 8 pour la communication ouverte (Open User Communication) active ou passive : TSEND_C, TRCV_C, TCON, TDISCON, TSEND et TRCV</li> <li>• 3 pour la communication S7 (CPU à CPU) GET/PUT serveur</li> <li>• 8 pour la communication S7 (CPU à CPU) GET/PUT client</li> </ul>
Débits	10/100 Mb/s
Isolation (signal externe à logique API)	Isolé par transformateur, 1500 V-, pour une protection contre les événements de courte durée uniquement
Type de câble	CAT5e blindé

- <sup>1</sup> La CPU fournit des liaisons IHM spécialisées pour la prise en charge de 3 appareils IHM au maximum (vous pouvez avoir 2 SIMATIC Comfort Panels au maximum). Le nombre total de stations IHM dépend des types de pupitres IHM dans votre configuration. Vous pouvez, par exemple, avoir trois SIMATIC Basic Panels au plus connectés à votre CPU ou bien deux SIMATIC Comfort Panels au plus avec un Basic Panel additionnel.

Tableau A- 67Alimentation

Caractéristiques techniques		CPU 1215C AC/DC/Relais	CPU 1215C DC/DC/Relais	CPU 1215C DC/DC/DC
Plage de tension		85 à 264 V~	20,4 V- à 28,8 V- 22,0 V- à 28,8 V- à une température ambiante de -20 °C à 0 °C	
Fréquence de ligne		47 à 63 Hz	--	
Courant d'entrée (charge max.)	CPU uniquement	100 mA à 120 V~ 50 mA à 240 V~	500 mA à 24 V-	
	CPU avec tous les accessoires d'extension	300 mA à 120 V~ 150 mA à 240 V~	1500 mA à 24 V-	
Appel de courant (max.)		20 A à 264 V~	12 A à 28,8 V-	
Isolation (alimentation d'entrée à logique)		1500 V~	Non isolée	
Fuite à la terre, ligne CA à la terre fonctionnelle		0,5 mA max.	-	
Temps de retard (perte d'alimentation)		20 ms à 120 V~ 80 ms à 240 V~	10 ms à 24 V-	
Fusible interne, non remplaçable par l'utilisateur		3 A, 250 V, action retardée		

Tableau A- 68Alimentation de capteur

Caractéristiques techniques	CPU 1215C AC/DC/Relais	CPU 1215C DC/DC/Relais	CPU 1215C DC/DC/DC
Plage de tension	20,4 à 28,8 V-	L+ moins 4 V- min. L+ moins 5 V- min. à une température ambiante de -20 °C à 0 °C	
Courant de sortie nominal (max.)	400 mA (protégé contre les courts-circuits)		
Bruit d'ondulation maximum (<10 MHz)	< 1 V crête à crête	Comme la ligne d'entrée	
Isolation (logique CPU à alimentation de capteur)	Non isolée		

### A.5.3 Entrées et sorties TOR

Tableau A- 69Entrées TOR

Caractéristiques techniques	CPU 1215C AC/DC/Relais	CPU 1215C DC/DC/Relais	CPU 1215C DC/DC/DC
Nombre d'entrées	14		
Type	P/N (CEI type 1 en mode P)		
Tension nominale	24 V- à 4 mA, nominal		
Tension continue admise	30 V-, max.		
Tension de choc	35 V- pour 0,5 s		
Signal 1 logique (min.)	15 V- à 2,5 mA		
Signal 0 logique (max.)	5 V- à 1 mA		
Isolation (site à logique)	500 V~ pendant 1 minute		
Groupes d'isolation	1		
Temps de filtre	0,2, 0,4, 0,8, 1,6, 3,2, 6,4 et 12,8 ms (sélectionnables par groupes de 4)		
Fréquences d'entrée d'horloge HSC (max.) (niveau 1 logique = 15 à 26 V-)	Monophasé : 100 KHz (Ia.0 à Ia.5) et 30 KHz (Ia.6 à Ib.5) Quadrature de phase : 80 KHz (Ia.0 à Ia.5) et 20 KHz (Ia.6 à Ib.5)		
Nombre d'entrées simultanément à 1	<ul style="list-style-type: none"> <li>• 7 (pas d'E/S adjacentes) à 60 °C horizontal ou 50 °C vertical</li> <li>• 14 à 55 °C horizontal ou 45 °C vertical</li> </ul>		
Longueur de câble (mètres)	500 m blindé, 300 m non blindé, 50 m blindé pour entrées HSC		

Tableau A- 70Sorties TOR

Caractéristiques techniques	CPU 1215C AC/DC/Relais et CPU 1215C DC/DC/Relais	CPU 1215C DC/DC/DC
Nombre de sorties	10	10
Type	Relais, contact sec	Transistor à effet de champ MOS (mode P)
Plage de tension	5 à 30 V- ou 5 à 250 V~	20,4 à 28,8 V-
Signal 1 logique à courant max.	--	20 V- min.
Signal 0 logique avec charge 10 kΩ	--	0,1 V- max.
Courant (max.)	2,0 A	0,5 A
Charge de lampe	30 W CC / 200 W CA	5 W
Résistance état activé	0,2 Ω max. lorsque neuf	0,6 Ω max.
Courant de fuite par sortie	--	10 μA max.
Courant de choc	7 A avec contacts fermés	8 A pour 100 ms max.
Protection contre la surcharge	Non	Non
Isolation (site à logique)	1500 V~ pour 1 minute (bobine à contact) Aucune (bobine à logique)	500 V~ pendant 1 minute
Résistance d'isolation	100 MΩ min. lorsque neuf	--
Isolation entre contacts ouverts	750 V~ pendant 1 minute	--
Groupes d'isolation	2	1
Tension de blocage inductive	--	L+ moins 48 V-, dissipation 1 W
Retard de commutation (Qa.0 à Qa.3)	10 ms max.	1,0 μs max., de 0 à 1 3,0 μs max., de 1 à 0
Retard de commutation (Qa.4 à Qb.1)	10 ms max.	50 μs max., de 0 à 1 200 μs max., de 1 à 0
Fréquence de commutation maximum des relais	1 Hz	--
Fréquence de sortie de trains d'impulsions (Qa.0 et Qa.2)	Non recommandé <sup>1</sup>	100 kHz max., 2 Hz min. <sup>2</sup>
Durée d'utilisation mécanique (sans charge)	10 000 000 cycles ouvert/fermé	--
Durée d'utilisation des contacts à la charge nominale	100 000 cycles ouvert/fermé	--
Comportement au passage de MARCHE à ARRÊT	Dernière valeur ou valeur de remplacement (valeur par défaut 0)	
Nombre de sorties simultanément à 1	<ul style="list-style-type: none"> <li>• 5 (pas d'E/S adjacentes) à 60 °C horizontal ou 50 °C vertical</li> <li>• 10 à 55 °C horizontal ou 45 °C vertical</li> </ul>	
Longueur de câble (mètres)	500 m blindé, 150 m non blindé	

<sup>1</sup> Pour les modèles de CPU avec sorties relais, vous devez installer un Signal Board (SB) TOR pour utiliser les sorties d'impulsions.

<sup>2</sup> Selon vos câble et récepteur d'impulsions, une résistance de charge supplémentaire (au moins 10% du courant nominal) peut améliorer la qualité du signal d'impulsion et l'immunité aux bruits.

## A.5.4 Entrées et sorties analogiques

### A.5.4.1 Caractéristiques des entrées analogiques

Tableau A- 71 Entrées analogiques

Caractéristiques techniques	Description
Nombre d'entrées	2
Type	Tension (mode simple)
Plage pleine échelle	0 à 10 V
Plage pleine échelle (mot de données)	0 à 27648
Plage de dépassement haut	10,001 à 11,759 V
Plage de dépassement (mot de données)	27 649 à 32 511
Plage de débordement haut	11,760 à 11,852 V
Plage de débordement haut (mot de données)	32 512 à 32 767
Résolution	10 bits
Tension de tenue maximum	35 V-
Lissage	Aucun, faible, moyen ou fort Voir le tableau pour la réponse indicielle (ms) des entrées analogiques de la CPU.
Réjection des bruits	10, 50 ou 60 Hz
Impédance	≥100 kΩ
Isolation (site à logique)	Aucune
Précision (25°C / -20 à 60°C)	3,0% / 3,5% de la pleine échelle
Longueur de câble (mètres)	100 m, paire torsadée blindée

### A.5.4.2 Réponse indicielle des entrées analogiques intégrées de la CPU

Tableau A- 72 Réponse indicielle (ms), 0 V à 10 V mesuré à 95%

Sélection de lissage (moyennage d'échantillon)	Fréquence de réjection (temps d'intégration)		
	60 Hz	50 Hz	10 Hz
Aucun (1 cycle) : Pas de moyennage	50 ms	50 ms	100 ms
Faible (4 cycles) : 4 échantillons	60 ms	70 ms	200 ms
Moyen (16 cycles) : 16 échantillons	200 ms	240 ms	1150 ms
Fort (32 cycles) : 32 échantillons	400 ms	480 ms	2300 ms
<b>Temps d'échantillonnage</b>	<b>4,17 ms</b>	<b>5 ms</b>	<b>25 ms</b>

### A.5.4.3 Temps d'échantillonnage pour les ports analogiques intégrés de la CPU

Tableau A- 73 Temps d'échantillonnage pour les entrées analogiques intégrées de la CPU

Fréquence de réjection (sélection de temps d'intégration)	Temps d'échantillonnage
60 Hz (16,6 ms)	4,17 ms
50 Hz (20 ms)	5 ms
10 Hz (100 ms)	25 ms

### A.5.4.4 Caractéristiques des sorties analogiques

#### Sorties analogiques

Tableau A- 74 Sorties analogiques

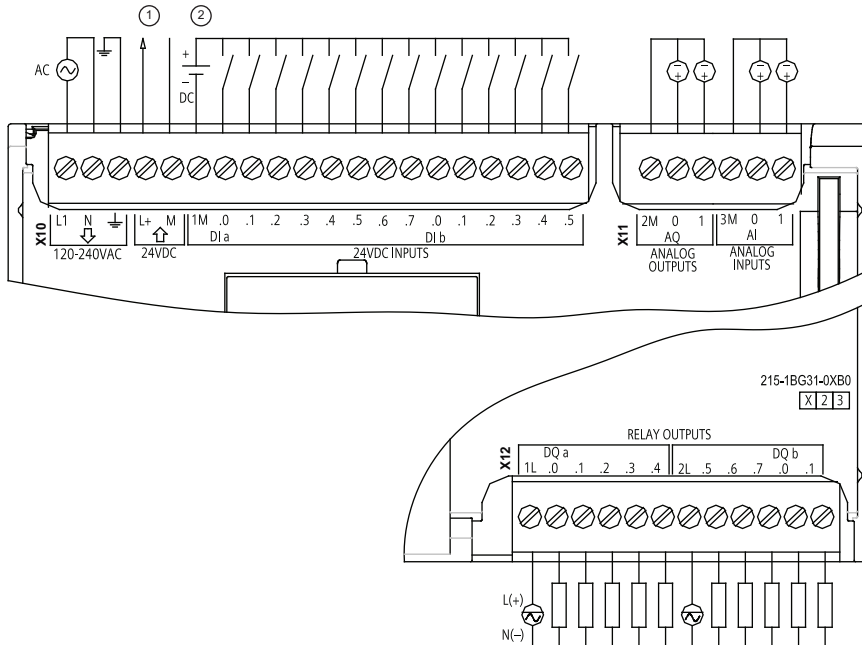
Caractéristiques techniques	Description
Nombre de sorties	2
Type	Courant
Plage pleine échelle	0 à 20 mA
Plage pleine échelle (mot de données)	0 à 27648
Plage de dépassement haut	20,01 à 23,52 mA <sup>1</sup>
Plage de dépassement (mot de données)	27 649 à 32 511
Plage de débordement haut	voir la note <sup>2</sup>
Plage de débordement haut (mot de données)	32 512 à 32 767
Résolution	10 bits
Impédance de sortie	≤500 Ω max.
Isolation (site à logique)	Aucune
Précision (25°C / -20 à 60°C)	3,0% / 3,5% de la pleine échelle
Temps d'établissement	2 ms
Longueur de câble (mètres)	100 m, paire torsadée blindée

<sup>1</sup> CPU 1215C avec alimentation en courant continu : à 20,4 V- de tension d'alimentation, une impédance de sortie allant jusqu'à 400 Ω est prise en charge dans la plage de dépassement haut.

<sup>2</sup> En cas de débordement haut, les sorties analogiques se comportent conformément aux paramétrages des propriétés de configuration d'appareil. Pour le paramètre "Réaction à l'arrêt de la CPU", sélectionnez : Appliquer valeur de remplacement ou Conserver dernière valeur.

### A.5.5 Schémas de câblage de la CPU 1215C

Tableau A- 75CPU 1215C AC/DC/Relais (6ES7 215-1BG31-0XB0)



- ① Sortie alim. capteur 24 V-  
Pour plus d'immunité aux bruits, connectez "M" à la terre du châssis même si vous n'utilisez pas l'alimentation de capteur.
- ② Pour les entrées de type P, connectez "-" à "M" (comme illustré).  
Pour les entrées de type N, connectez "+" à "M".

Remarque : Les connecteurs X11 doivent être en or. Vous trouverez les numéros de référence à l'annexe C, "Pièces de rechange".

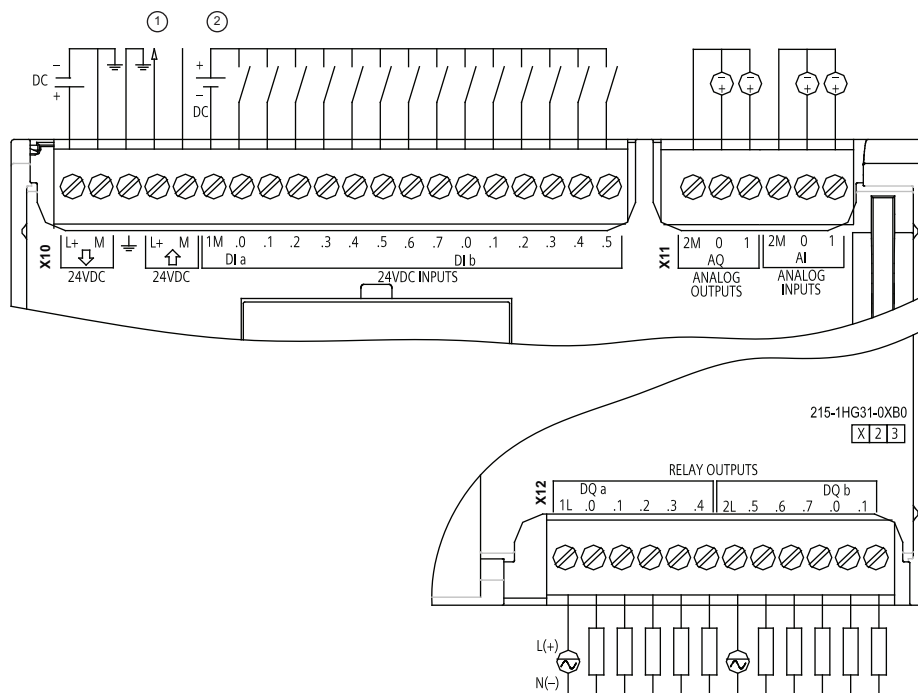
Tableau A- 76Brochage pour la CPU 1215C AC/DC/Relais (6ES7 215-1BG31-0XB0)

Broche	X10	X11 (or)	X12
1	L1 / 120-240 V~	2 M	1L
2	N / 120 - 240 V~	AQ 0	DQ a.0
3	Terre fonctionnelle	AQ 1	DQ a.1
4	L+ / Sortie capteur 24 V-	3M	DQ a.2
5	M / Sortie capteur 24 V-	AI 0	DQ a.3
6	1M	AI 1	DQ a.4
7	DI a.0	--	2L
8	DI a.1	--	DQ a.5
9	DI a.2	--	DQ a.6
10	DI a.3	--	DQ a.7
11	DI a.4	--	DQ b.0



Broche	X10	X11 (or)	X12
12	DI a.5	--	DQ b.1
13	DI a.6	--	--
14	DI a.7	--	--
15	DI b.0	--	--
16	DI b.1	--	--
17	DI b.2	--	--
18	DI b.3	--	--
19	DI b.4	--	--
20	DI b.5	--	--

Tableau A- 77CPU 1215C DC/DC/Relais (6ES7 215-1HG31-0XB0)



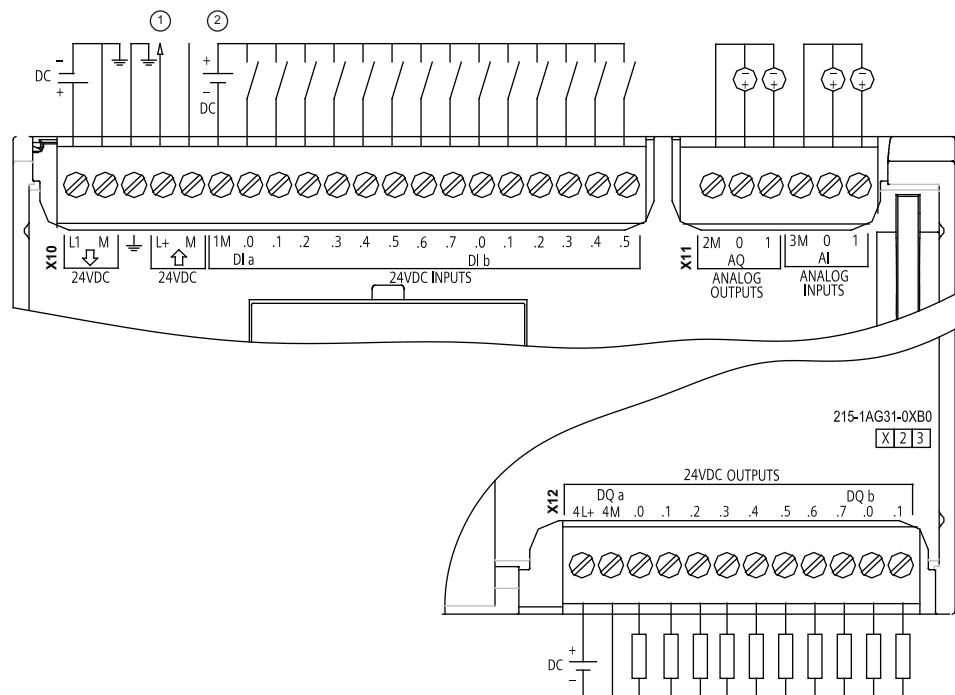
- ① Sortie alim. capteur 24 V-  
Pour plus d'immunité aux bruits, connectez "M" à la terre du châssis même si vous n'utilisez pas l'alimentation de capteur.
- ② Pour les entrées de type P, connectez "-" à "M" (comme illustré). Pour les entrées de type N, connectez "+" à "M".

Remarque : Les connecteurs X11 doivent être en or. Vous trouverez les numéros de référence à l'annexe C, "Pièces de rechange".

Tableau A- 78 Brochage pour la CPU 1215C DC/DC/Relais (6ES7 215-1HG31-0XB0)

Broche	X10	X11 (or)	X12
1	L+ / 24 V-	2 M	1L
2	M / 24 V-	AQ 0	DQ a.0
3	Terre fonctionnelle	AQ 1	DQ a.1
4	L+ / Sortie capteur 24 V-	2M	DQ a.2
5	M / Sortie capteur 24 V-	AI 0	DQ a.3
6	1M	AI 1	DQ a.4
7	DI a.0	--	2L
8	DI a.1	--	DQ a.5
9	DI a.2	--	DQ a.6
10	DI a.3	--	DQ a.7
11	DI a.4	--	DQ b.0
12	DI a.5	--	DQ b.1
13	DI a.6	--	--
14	DI a.7	--	--
15	DI b.0	--	--
16	DI b.1	--	--
17	DI b.2	--	--
18	DI b.3	--	--
19	DI b.4	--	--
20	DI b.5	--	--

Tableau A- 79CPU 1215C DC/DC/DC (6ES7 215-1AG31-0XB0)



- ① Sortie alim. capteur 24 V-  
Pour plus d'immunité aux bruits, connectez "M" à la terre du châssis même si vous n'utilisez pas l'alimentation de capteur.
- ② Pour les entrées de type P, connectez "-" à "M" (comme illustré). Pour les entrées de type N, connectez "+" à "M".

Remarque : Les connecteurs X11 doivent être en or. Vous trouverez les numéros de référence à l'annexe C, "Pièces de rechange".

Tableau A- 80Brochage pour la CPU 1215C DC/DC/DC (6ES7 215-1AG31-0XB0)

Broche	X10	X11 (or)	X12
1	L1 / 24 V-	2 M	4L+
2	M / 24 V-	AQ 0	4M
3	Terre fonctionnelle	AQ 1	DQ a.0
4	L+ / Sortie capteur 24 V-	3M	DQ a.1
5	M / Sortie capteur 24 V-	AI 0	DQ a.2
6	1M	AI 1	DQ a.3
7	DI a.0	--	DQ a.4
8	DI a.1	--	DQ a.5
9	DI a.2	--	DQ a.6
10	DI a.3	--	DQ a.7
11	DI a.4	--	DQ b.0
12	DI a.5	--	DQ b.1

Broche	X10	X11 (or)	X12
13	DI a.6	--	--
14	DI a.7	--	--
15	DI b.0	--	--
16	DI b.1	--	--
17	DI b.2	--	--
18	DI b.3	--	--
19	DI b.4	--	--
20	DI b.5	--	--

**Remarque**

Les entrées analogiques inutilisées doivent être court-circuitées.

## A.6 Modules d'entrées-sorties TOR (SM)

### A.6.1 Caractéristiques des modules d'entrées TOR SM 1221

Tableau A- 81Caractéristiques générales

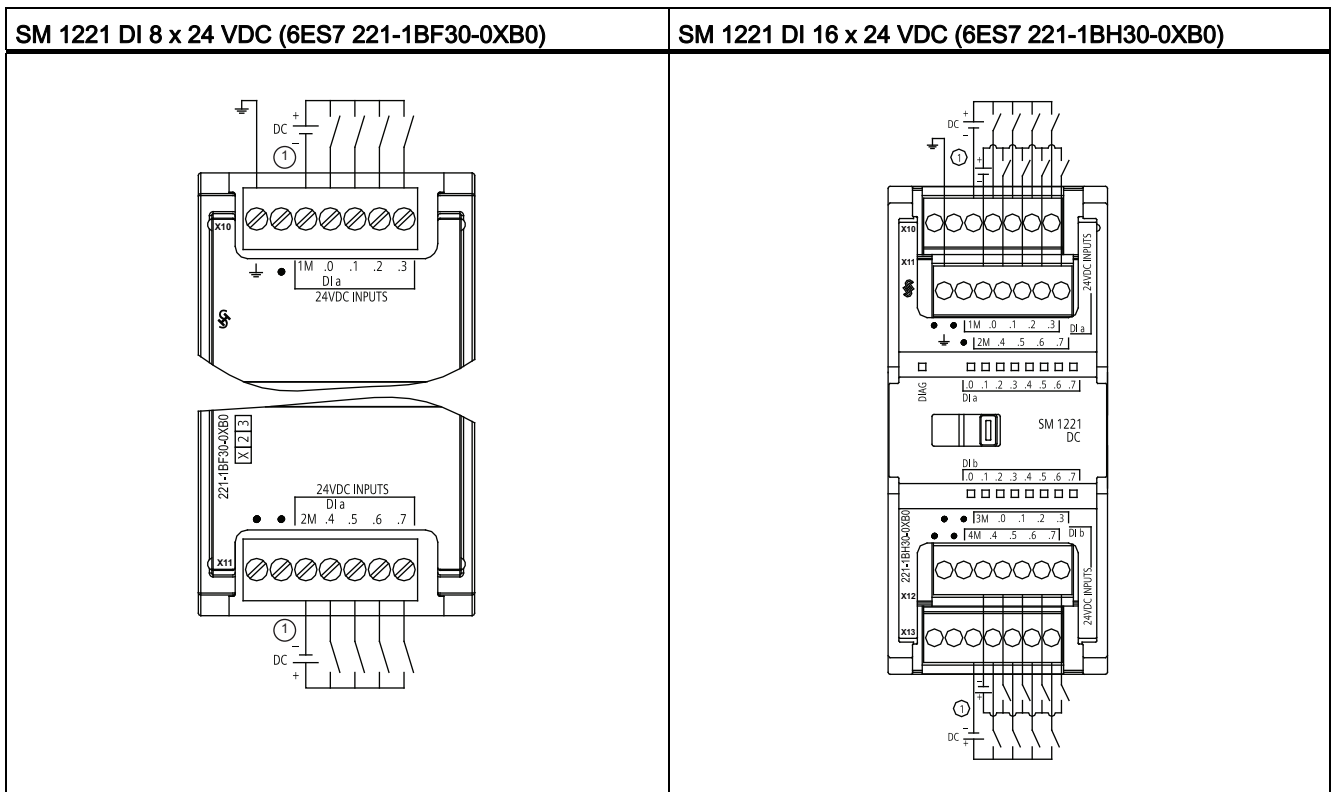
Modèle	SM 1221 DI 8 x 24 VDC	SM 1221 DI 16 x 24 VDC
Numéro de référence	6ES7 221-1BF30-0XB0	6ES7 221-1BH30-0XB0
Dimensions L x H x P (mm)	45 x 100 x 75	45 x 100 x 75
Poids	170 grammes	210 grammes
Dissipation de courant	1,5 W	2,5 W
Consommation de courant (bus SM)	105 mA	130 mA
Consommation de courant (24 V-)	4 mA/entrée utilisée	4 mA/entrée utilisée

Tableau A- 82Entrées TOR

Modèle	SM 1221 DI 8 x 24 VDC	SM 1221 DI 16 x 24 VDC
Nombre d'entrées	8	16
Type	P/N (CEI type 1 en mode P)	P/N (CEI type 1 en mode P)
Tension nominale	24 V- à 4 mA, nominal	24 V- à 4 mA, nominal
Tension continue admise	30 V-, max.	30 V-, max.
Tension de choc	35 V- pour 0,5 s	35 V- pour 0,5 s
Signal 1 logique (min.)	15 V- à 2,5 mA	15 V- à 2,5 mA
Signal 0 logique (max.)	5 V- à 1 mA	5 V- à 1 mA

Modèle	SM 1221 DI 8 x 24 VDC	SM 1221 DI 16 x 24 VDC
Isolation (site à logique)	500 V~ pour 1 minute	500 V~ pour 1 minute
Groupes d'isolation	2	4
Temps de filtre	0,2, 0,4, 0,8, 1,6, 3,2, 6,4 et 12,8 ms (sélectionnables par groupes de 4)	0,2, 0,4, 0,8, 1,6, 3,2, 6,4 et 12,8 ms (sélectionnables par groupes de 4)
Nombre d'entrées simultanément à 1	8	16
Longueur de câble (mètres)	500 m blindé, 300 m non blindé	500 m blindé, 300 m non blindé

Tableau A- 83 Schémas de câblage des modules SM d'entrées TOR



① Pour les entrées de type P, connectez "-" à "M". Pour les entrées de type N, connectez "+" à "M".

Tableau A- 84 Brochage pour le SM 1221 DI 8 x 24 VDC (6ES7 221-1BF30-0XB0)

Broche	X10	X11
1	GND	Pas de connexion
2	Pas de connexion	Pas de connexion
3	1M	2M
4	DI a.0	DI a.4
5	DI a.1	DI a.5
6	DI a.2	DI a.6
7	DI a.3	DI a.7

Tableau A- 85 Brochage pour le SM 1221 DI 16 x 24 VDC (6ES7 221-1BH30-0XB0)

Broche	X10	X11	X12	X13
1	GND	Pas de connexion	Pas de connexion	Pas de connexion
2	Pas de connexion	Pas de connexion	Pas de connexion	Pas de connexion
3	1M	2M	3 M	4 M
4	DI a.0	DI a.4	DI b.0	DI b.4
5	DI a.1	DI a.5	DI b.1	DI b.5
6	DI a.2	DI a.6	DI b.2	DI b.6
7	DI a.3	DI a.7	DI b.3	DI b.7

## A.6.2 Caractéristiques des modules 8 sorties TOR SM 1222

Tableau A- 86 Caractéristiques générales

Modèle	SM 1222 DQ 8 x Relais	SM 1222 DQ8 RLY inverseur	SM 1222 DQ 8 x 24 VDC
Numéro de référence	6ES7 222-1HF30-0XB0	6ES7 222-1XF30-0XB0	6ES7 222-1BF30-0XB0
Dimensions L x H x P (mm)	45 x 100 x 75	70 x 100 x 75	45 x 100 x 75
Poids	190 grammes	310 grammes	180 grammes
Dissipation de courant	4,5 W	5 W	1,5 W
Consommation de courant (bus SM)	120 mA	140 mA	120 mA
Consommation de courant (24 V-)	11 mA / bobine de relais utilisée	16,7 mA / bobine de relais utilisée	11 mA / bobine de relais utilisée

Tableau A- 87 Sorties TOR

Modèle	SM 1222 DQ 8 x Relais	SM 1222 DQ8 RLY inverseur	SM 1222 DQ 8 x 24 VDC
Nombre de sorties	8	8	8
Type	Relais, contact sec	Relais, contact inverseur	Transistor à effet de champ MOS (mode P)
Plage de tension	5 à 30 VCC ou 5 à 250 VCA	5 à 30 VCC ou 5 à 250 VCA	20,4 à 28,8 V-
Signal 1 logique à courant max.	--	--	20 V- min.
Signal 0 logique avec charge 10 kΩ	--	--	0,1 V- max.
Courant (max.)	2,0 A	2,0 A	0,5 A
Charge de lampe	30 W CC / 200 W CA	30 W CC / 200 W CA	5 W
Résistance état activé (contact)	0,2 Ω max. lorsque neuf	0,2 Ω max. lorsque neuf	0,6 Ω max.
Courant de fuite par sortie	--	--	10 µA max.

Modèle	SM 1222 DQ 8 x Relais	SM 1222 DQ8 RLY inverseur	SM 1222 DQ 8 x 24 VDC
Courant de choc	7 A avec contacts fermés	7 A avec contacts fermés	8 A pour 100 ms max.
Protection contre la surcharge	Non	Non	Non
Isolation (site à logique)	1500 V~ pour 1 minute (bobine à contact) Aucune (bobine à logique)	1500 V~ pour 1 minute (bobine à contact)	500 V~ pour 1 minute
Résistance d'isolation	100 MΩ min. lorsque neuf	100 MΩ min. lorsque neuf	--
Isolation entre contacts ouverts	750 V~ pour 1 minute	750 V~ pour 1 minute	--
Groupes d'isolation	2	8	1
Courant par commun (max.)	10 A	2 A	4 A
Tension de blocage inductive	--	--	L+ moins 48 V, dissipation 1 W
Retard de commutation	10 ms max.	10 ms max.	50 μs max. de 0 à 1 200 μs max. de 1 à 0
Fréquence de commutation maximum des relais	1 Hz	1 Hz	--
Durée d'utilisation mécanique (sans charge)	10 000 000 cycles ouvert/fermé	10 000 000 cycles ouvert/fermé	--
Durée d'utilisation des contacts à la charge nominale	100 000 cycles ouvert/fermé	100 000 cycles ouvert/fermé	--
Comportement au passage de MARCHE à ARRÊT	Dernière valeur ou valeur de remplacement (valeur par défaut 0)	Dernière valeur ou valeur de remplacement (valeur par défaut 0)	Dernière valeur ou valeur de remplacement (valeur par défaut 0)
Nombre de sorties simultanément à 1	8	<ul style="list-style-type: none"> <li>• 4 (pas d'E/S adjacentes) à 60 °C horizontal ou 50 °C vertical</li> <li>• 8 à 55 °C horizontal ou 45 °C vertical</li> </ul>	8
Longueur de câble (mètres)	500 m blindé, 150 m non blindé	500 m blindé, 150 m non blindé	500 m blindé, 150 m non blindé

### A.6.3 Caractéristiques des modules 16 sorties TOR SM 1222

Tableau A- 88Caractéristiques générales

Modèle	SM 1222 DQ 16 x Relais	SM 1222 DQ 16 x 24 VDC
Numéro de référence	6ES7 222-1HH30-0XB0	6ES7 222-1BH30-0XB0
Dimensions L x H x P (mm)	45 x 100 x 75	45 x 100 x 75
Poids	260 grammes	220 grammes

*Caractéristiques techniques*

*A.6 Modules d'entrées-sorties TOR (SM)*

<b>Modèle</b>	<b>SM 1222 DQ 16 x Relais</b>	<b>SM 1222 DQ 16 x 24 VDC</b>
Dissipation de courant	8,5 W	2,5 W
Consommation de courant (bus SM)	135 mA	140 mA
Consommation de courant (24 V-)	11 mA / bobine de relais utilisée	-

Tableau A- 89Sorties TOR

<b>Modèle</b>	<b>SM1222 DQ 16 x Relais</b>	<b>SM1222 DQ 16 x 24 VDC</b>
Nombre de sorties	16	16
Type	Relais, contact sec	Transistor à effet de champ MOS (mode P)
Plage de tension	5 à 30 V- ou 5 à 250 V~	20,4 à 28,8 V-
Signal 1 logique à courant max.	-	20 V- min.
Signal 0 logique avec charge 10 kΩ	-	0,1 V- max.
Courant (max.)	2,0 A	0,5 A
Charge de lampe	30 W CC / 200 W CA	5 W
Résistance état activé (contact)	0,2 Ω max. lorsque neuf	0,6 Ω max.
Courant de fuite par sortie	--	10 µA max.
Courant de choc	7 A avec contacts fermés	8 A pour 100 ms max.
Protection contre la surcharge	Non	Non
Isolation (site à logique)	1500 V~ pour 1 minute (bobine à contact) Aucune (bobine à logique)	500 V~ pour 1 minute
Résistance d'isolation	100 MΩ min. lorsque neuf	-
Isolation entre contacts ouverts	750 V~ pour 1 minute	-
Groupes d'isolation	4	1
Courant par commun (max.)	10 A	8 A
Tension de blocage inductive	-	L+ moins 48 V, dissipation 1 W
Retard de commutation	10 ms max.	50 µs max. de 0 à 1 200 µs max. de 1 à 0
Fréquence de commutation maximum des relais	1 Hz	-
Durée d'utilisation mécanique (sans charge)	10 000 000 cycles ouvert/fermé	-
Durée d'utilisation des contacts à la charge nominale	100 000 cycles ouvert/fermé	-
Comportement au passage de MARCHE à ARRET	Dernière valeur ou valeur de remplacement (valeur par défaut 0)	Dernière valeur ou valeur de remplacement (valeur par défaut 0)



Modèle	SM1222 DQ 16 x Relais	SM1222 DQ 16 x 24 VDC
Nombre de sorties simultanément à 1	<ul style="list-style-type: none"> <li>8 (pas d'E/S adjacentes) à 60 °C horizontal ou 50 °C vertical</li> <li>16 à 55 °C horizontal ou 45 °C vertical</li> </ul>	16
Longueur de câble (mètres)	500 m blindé, 150 m non blindé	500 m blindé, 150 m non blindé

Tableau A- 90 Schémas de câblage des modules SM à 8 sorties TOR

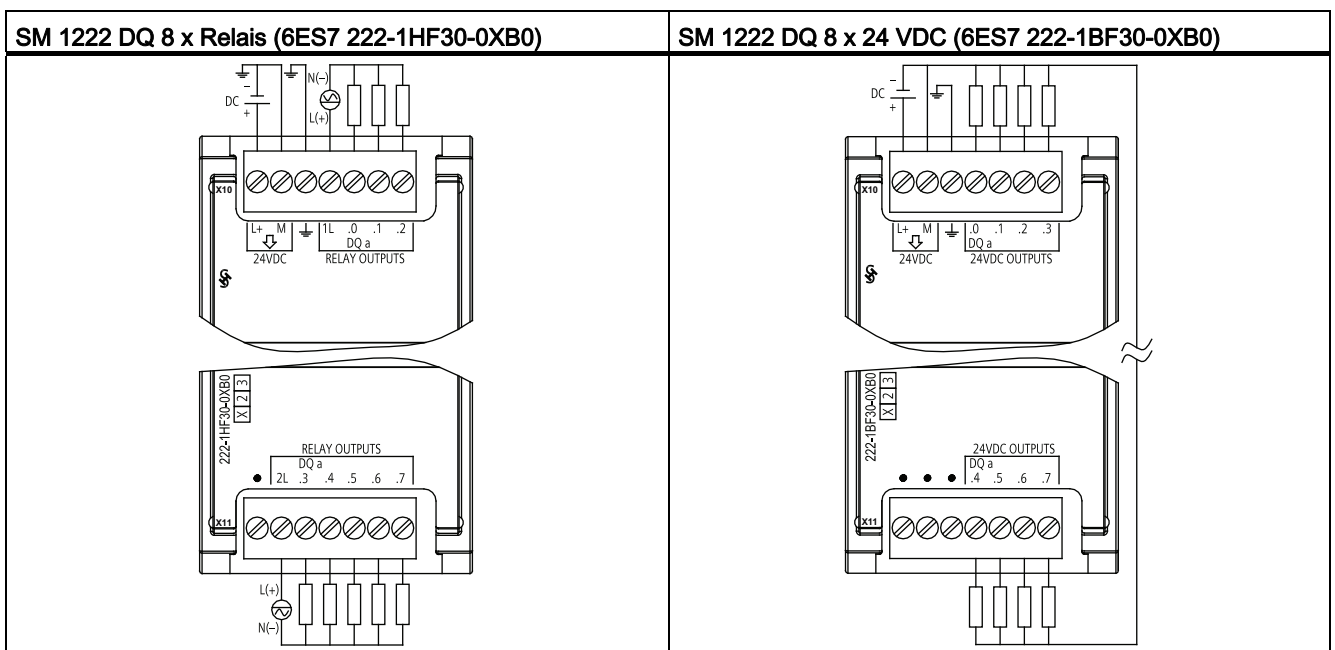


Tableau A- 91 Brochage pour le SM 1222 DQ 8 x Relais (6ES7 222-1HF30-0XB0)

Broche	X10	X11
1	L+ / 24 V-	Pas de connexion
2	M / 24 V-	2L
3	Terre fonctionnelle	DQ a.3
4	1L	DQ a.4
5	DQ a.0	DQ a.5
6	DQ a.1	DQ a.6
7	DQ a.2	DQ a.7

Tableau A- 92 Brochage pour le SM 1222 DQ 8 x 24 VDC (6ES7 222-1BF30-0XB0)

Broche	X10	X11
1	L+ / 24 V-	Pas de connexion
2	M / 24 V-	Pas de connexion
3	Terre fonctionnelle	Pas de connexion
4	DQ a.0	DQ a.4
5	DQ a.1	DQ a.5
6	DQ a.2	DQ a.6
7	DQ a.2	DQ a.7

Tableau A- 93 Schéma de câblage du module SM à 8 sorties TOR relais inverseur

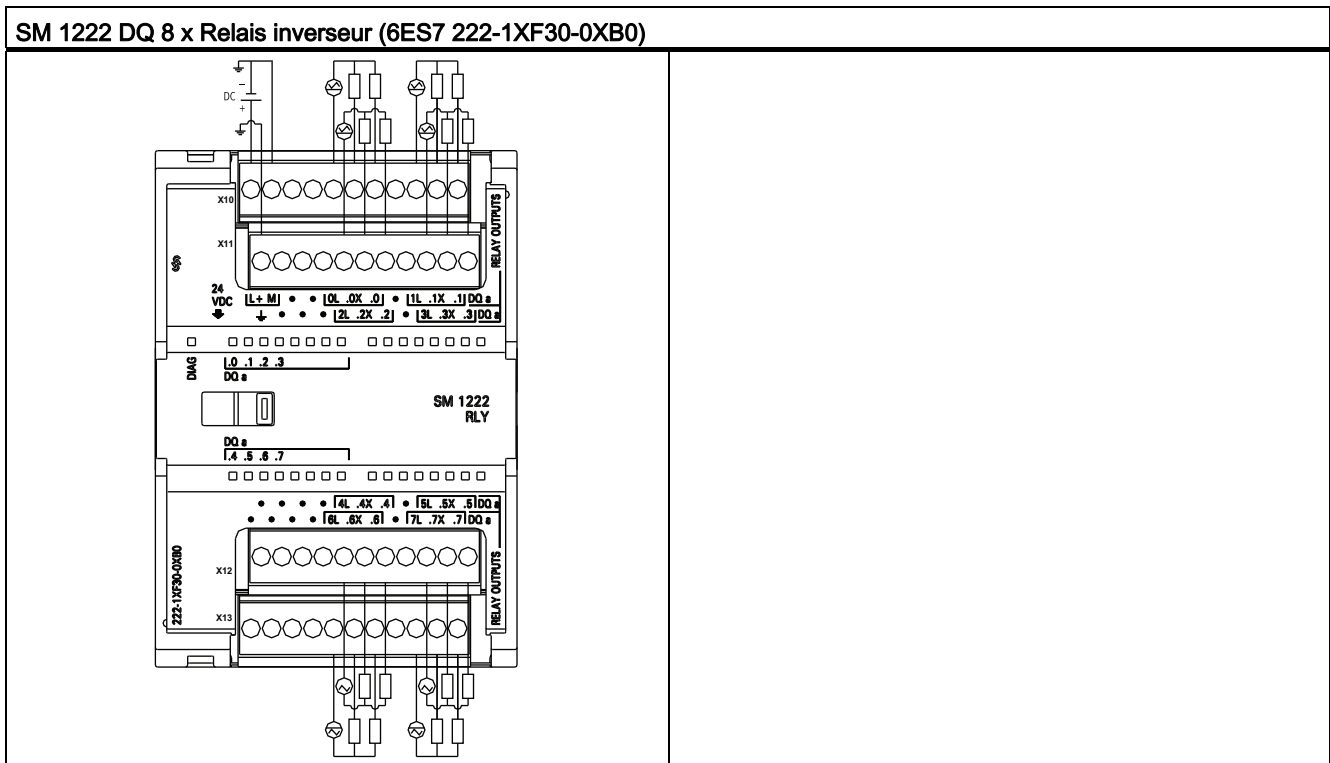


Tableau A- 94 Brochage pour le SM 1222 DQ 8 x Relais inverseur (6ES7 222-1XF30-0XB0)

Broche	X10	X11	X12	X13
1	L+ / 24 V-	Terre fonctionnelle	Pas de connexion	Pas de connexion
2	M / 24 V-	Pas de connexion	Pas de connexion	Pas de connexion
3	Pas de connexion	Pas de connexion	Pas de connexion	Pas de connexion
4	Pas de connexion	Pas de connexion	Pas de connexion	Pas de connexion
5	0L	2L	4L	6L
6	DQ a.0X	DQ a.2X	DQ a.4X	DQ a.6X

Broche	X10	X11	X12	X13
7	DQ a.0	DQ a.2	DQ a.4	DQ a.6
8	Pas de connexion	Pas de connexion	Pas de connexion	Pas de connexion
9	1L	3L	5L	7L
10	DQ a.1X	DQ a.3X	DQ a.5X	DQ a.7X
11	DQ a.1	DQ a.3	DQ a.5	DQ a.7

Tableau A- 95 Schémas de câblage des modules SM à 16 sorties TOR

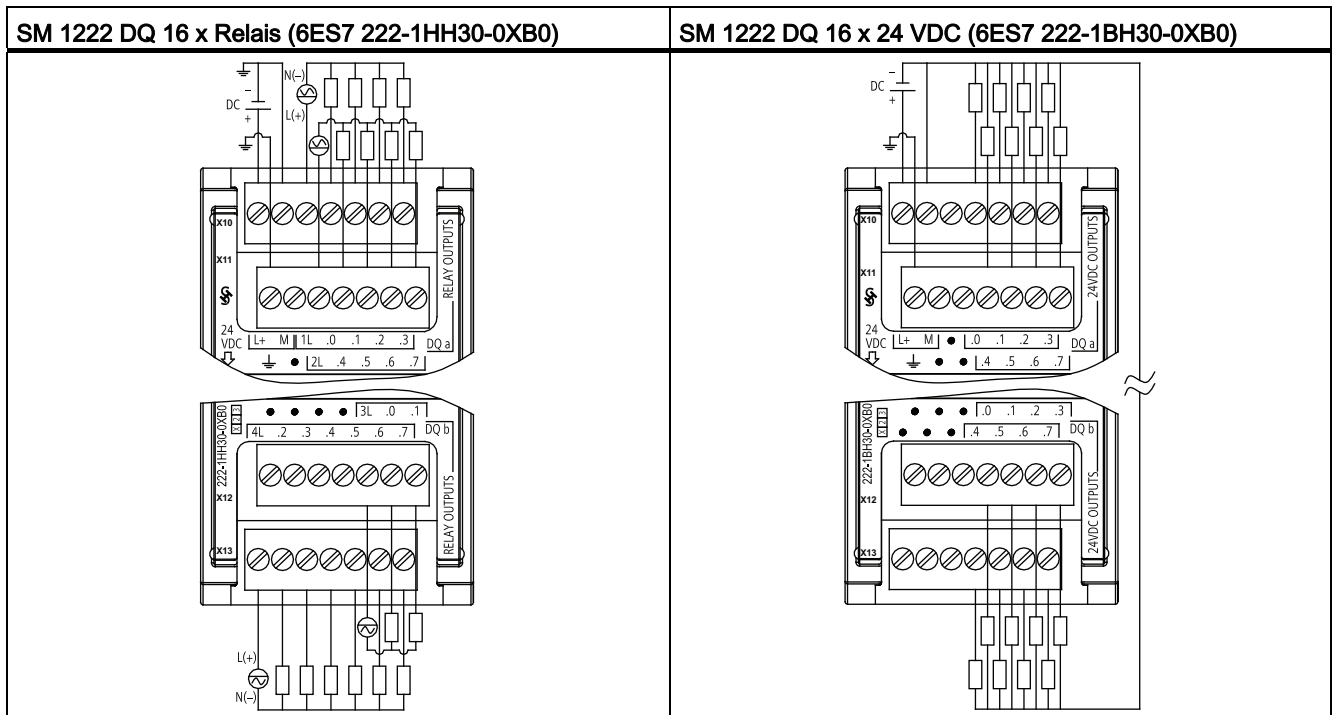


Tableau A- 96 Brochage pour le SM 1222 DQ 16 x Relais (6ES7 222-1HH30-0XB0)

Broche	X10	X11	X12	X13
1	L+ / 24 V-	Terre fonctionnelle	Pas de connexion	4L
2	M / 24 V-	Pas de connexion	Pas de connexion	DQ b.2
3	1L	2L	Pas de connexion	DQ b.3
4	DQ a.0	DQ a.4	Pas de connexion	DQ b.4
5	DQ a.1	DQ a.5	3L	DQ b.5
6	DQ a.2	DQ a.6	DQ b.0	DQ b.6
7	DQ a.3	DQ a.7	DQ b.1	DQ b.7

Tableau A- 97 Brochage pour le SM 1222 DQ 16 x 24 VDC (6ES7 222-1BH30-0XB0)

Broche	X10	X11	X12	X13
1	L+ / 24 V-	Terre fonctionnelle	Pas de connexion	Pas de connexion
2	M / 24 V-	Pas de connexion	Pas de connexion	Pas de connexion
3	Pas de connexion	Pas de connexion	Pas de connexion	Pas de connexion
4	DQ a.0	DQ a.4	DQ b.0	DQ b.4
5	DQ a.1	DQ a.5	DQ b.1	DQ b.5
6	DQ a.2	DQ a.6	DQ b.2	DQ b.6
7	DQ a.3	DQ a.7	DQ b.3	DQ b.7

### A.6.4 Caractéristiques des modules d'entrées/sorties TOR SM 1223 VDC

Tableau A- 98 Caractéristiques générales

Modèle	SM 1223 DI 8 x 24 VDC, DQ 8 x Relais	SM 1223 DI 16 x 24 VDC, DQ 16 x Relais	SM 1223 DI 8 x 24 VDC, DQ 8 x 24 VDC	SM 1223 DI 16 x 24 VDC, DQ 16 x 24 VDC
Numéro de référence	6ES7 223-1PH30-0XB0	6ES7 223-1PL30-0XB0	6ES7 223-1BH30-0XB0	6ES7 223-1BL30-0XB0
Dimensions L x H x P (mm)	45 x 100 x 75	70 x 100 x 75	45 x 100 x 75	70 x 100 x 75
Poids	230 grammes	350 grammes	210 grammes	310 grammes
Dissipation de courant	5,5 W	10 W	2,5 W	4,5 W
Consommation de courant (bus SM)	145 mA	180 mA	145 mA	185 mA
Consommation de courant (24 V-)	4 mA / entrée utilisée 11 mA / bobine de relais utilisée		4 mA/entrée utilisée	

Tableau A- 99 Entrées TOR

Modèle	SM 1223 DI 8 x 24 VDC, DQ 8 x Relais	SM 1223 DI 16 x 24 VDC, DQ 16 x Relais	SM 1223 DI 8 x 24 VDC, DQ 8 x 24 VDC	SM 1223 DI 16 x 24 VDC, DQ 16 x 24 VDC
Nombre d'entrées	8	16	8	16
Type	P/N (CEI type 1 en mode P)			
Tension nominale	24 V- à 4 mA, nominal			
Tension continue admise	30 V- max.			
Tension de choc	35 V- pour 0,5 s			
Signal 1 logique (min.)	15 V- à 2,5 mA			
Signal 0 logique (max.)	5 V- à 1 mA			
Isolation (site à logique)	500 V~ pour 1 minute			
Groupes d'isolation	2	2	2	2
Temps de filtre	0,2, 0,4, 0,8, 1,6, 3,2, 6,4 et 12,8 ms, sélectionnable en groupes de 4			

Modèle	SM 1223 DI 8 x 24 VDC, DQ 8 x Relais	SM 1223 DI 16 x 24 VDC, DQ 16 x Relais	SM 1223 DI 8 x 24 VDC, DQ 8 x 24 VDC	SM 1223 DI 16 x 24 VDC, DQ 16 x 24 VDC
Nombre d'entrées simultanément à 1	8	<ul style="list-style-type: none"> <li>8 (pas d'E/S adjacentes) à 60 °C horizontal ou 50 °C vertical</li> <li>16 à 55 °C horizontal ou 45 °C vertical</li> </ul>	8	16
Longueur de câble (mètres)	500 m blindé, 300 m non blindé			

Tableau A- 100 Sorties TOR

Modèle	SM 1223 DI 8 x 24 VDC, DQ 8 x Relais	SM 1223 DI 16 x 24 VDC, DQ 16 x Relais	SM 1223 DI 8 x 24 VDC, DQ 8 x 24 VDC	SM 1223 DI 16 x 24 VDC, DQ 16 x 24 VDC
Nombre de sorties	8	16	8	16
Type	Relais, contact sec		Transistor à effet de champ MOS (mode P)	
Plage de tension	5 à 30 V- ou 5 à 250 V~		20,4 à 28,8 V-	
Signal 1 logique à courant max.	--		20 V- min.	
Signal 0 logique avec charge 10 kΩ	--		0,1 V- max.	
Courant (max.)	2,0 A		0,5 A	
Charge de lampe	30 W CC / 200 W CA		5 W	
Résistance état activé (contact)	0,2 Ω max. lorsque neuf		0,6 Ω max.	
Courant de fuite par sortie	--		10 µA max.	
Courant de choc	7 A avec contacts fermés		8 A pour 100 ms max.	
Protection contre la surcharge	Non			
Isolation (site à logique)	1500 V~ pour 1 minute (bobine à contact) Aucune (bobine à logique)		500 V~ pour 1 minute	
Résistance d'isolation	100 MΩ min. lorsque neuf		--	
Isolation entre contacts ouverts	750 V~ pour 1 minute		--	
Groupes d'isolation	2	4	1	1
Courant par commun	10 A	8 A	4 A	8 A
Tension de blocage inductive	--		L+ moins 48 V, dissipation 1 W	
Retard de commutation	10 ms max.		50 µs max. de 0 à 1 200 µs max. de 1 à 0	
Fréquence de commutation maximum des relais	1 Hz		--	
Durée d'utilisation mécanique (sans charge)	10 000 000 cycles ouvert/fermé		--	

Caractéristiques techniques

A.6 Modules d'entrées-sorties TOR (SM)

Modèle	SM 1223 DI 8 x 24 VDC, DQ 8 x Relais	SM 1223 DI 16 x 24 VDC, DQ 16 x Relais	SM 1223 DI 8 x 24 VDC, DQ 8 x 24 VDC	SM 1223 DI 16 x 24 VDC, DQ 16 x 24 VDC
Durée d'utilisation des contacts à la charge nominale	100 000 cycles ouvert/fermé		--	
Comportement au passage de MARCHE à ARRÊT	Dernière valeur ou valeur de remplacement (valeur par défaut 0)			
Nombre de sorties simultanément à 1	8	<ul style="list-style-type: none"> <li>8 (pas d'E/S adjacentes) à 60 °C horizontal ou 50 °C vertical</li> <li>16 à 55 °C horizontal ou 45 °C vertical</li> </ul>	8	16
Longueur de câble (mètres)	500 m blindé, 150 m non blindé			

Tableau A- 101 Schémas de câblage des modules SM entrées TOR VDC/sorties relais

SM 1223 DI 8 x 24 VDC, DQ 8 x Relais (6ES7 223-1PH30-0XB0)	SM 1223 DI 16 x 24 VDC, DQ 16 x Relais (6ES7 223-1PL30-0XB0)	Remarques
		<p>① Pour les entrées de type P, connectez "-" à "M" (comme illustré). Pour les entrées de type N, connectez "+" à "M".</p>

Tableau A- 102 Brochage pour le SM 1223 DI 8 x 24 VDC, DQ 8 x Relais (6ES7 223-1PH30-0XB0)

Broche	X10	X11	X12	X13
1	L+ / 24 V-	GND	Pas de connexion	Pas de connexion
2	M / 24 V-	Pas de connexion	Pas de connexion	Pas de connexion
3	1M	2M	1L	2L
4	DI a.0	DI a.4	DQ a.0	DQ a.4
5	DI a.1	DI a.5	DQ a.1	DQ a.5
6	DI a.2	DI a.6	DQ a.2	DQ a.6
7	DI a.3	DI a.7	DQ a.3	DQ a.7

Tableau A- 103 Brochage pour le SM 1223 DI 16 x 24 VDC, DQ 16 x Relais (6ES7 223-1PL30-0XB0)

Broche	X10	X11	X12	X13
1	L+ / 24 V-	GND	1L	3L
2	M / 24 V-	Pas de connexion	DQ a.0	DQ b.0
3	1M	2M	DQ a.1	DQ b.1
4	DI a.0	DI b.0	DQ a.2	DQ b.2
5	DI a.1	DI b.1	DQ a.3	DQ b.3
6	DI a.2	DI b.2	Pas de connexion	Pas de connexion
7	DI a.3	DI b.3	2L	4L
8	DI a.4	DI b.4	DQ a.4	DQ b.4
9	DI a.5	DI b.5	DQ a.5	DQ b.5
10	DI a.6	DI b.6	DQ a.6	DQ b.6
11	DI a.7	DI b.7	DQ a.7	DQ b.7

A.6 Modules d'entrées-sorties TOR (SM)

Tableau A- 104 Schémas de câblage des modules SM entrées/sorties TOR VDC

SM 1223 DI 8 x 24 VDC, DQ 8 x 24 VDC (6ES7 223-1BH30-0XB0)	SM 1223 DI 16 x 24 VDC, DQ 16 x 24 VDC (6ES7 223-1BL30-0XB0)	Remarques
		<p>① Pour les entrées de type P, connectez "-" à "M" (comme illustré). Pour les entrées de type N, connectez "+" à "M".</p>

Tableau A- 105 Brochage pour le SM 1223 DI 8 x 24 VDC, DQ 8 x 24 VDC (6ES7 223-1BH30-0XB0)

Broche	X10	X11	X12	X13
1	L+ / 24 V-	GND	Pas de connexion	Pas de connexion
2	M / 24 V-	Pas de connexion	Pas de connexion	Pas de connexion
3	1M	2M	Pas de connexion	Pas de connexion
4	DI a.0	DI a.4	DQ a.0	DQ a.4
5	DI a.1	DI a.5	DQ a.1	DQ a.5
6	DI a.2	DI a.6	DQ a.2	DQ a.6
7	DI a.3	DI a.7	DQ a.3	DQ a.7

Tableau A- 106 Brochage pour le SM 1223 DI 16 x 24 VDC, DQ 16 x 24 VDC (6ES7 223-1BL30-0XB0)

Broche	X10	X11	X12	X13
1	L+ / 24 V-	GND	Pas de connexion	Pas de connexion
2	M / 24 V-	Pas de connexion	Pas de connexion	Pas de connexion
3	1M	2M	Pas de connexion	Pas de connexion
4	DI a.0	DI b.0	DQ a.0	DQ b.0
5	DI a.1	DI b.1	DQ a.1	DQ b.1



Broche	X10	X11	X12	X13
6	DI a.2	DI b.2	DQ a.2	DQ b.2
7	DI a.3	DI b.3	DQ a.3	DQ b.3
8	DI a.4	DI b.4	DQ a.4	DQ b.4
9	DI a.5	DI b.5	DQ a.5	DQ b.5
10	DI a.6	DI b.6	DQ a.6	DQ b.6
11	DI a.7	DI b.7	DQ a.7	DQ b.7

### A.6.5 Caractéristiques du module d'entrées/sorties TOR SM 1223 VAC

Tableau A- 107 Caractéristiques générales

Modèle	SM 1223 DI 8 x120/230 VAC, DQ 8 x Relais
Numéro de référence	6ES7 223-1QH30-0XB0
Dimensions L x H x P (mm)	45 x 100 x 75 mm
Poids	190 grammes
Dissipation de courant	7,5 W
Consommation de courant (bus SM)	120 mA
Consommation de courant (24 V-)	11 mA par sortie activée

Tableau A- 108 Entrées TOR

Modèle	SM 1223 DI 8 x 120/230 VAC, DQ 8 x Relais
Nombre d'entrées	8
Type	CEI type 1
Tension nominale	120 V~ à 6 mA, 230 V~ à 9 mA
Tension continue admise	264 V~
Tension de choc	--
Signal 1 logique (min.)	79 V~ à 2,5 mA
Signal 0 logique (max.)	20 V~ à 1 mA
Courant de fuite (max.)	1 mA
Isolation (site à logique)	1500 V~ pour 1 minute
Groupes d'isolation <sup>1</sup>	4
Temps de retard d'entrée	Typique : 0,2 à 12,8 ms, personnalisable Maximum : -
Connexion de capteur de proximité à 2 fils (Bero) (max.)	1 mA
Longueur de câble	Non blindé : 300 mètres Blindé : 500 mètres
Nombre d'entrées simultanément à 1	8

<sup>1</sup> Les voies à l'intérieur d'un groupe doivent être de la même phase.

Tableau A- 109 Sorties TOR

Modèle	SM 1223 DI 8 x 120/230 VAC, DQ 8 x Relais
Nombre de sorties	8
Type	Relais, contact sec
Plage de tension	5 à 30 V- ou 5 à 250 V~
Signal 1 logique à courant max.	--
Signal 0 logique avec charge 10 kΩ	--
Courant (max.)	2,0 A
Charge de lampe	30 W CC / 200 W CA
Résistance état activé (contact)	0,2 Ω max. lorsque neuf
Courant de fuite par sortie	--
Courant de choc	7 A avec contacts fermés
Protection contre la surcharge	Non
Isolation (site à logique)	1500 V~ pour 1 minute (bobine à contact) Aucune (bobine à logique)
Résistance d'isolation	100 MΩ min. lorsque neuf
Isolation entre contacts ouverts	750 V~ pour 1 minute
Groupes d'isolation	2
Courant par commun (max.)	10 A
Tension de blocage inductive	--
Retard de commutation (max.)	10 ms
Fréquence de commutation maximum des relais	1 Hz
Durée d'utilisation mécanique (sans charge)	10 000 000 cycles ouvert/fermé
Durée d'utilisation des contacts à la charge nominale	1 000 000 cycles ouvert/fermé
Comportement au passage de MARCHE à ARRET	Dernière valeur ou valeur de remplacement (valeur par défaut 0)
Nombre de sorties simultanément à 1	<ul style="list-style-type: none"> <li>• 4 (pas d'E/S adjacentes) à 60 °C horizontal ou 50 °C vertical</li> <li>• 8 à 55 °C horizontal ou 45 °C vertical</li> </ul>
Longueur de câble (mètres)	500 m blindé, 150 m non blindé

Tableau A- 110 SM 1223 DI 8 x 120/230 VAC, DQ 8 x Relais (6ES7 223-1QH30-0XB0)

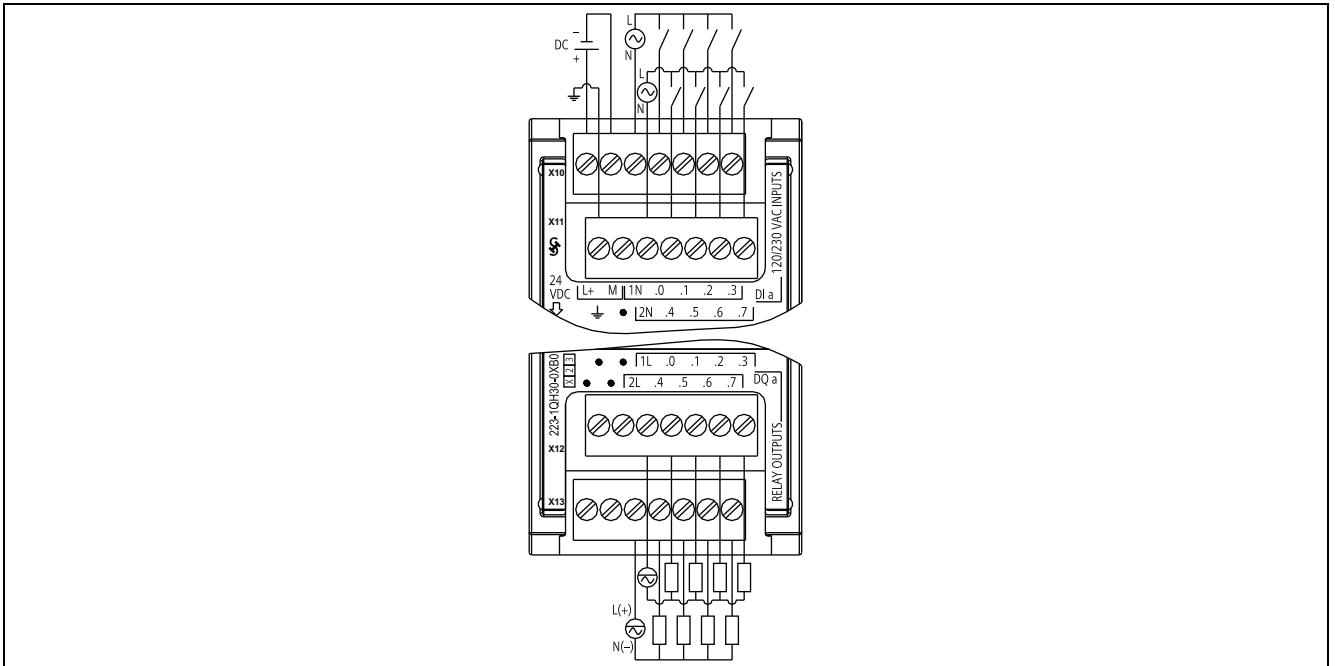


Tableau A- 111 Brochage pour le SM 1223 DI 8 x 120/240 VAC, DQ 8 x Relais (6ES7 223-1QH30-0XB0)

Broche	X10	X11	X12	X13
1	L+ / 24 V-	GND	Pas de connexion	Pas de connexion
2	M / 24 V-	Pas de connexion	Pas de connexion	Pas de connexion
3	1N	2N	1L	2L
4	DI a.0	DI a.4	DQ a.0	DQ a.4
5	DI a.1	DI a.5	DQ a.1	DQ a.5
6	DI a.2	DI a.6	DQ a.2	DQ a.6
7	DI a.3	DI a.7	DQ a.3	DQ a.7

## A.7 Modules d'entrées-sorties analogiques (SM)

### A.7.1 Caractéristiques des modules d'entrées analogiques SM 1231

Tableau A- 112 Caractéristiques générales

Modèle	SM 1231 AI 4 x 13 bits	SM 1231 AI 8 x 13 bits	SM 1231 AI 4 x 16 bits
Numéro de référence	6ES7 231-4HD30-0XB0	6ES7 231-4HF30-0XB0	6ES7 231-5ND30-0XB0
Dimensions L x H x P (mm)	45 x 100 x 75	45 x 100 x 75	45 x 100 x 75
Poids	180 grammes	180 grammes	180 grammes
Dissipation de courant	2,2 W	2,3 W	2,0 W
Consommation de courant (bus SM)	80 mA	90 mA	80 mA
Consommation de courant (24 V-)	45 mA	45 mA	65 mA

Tableau A- 113 Entrées analogiques

Modèle	SM 1231 AI 4 x 13 bits	SM 1231 AI 8 x 13 bits	SM 1231 AI 4 x 16 bits
Nombre d'entrées	4	8	4
Type	Tension ou courant (différentiel) : sélectionnable en groupes de 2		Tension ou courant (différentiel)
Plage	±10 V, ±5 V, ±2,5 V ou 0 à 20 mA		±10 V, ±5 V, ±2,5 V, ±1,25 V, 0 à 20 mA ou 4 mA à 20 mA
Plage pleine échelle (mot de données)	-27 648 à 27 648 tension / 0 à 27 648 courant		
Plage de dépassement haut/bas (mot de données) Voir la partie sur les plages des entrées analogiques pour la tension et le courant (Page 822).	Tension : 32 511 à 27 649 / -27 649 à -32 512 Courant : 32 511 à 27 649 / 0 à -4864		Tension : 32 511 à 27 649 / -27 649 à -32 512 Courant : 0-20 mA : 32 511 à 27 649 / 0 à -4864 ; 4-20 mA : 32 511 à 27 649 / -1 à -4864
Débordement haut / bas (mot de données) Voir la partie sur les plages des entrées pour la tension et le courant (Page 822).	Tension : 32 767 à 32 512 / -32 513 à -32 768 Courant : 32 767 à 32 512 / -4865 à -32 768		Tension : 32 767 à 32 512 / -32 513 à -32 768 Courant : 0 à 20 mA 32 767 à 32 512 / -4865 à -32 768 4-20 mA 32 767 à 32 512 / -4865 à -32 768
Résolution1	12 bits + bit de signe		15 bits + bit de signe
Tension/courant de tenue maximum	±35 V / ±40 mA		

Modèle	SM 1231 AI 4 x 13 bits	SM 1231 AI 8 x 13 bits	SM 1231 AI 4 x 16 bits
Lissage	Aucun, faible, moyen ou fort Voir la partie sur les temps de réponse indicielle (Page 821).		
Réjection des bruits	400, 60, 50 ou 10 Hz Voir la partie sur les taux d'échantillonnage (Page 821).		
Impédance d'entrée	≥ 9 MΩ (tension) / 280 Ω (courant)		≥ 1 MΩ (tension) / <315 Ω, >280 Ω (courant)
Isolation	Aucune		
Site à logique			500 V~
Logique à 24 V CC			500 V~
Site à 24 V CC			500 V~
Voie à voie			aucune
Précision (25°C / -20 à 60°C)	±0,1% / ±0,2% de la pleine échelle		±0,1% / ±0,3% de la pleine échelle
Principe de mesure	Conversion de valeur effective		
Réjection en mode commun	40 dB, CC pour 60 Hz		
Plage de signal opérationnelle <sup>1</sup>	La tension signal plus mode commun doit être inférieure à +12 V et supérieure à -12 V.		
Longueur de câble (mètres)	100 m, torsadé et blindé		

<sup>1</sup> Des tensions en dehors de la plage opérationnelle appliquées à une voie peuvent provoquer des interférences sur les autres voies.

Tableau A- 114 Diagnostic

Modèle	SM 1231 AI 4 x 13 bits	SM 1231 AI 8 x 13 bits	SM 1231 AI 4 x 16 bits
Débordement haut/bas	Oui <sup>1</sup>	Oui	Oui
Basse tension 24 V-	Oui	Oui	Oui
Fil ouvert	--	--	Plage 4 à 20 mA uniquement (si l'entrée est en dessous de -4164 ; 1,0 mA)

<sup>1</sup> Pour le SM 1231 AI 4 x 13 bits : Si une tension supérieure à +30 V- ou inférieure à -15 V- est appliquée à l'entrée, la valeur résultante sera inconnue et le débordement haut ou bas correspondant pourra ne pas être activé.

Tableau A- 115 Schémas de câblage des modules SM d'entrées analogiques

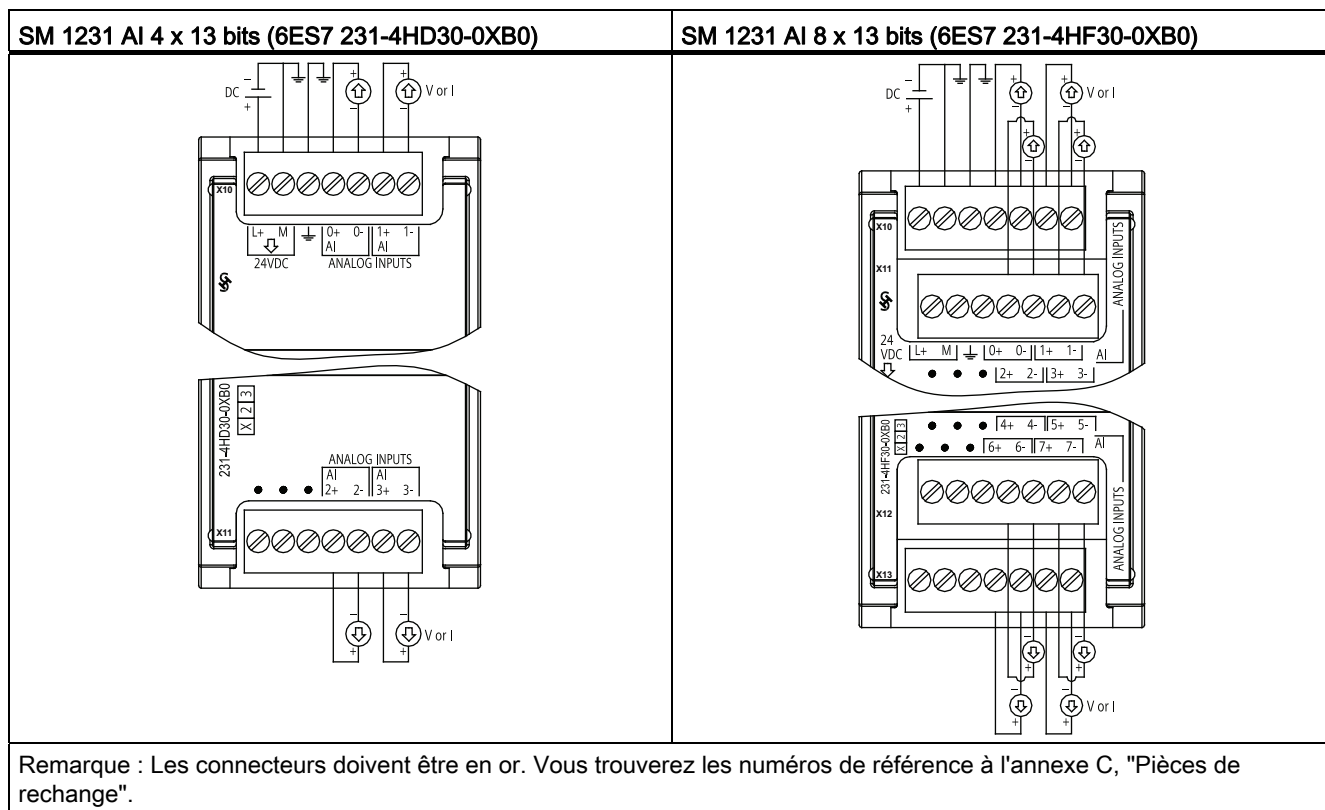


Tableau A- 116 Brochage pour le SM 1231 AI 4 x 13 bits (6ES7 231-4HD30-0XB0)

Broche	X10 (or)	X11 (or)
1	L+ / 24 V-	Pas de connexion
2	M / 24 V-	Pas de connexion
3	GND	Pas de connexion
4	AI 0+	AI 2+
5	AI 0-	AI 2-
6	AI 1+	AI 3+
7	AI 1-	AI 3-

Tableau A- 117 Brochage pour le SM 1231 AI 8 x 13 bits (6ES7 231-4HF30-0XB0)

Broche	X10 (or)	X11 (or)	X12 (or)	X13 (or)
1	L+ / 24 V-	Pas de connexion	Pas de connexion	Pas de connexion
2	M / 24 V-	Pas de connexion	Pas de connexion	Pas de connexion
3	GND	Pas de connexion	Pas de connexion	Pas de connexion
4	AI 0+	AI 2+	AI 4+	AI 6+
5	AI 0-	AI 2-	AI 4-	AI 6-
6	AI 1+	AI 3+	AI 5+	AI 7+
7	AI 1-	AI 3-	AI 5-	AI 7-

Tableau A- 118 Schéma de câblage du SM d'entrées analogiques

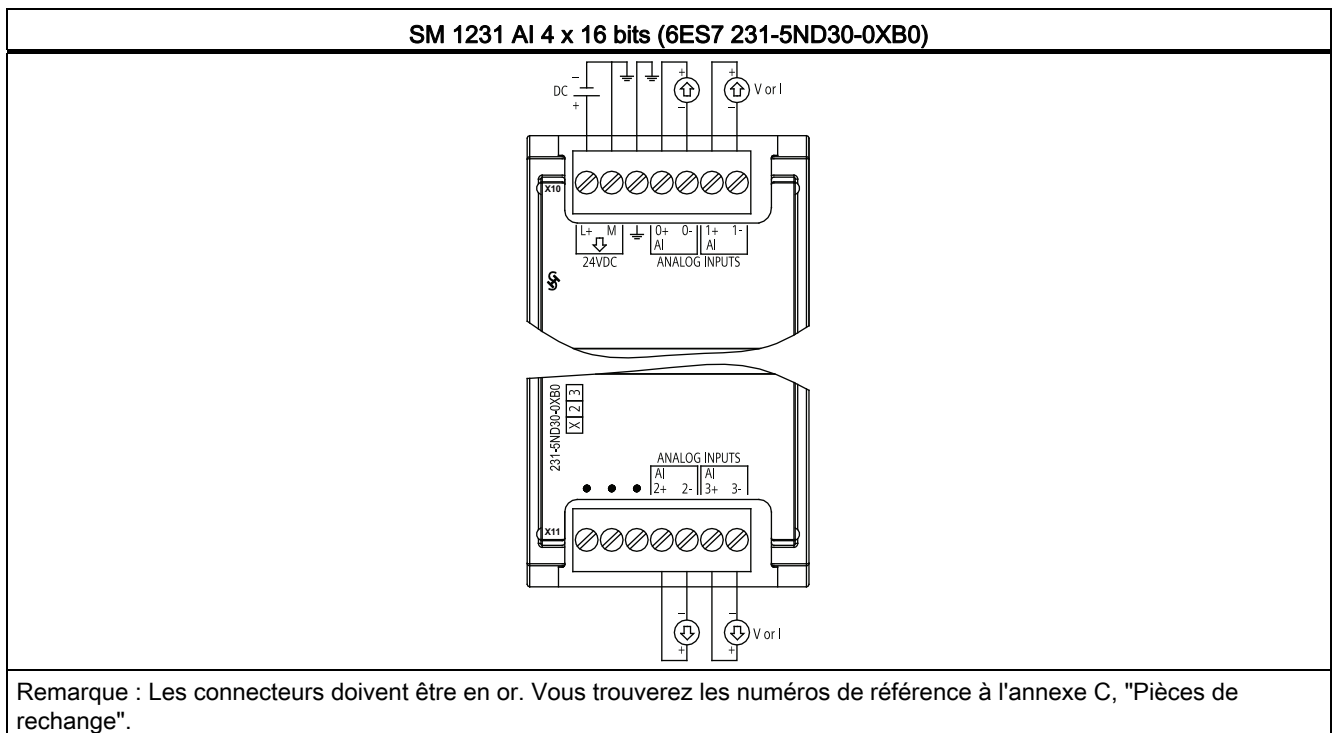


Tableau A- 119 Brochage pour le SM 1231 AI 4 x 16 bits (6ES7 231-5ND30-0XB0)

Broche	X10 (or)	X11 (or)
1	L+ / 24 V-	Pas de connexion
2	M / 24 V-	Pas de connexion
3	GND	Pas de connexion
4	AI 0+	AI 2+
5	AI 0-	AI 2-
6	AI 1+	AI 3+
7	AI 1-	AI 3-

**Remarque**

Les entrées analogiques inutilisées doivent être court-circuitées.

Lorsque les entrées sont configurées pour le mode "courant", aucun courant ne s'écoule à travers l'entrée à moins que vous ne fournissiez une alimentation externe au module.

**A.7.2 Caractéristiques des modules de sorties analogiques SM 1232**

Tableau A- 120 Caractéristiques générales

Caractéristiques techniques	SM 1232 AQ 2 x 14 bits	SM 1232 AQ 4 x 14 bits
Numéro de référence	6ES7 232-4HB30-0XB0	6ES7 232-4HD30-0XB0
Dimensions L x H x P (mm)	45 x 100 x 75	45 x 100 x 75
Poids	180 grammes	180 grammes
Dissipation de courant	1,8 W	2,0 W
Consommation de courant (bus SM)	80 mA	80 mA
Consommation de courant (24 V-)	45 mA (sans charge)	45 mA (sans charge)

Tableau A- 121 Sorties analogiques

Caractéristiques techniques	SM 1232 AQ 2 x 14 bits	SM 1232 AQ 4 x 14 bits
Nombre de sorties	2	4
Type	Tension ou courant	Tension ou courant
Plage	±10 V ou 0 à 20 mA	±10 V ou 0 à 20 mA
Résolution	Tension : 14 bits Courant : 13 bits	Tension : 14 bits Courant : 13 bits
Plage pleine échelle (mot de données)	Tension : -27 648 à 27 648 ; Courant : 0 à 27 648 Voir les plages des sorties pour la tension et le courant (Page 823).	
Précision (25°C / -20 à 60°C)	±0,3% / ±0,6% de la pleine échelle	
Temps d'établissement (95% de la nouvelle valeur)	Tension : 300 µs (R), 750 µs (1 µF) Courant : 600 µs (1 mH), 2 ms (10 mH)	
Impédance de charge	Tension : ≥ 1000 Ω Courant : ≤ 600 Ω	
Comportement au passage de MARCHE à ARRET	Dernière valeur ou valeur de remplacement (valeur par défaut 0)	
Isolation (site à logique)	aucune	
Longueur de câble (mètres)	100 m, torsadé et blindé	



Tableau A- 122 Diagnostic

Caractéristiques techniques	SM 1232 AQ 2 x 14 bits	SM 1232 AQ 4 x 14 bits
Débordement haut/bas	Oui	Oui
Court-circuit à la terre (mode tension uniquement)	Oui	Oui
Rupture de fil (mode courant uniquement)	Oui	Oui
Basse tension 24 V-	Oui	Oui

Tableau A- 123 Schémas de câblage des modules SM de sorties analogiques

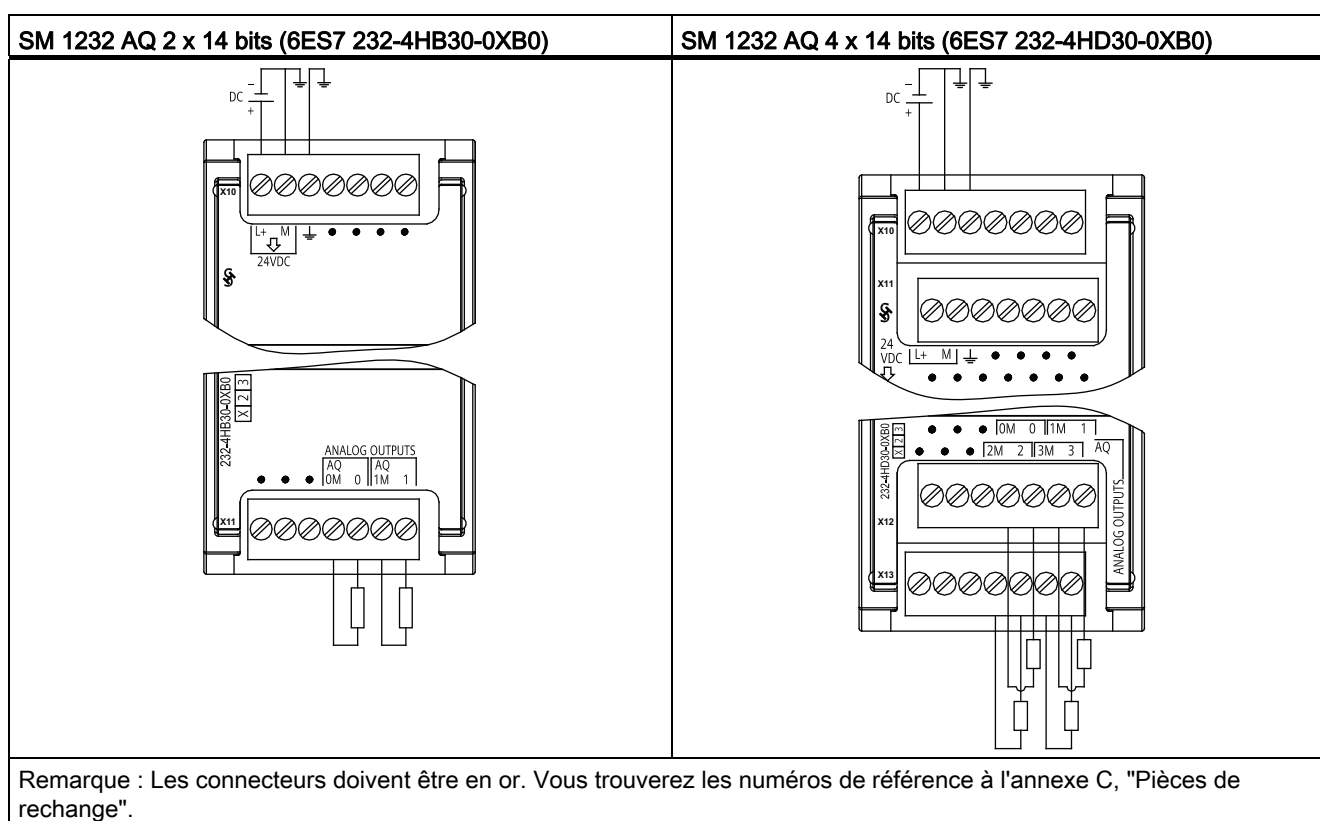


Tableau A- 124 Brochage pour le SM 1232 AQ 2 x 14 bits (6ES7 232-4HB30-0XB0)

Broche	X10 (or)	X11 (or)
1	L+ / 24 V-	Pas de connexion
2	M / 24 V-	Pas de connexion
3	GND	Pas de connexion
4	Pas de connexion	AQ 0M
5	Pas de connexion	AQ 0

A.7 Modules d'entrées-sorties analogiques (SM)

Broche	X10 (or)	X11 (or)
6	Pas de connexion	AQ 1M
7	Pas de connexion	AQ 1

Tableau A- 125 Brochage pour le SM 1232 AQ 4 x 14 bits (6ES7 232-4HD30-0XB0)

Broche	X10 (or)	X11 (or)	X12 (or)	X13 (or)
1	L+ / 24 V-	Pas de connexion	Pas de connexion	Pas de connexion
2	M / 24 V-	Pas de connexion	Pas de connexion	Pas de connexion
3	GND	Pas de connexion	Pas de connexion	Pas de connexion
4	Pas de connexion	Pas de connexion	AQ 0M	AQ 2M
5	Pas de connexion	Pas de connexion	AQ 0	AQ 2
6	Pas de connexion	Pas de connexion	AQ 1M	AQ 3M
7	Pas de connexion	Pas de connexion	AQ 1	AIQ 3

**A.7.3 Caractéristiques du module d'entrées/sorties analogiques SM 1234**

Tableau A- 126 Caractéristiques générales

Caractéristiques techniques	SM 1234 AI 4 x 13 bits / AQ 2 x 14 bits
Numéro de référence	6ES7 234-4HE30-0XB0
Dimensions L x H x P (mm)	45 x 100 x 75
Poids	220 grammes
Dissipation de courant	2,4 W
Consommation de courant (bus SM)	80 mA
Consommation de courant (24 V-)	60 mA (sans charge)

Tableau A- 127 Entrées analogiques

Modèle	SM 1234 AI 4 x 13 bits / AQ 2 x 14 bits
Nombre d'entrées	4
Type	Tension ou courant (différentiel) : sélectionnable en groupes de 2
Plage	±10 V, ±5 V, ±2,5 V ou 0 à 20 mA
Plage pleine échelle (mot de données)	-27 648 à 27 648
Plage de dépassement haut/bas (mot de données)	Tension : 32 511 à 27 649 / -27 649 à -32 512 Courant : 32 511 à 27 649 / 0 à -4864 Voir la partie sur les plages des entrées pour la tension et le courant (Page 822).

Modèle	SM 1234 AI 4 x 13 bits / AQ 2 x 14 bits
Débordement haut / bas (mot de données)	Tension : 32 767 à 32 512 / -32 513 à -32 768 Courant : 32 767 à 32 512 / -4865 à -32 768 Voir la partie sur les plages des entrées pour la tension et le courant (Page 822).
Résolution	12 bits + bit de signe
Tension/courant de tenue maximum	±35 V / ±40 mA
Lissage	Aucun, faible, moyen ou fort Voir la partie sur les temps de réponse indicielle (Page 821).
Réjection des bruits	400, 60, 50 ou 10 Hz Voir la partie sur les taux d'échantillonnage (Page 821).
Impédance d'entrée	≥ 9 MΩ (tension) / 280 Ω (courant)
Isolation (site à logique)	Aucune
Précision (25 °C / -20 à 60°C)	±0,1% / ±0,2% de la pleine échelle
Temps de conversion analogique-numérique	625 μs (réjection 400 Hz)
Réjection en mode commun	40 dB, CC pour 60 Hz
Plage de signal opérationnelle <sup>1</sup>	La tension signal plus mode commun doit être inférieure à +12 V et supérieure à -12 V.
Longueur de câble (mètres)	100 m, torsadé et blindé

<sup>1</sup> Des tensions en dehors de la plage opérationnelle appliquées à une voie peuvent provoquer des interférences sur les autres voies.

Tableau A- 128 Sorties analogiques

Caractéristiques techniques	SM 1234 AI 4 x 13 bits / AQ 2 x 14 bits
Nombre de sorties	2
Type	Tension ou courant
Plage	±10 V ou 0 à 20 mA
Résolution	Tension : 14 bits ; courant : 13 bits
Plage pleine échelle (mot de données)	Tension : -27 648 à 27 648 ; courant : 0 à 27 648 Voir la partie sur les plages des sorties pour la tension et le courant (Page 823).
Précision (25°C / -20 à 60°C)	±0,3% / ±0,6% de la pleine échelle
Temps d'établissement (95% de la nouvelle valeur)	Tension : 300 μs (R), 750 μs (1 μF) Courant : 600 μs (1 mH), 2 ms (10 mH)
Impédance de charge	Tension : ≥ 1000 Ω Courant : ≤ 600 Ω
Comportement au passage de MARCHE à ARRET	Dernière valeur ou valeur de remplacement (valeur par défaut 0)
Isolation (site à logique)	aucune
Longueur de câble (mètres)	100 m, torsadé et blindé

A.7 Modules d'entrées-sorties analogiques (SM)

Tableau A- 129 Diagnostic

Modèle	SM 1234 AI 4 x 13 bits / AQ 2 x 14 bits
Débordement haut/bas	Oui <sup>1</sup>
Court-circuit à la terre (mode tension uniquement)	Oui sur les sorties
Rupture de fil (mode courant uniquement)	Oui sur les sorties
Basse tension 24 V-	Oui

<sup>1</sup> Si une tension supérieure à +30 V- ou inférieure à -15 V- est appliquée à l'entrée, la valeur résultante sera inconnue et le débordement haut ou bas correspondant pourra ne pas être activé.

Tableau A- 130 Schémas de câblage du SM d'entrées/sorties analogiques

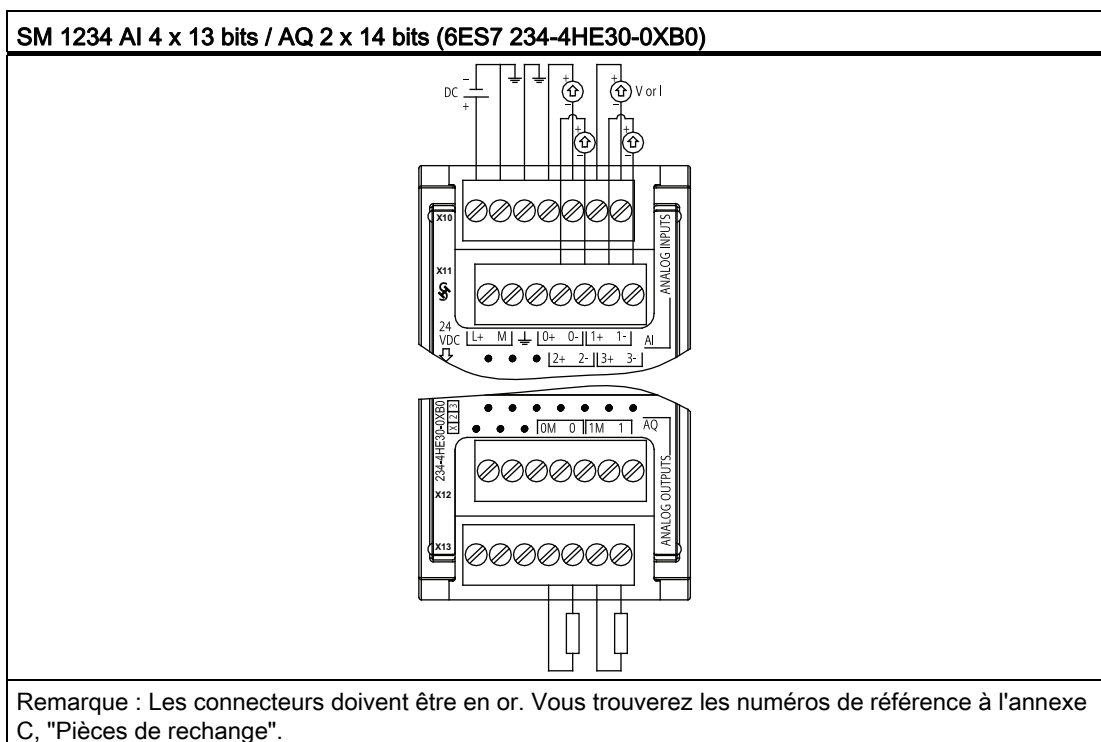


Tableau A- 131 Brochage pour le SM 1234 AI 4 x 13 bits / AQ 2 x 14 bits (6ES7 234-4HE30-0XB0)

Broche	X10 (or)	X11 (or)	X12 (or)	X13 (or)
1	L+ / 24 V-	Pas de connexion	Pas de connexion	Pas de connexion
2	M / 24 V-	Pas de connexion	Pas de connexion	Pas de connexion
3	GND	Pas de connexion	Pas de connexion	Pas de connexion
4	AI 0+	AI 2+	Pas de connexion	AQ 0M
5	AI 0-	AI 2-	Pas de connexion	AQ 0

Broche	X10 (or)	X11 (or)	X12 (or)	X13 (or)
6	AI 1+	AI 3+	Pas de connexion	AQ 1M
7	AI 1-	AI 3-	Pas de connexion	AIQ 1

**Remarque**

Les entrées analogiques inutilisées doivent être court-circuitées.

Lorsque les entrées sont configurées pour le mode "courant", aucun courant ne s'écoule à travers l'entrée à moins que vous ne fournissiez une alimentation externe au module.

### A.7.4 Réponse indicielle des entrées analogiques

Tableau A- 132 Réponse indicielle (ms), 0 à pleine échelle mesuré à 95%

Sélection de lissage (moyennage d'échantillon)	Réduction des bruits/fréquence de réjection (sélection de temps d'intégration)			
	400 Hz (2,5 ms)	60 Hz (16,6 ms)	50 Hz (20 ms)	10 Hz (100 ms)
Aucun (1 cycle) : Pas de moyennage	4 ms	18 ms	22 ms	100 ms
Faible (4 cycles) : 4 échantillons	9 ms	52 ms	63 ms	320 ms
Moyen (16 cycles) : 16 échantillons	32 ms	203 ms	241 ms	1200 ms
Fort (32 cycles) : 32 échantillons	61 ms	400 ms	483 ms	2410 ms
<b>Temps d'échantillonnage</b>				
• 4 AI x 13 bits	• 0,625 ms	• 4,17 ms	• 5 ms	• 25 ms
• 8 AI x 13 bits	• 1,25 ms	• 4,17 ms	• 5 ms	• 25 ms
• 4 AI x 16 bits	• 0,417 ms	• 0,397 ms	• 0,400 ms	• 0,400 ms

### A.7.5 Temps d'échantillonnage et temps d'actualisation pour les entrées analogiques

Tableau A- 133 Temps d'échantillonnage et temps d'actualisation

Fréquence de réjection (temps d'intégration)	Temps d'échantillonnage	Temps d'actualisation du module pour toutes les voies	
		SM 4 voies	SM 8 voies
400 Hz (2,5 ms)	<ul style="list-style-type: none"> <li>SM 4 voies : 0,625 ms</li> <li>SM 8 voies : 1,250 ms</li> </ul>	0,625 ms	1,250 ms
60 Hz (16,6 ms)	4,170 ms	4,17 ms	4,17 ms
50 Hz (20 ms)	5,000 ms	5 ms	5 ms
10 Hz (100 ms)	25,000 ms	25 ms	25 ms

### A.7.6 Plages de mesure des entrées analogiques pour la tension

Tableau A- 134 Représentation des entrées analogiques pour la tension

Technologie		Plage de mesure de tension						
Décimal	Hexadécimal	± 10 V	± 5 V	± 2,5 V	± 1,25V	0 à 10 V		
32767	7FFF	11,851 V	5,926 V	2,963 V	1,481 V	Débordement haut	11,851 V	Débordement haut
32512	7F00							
32511	7EFF	11,759 V	5,879 V	2,940 V	1,470 V	Plage de dépassement haut	11,759 V	Plage de dépassement haut
27649	6C01							
27648	6C00	10 V	5 V	2,5 V	1,250 V	Plage nominale	10 V	Plage nominale
20736	5100	7,5 V	3,75 V	1,875 V	0,938 V		7,5 V	
1	1	361,7 µV	180,8 µV	90,4 µV	45,2 µV		361,7 µV	
0	0	0 V	0 V	0 V	0 V		0 V	
-1	FFFF						Valeurs négatives impossibles	
-20736	AF00	-7,5 V	-3,75 V	-1,875 V	-0,938 V			
-27648	9400	-10 V	-5 V	-2,5 V	-1,250 V			
-27649	93FF							
-32512	8100	-11,759 V	-5,879 V	-2,940 V	-1,470 V	Plage de dépassement bas		
-32513	80FF					Débordement bas		
-32768	8000	-11,851 V	-5,926 V	-2,963 V	-1,481 V			

### A.7.7 Plages de mesure des entrées analogiques pour le courant

Tableau A- 135 Représentation des entrées analogiques pour le courant

Technologie		Plage de mesure de courant		
Décimal	Hexadécimal	0 mA à 20 mA	4 mA à 20 mA	
32767	7FFF	23,70 mA	22,96 mA	Débordement haut
32512	7F00			
32511	7EFF	23,52 mA	22,81 mA	Plage de dépassement haut
27649	6C01			
27648	6C00	20 mA	20 mA	Plage nominale
20736	5100	15 mA	16 mA	
1	1	723,4 nA	4 mA + 578,7 nA	
0	0	0 mA	4 mA	
-1	FFFF			
-4864	ED00	-3,52 mA	1,185 mA	Plage de dépassement bas
-4865	ECFF			
-32768	8000			

### A.7.8 Plages de mesures de sortie (AQ) pour tension et courant (SB et SM)

Tableau A- 136 Représentation des sorties analogiques pour la tension

Technologie		Plage de sortie de tension	
Décimal	Hexadécimal	± 10 V	
32767	7FFF	Voir note 1	Débordement haut
32512	7F00	Voir note 1	
32511	7EFF	11,76 V	Plage de dépassement haut
27649	6C01		
27648	6C00	10 V	Plage nominale
20736	5100	7,5 V	
1	1	361,7 µV	
0	0	0 V	
-1	FFFF	-361,7 µV	
-20736	AF00	-7,5 V	
-27648	9400	-10 V	
-27649	93FF		
-32512	8100	-11,76 V	Plage de dépassement bas
-32513	80FF	Voir note 1	
-32768	8000	Voir note 1	

<sup>1</sup> En cas de dépassement haut ou bas, les sorties analogiques se comporteront selon les propriétés de configuration de l'appareil établies pour le module d'entrées-sorties analogique. Pour le paramètre "Réaction à l'arrêt de la CPU", sélectionnez : Appliquer valeur de remplacement ou Conserver dernière valeur.

Tableau A- 137 Représentation des sorties analogiques pour le courant

Technologie		Plage de sortie de courant	
Décimal	Hexadécimal	0 mA à 20 mA	
32767	7FFF	Voir note 1	Débordement haut
32512	7F00	Voir note 1	
32511	7EFF	23,52 mA	Plage de dépassement haut
27649	6C01		
27648	6C00	20 mA	Plage nominale
20736	5100	15 mA	
1	1	723,4 nA	
0	0	0 mA	

<sup>1</sup> En cas de dépassement haut ou bas, les sorties analogiques se comporteront selon les propriétés de configuration de l'appareil établies pour le module d'entrées-sorties analogique. Pour le paramètre "Réaction à l'arrêt de la CPU", sélectionnez : Appliquer valeur de remplacement ou Conserver dernière valeur.

## A.8 Modules d'entrées-sorties (SM) Thermocouple et RTD

### A.8.1 SM 1231 Thermocouple

Tableau A- 138 Caractéristiques générales

Modèle	SM 1231 AI 4 x 16 bits TC	SM 1231 AI 8 x 16 bits TC
Numéro de référence	6ES7 231-5QD30-0XB0	6ES7 231-5PD30-0XB0
Dimensions L x H x P (mm)	45 x 100 x 75	45 x 100 x 75
Poids	180 grammes	190 grammes
Dissipation de courant	1,5 W	1,5 W
Consommation de courant (bus SM)	80 mA	80 mA
Consommation de courant (24 V-) <sup>1</sup>	40 mA	40 mA

<sup>1</sup> 20,4 à 28,8 V- (classe 2, puissance limitée, ou alimentation de capteur de l'API)

Tableau A- 139 Entrées analogiques

Modèle	SM 1231 AI 4 x 16 bits TC	SM 1231 AI 8 x 16 bits TC
Nombre d'entrées	4	8
Plage	Voir le tableau de sélection pour thermocouples (Page 828).	Voir le tableau de sélection pour thermocouples (Page 828).
Plage nominale (mot de données)		
Dépassement haut/bas (mot de données)		
Débordement haut / bas (mot de données)		
Résolution		
Température	0,1° C/0,1° F	0,1° C/0,1° F
Tension	15 bits plus signe	15 bits plus signe
Tension de tenue maximum	± 35 V	± 35 V
Réjection des bruits	85 dB pour le réglage de filtre sélectionné (10 Hz, 50 Hz, 60 Hz ou 400 Hz)	85 dB pour le réglage de filtre sélectionné (10 Hz, 50 Hz, 60 Hz ou 400 Hz)
Réjection en mode commun	> 120 dB à 120 V~	> 120 dB à 120 V~
Impédance	≥ 10 MΩ	≥ 10 MΩ
Isolation		
Site à logique	500 V~	500 V~
Site à 24 V-	500 V~	500 V~
24 V- à logique	500 V~	500 V~
Voie à voie	120 V~	120 V~
Précision	Voir le tableau de sélection pour thermocouples (Page 828).	Voir le tableau de sélection pour thermocouples (Page 828).
Répétabilité	± 0,05% pleine échelle	± 0,05% pleine échelle
Principe de mesure	Intégration	Intégration
Temps d'actualisation du module	Voir le tableau de sélection pour la réduction des bruits (Page 828).	Voir le tableau de sélection pour la réduction des bruits (Page 828).



Modèle	SM 1231 AI 4 x 16 bits TC	SM 1231 AI 8 x 16 bits TC
Erreur de soudure froide	± 1,5°C	± 1,5°C
Longueur de câble (mètres)	max. 100 mètres au capteur	max. 100 mètres au capteur
Résistance de fil	100 Ω max.	100 Ω max.

Tableau A- 140 Diagnostic

Modèle	SM 1231 AI 4 x 16 bits TC	SM 1231 AI 8 x 16 bits TC
Débordement haut/bas <sup>1</sup>	Oui	Oui
Rupture de fil (mode courant uniquement) <sup>2</sup>	Oui	Oui
Basse tension 24 V- <sup>1</sup>	Oui	Oui

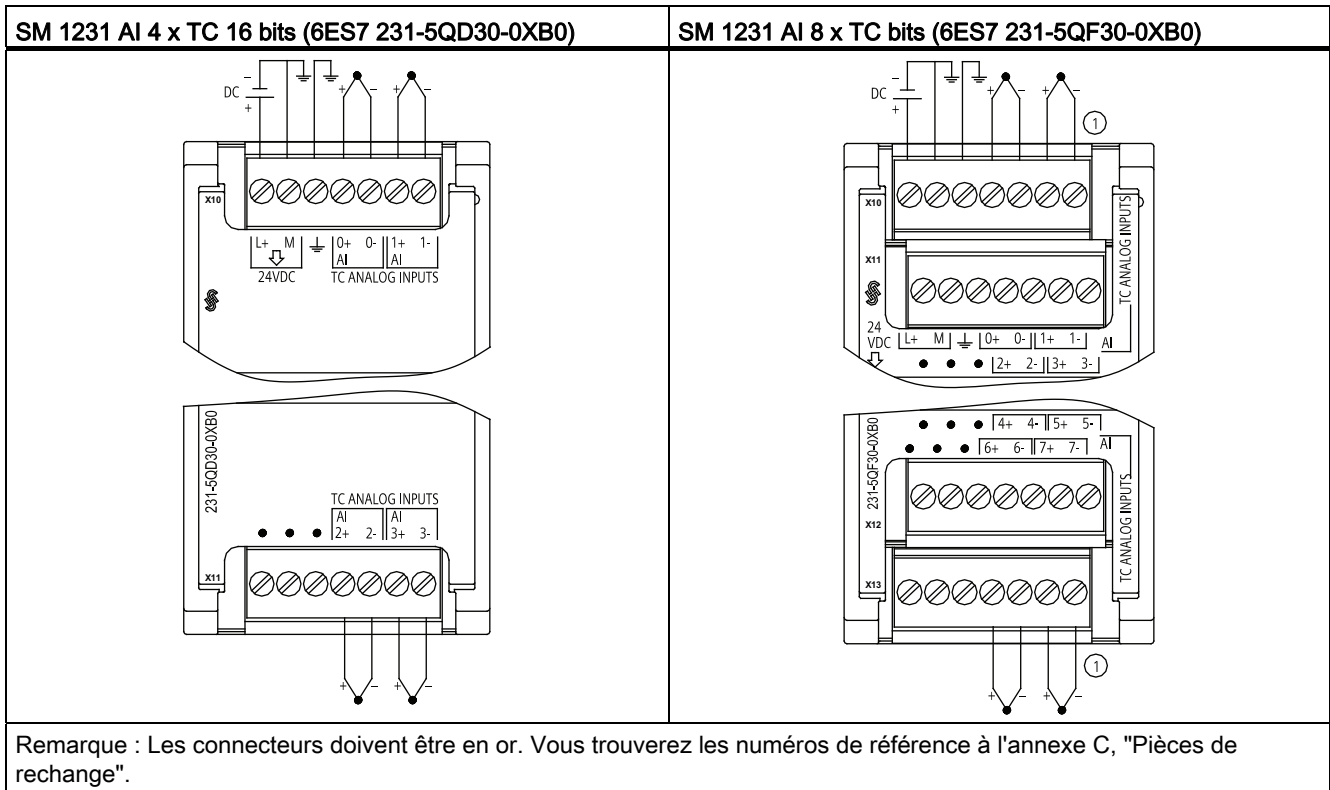
<sup>1</sup> Les informations d'alarmes de diagnostic débordement haut, débordement bas et basse tension sont notifiées dans les valeurs de données analogiques même si les alarmes sont désactivées dans la configuration du module.

<sup>2</sup> Lorsque l'alarme de rupture de fil est désactivée et qu'un fil est ouvert dans le câblage du capteur, le module peut émettre des valeurs aléatoires.

Le module d'entrées analogiques SM 1231 Thermocouple (TC) mesure la valeur de tension connectée aux entrées du module. Le type de mesure de température peut être soit "Thermocouple", soit "Tension".

- "Thermocouple" : La valeur sera notifiée en degrés multipliés par dix (par exemple, 25,3 degrés seront indiqués sous la forme 253 en décimal).
- "Tension" : La valeur pleine échelle de la plage nominale sera 27648 en décimal.

Tableau A- 141 Schémas de câblage pour les SM Thermocouple



① Pour plus de clarté, les TC 2, 3, 4 et 5 ne sont pas connectés sur l'illustration.

Tableau A- 142 Brochage pour le SM 1231 AI 4 x TC 16 bits (6ES7 231-5QD30-0XB0)

Broche	X10 (or)	X11 (or)
1	L+ / 24 V-	Pas de connexion
2	M / 24 V-	Pas de connexion
3	GND	Pas de connexion
4	AI 0+ /TC	AI 2+ /TC
5	AI 0- /TC	AI 2- /TC
6	AI 1+ /TC	AI 3+ /TC
7	AI 1- /TC	AI 3- /TC

Tableau A- 143 Brochage pour le SM 1231 AI 8 x TC bits (6ES7 231-5QF30-0XB0)

Broche	X10 (or)	X11 (or)	X12 (or)	X13 (or)
1	L+ / 24 V-	Pas de connexion	Pas de connexion	Pas de connexion
2	M / 24 V-	Pas de connexion	Pas de connexion	Pas de connexion
3	GND	Pas de connexion	Pas de connexion	Pas de connexion
4	AI 0+ /TC	AI 2+ /TC	AI 4 I- /TC	AI 6 I- /TC

Broche	X10 (or)	X11 (or)	X12 (or)	X13 (or)
5	AI 0- /TC	AI 2- /TC	AI 4 I+ /TC	AI 6 I+ /TC
6	AI 1+ /TC	AI 3+ /TC	AI 5 M- /TC	AI 7 M- /TC
7	AI 1- /TC	AI 3- /TC	AI 5 M+ /TC	AI 7 M+ /TC

---

#### Remarque

Les entrées analogiques inutilisées doivent être court-circuitées.

Les voies thermocouple inutilisées peuvent être désactivées. Aucune erreur ne se produira si une voie inutilisée est désactivée.

---

### A.8.1.1 Principe de fonctionnement des thermocouples

Un thermocouple est composé de deux conducteurs métalliques de nature différente, reliés électriquement par soudure l'un à l'autre. Une tension proportionnelle à la température de la soudure froide est générée. Cette tension est faible ; un microvolt peut représenter beaucoup de degrés. Mesurer la tension provenant d'un thermocouple, compenser les soudures supplémentaires, puis linéariser le résultat constituent le principe fondamental des mesures de température à l'aide de thermocouples.

Lorsque vous raccordez un thermocouple au module SM 1231 Thermocouple, les deux fils métalliques de nature différente sont reliés au module, au niveau du connecteur de signaux du module. L'endroit où les deux fils sont reliés l'un à l'autre constitue le point de mesure du thermocouple.

Deux autres thermocouples sont constitués à l'endroit où les deux fils de nature différente sont raccordés au connecteur de signaux. La température du connecteur provoque une tension qui s'ajoute à la tension du thermocouple capteur. Si l'on ne corrige pas cette tension, la température indiquée sera différente de la température du capteur.

On utilise donc la compensation de soudure froide pour compenser le thermocouple du connecteur. Les tables de données de thermocouples se basent sur une température de soudure de référence, 0 degré Celsius en général. La compensation de soudure froide compense le connecteur à 0 degré Celsius. Elle restaure la tension ajoutée par les thermocouples du connecteur. La température du module est mesurée en interne, puis convertie en une valeur à ajouter à la conversion de capteur. On linéarise ensuite la conversion de capteur corrigée à l'aide des tables de thermocouples.

Pour un fonctionnement optimal de la compensation de soudure froide, le module Thermocouple doit se trouver dans un environnement thermique stable. Une variation lente (inférieure à 0,1 °C/minute) à une température ambiante du module est correctement compensée dans les limites des spécifications du module. Un déplacement d'air à travers le module entraînera également des erreurs de compensation de la soudure froide.

Si une meilleure compensation d'erreur de soudure froide est souhaitée, un bornier isotherme externe peut être utilisé. Le module Thermocouple met à disposition un bornier référencé à 0 °C ou à 50 °C.

**A.8.1.2 Tableaux de sélection pour les SM 1231 Thermocouple**

Les plages et la précision des différents types de thermocouple pris en charge par le module d'entrées SM 1231 Thermocouple sont présentées dans le tableau ci-dessous.

Tableau A- 144 Tableau de sélection du module SM 1231 Thermocouple

Type	Dépassement bas minimum <sup>1</sup>	Limite basse plage nominale	Limite haute plage nominale	Dépassement haut maximum <sup>2</sup>	Précision plage normale <sup>3,4</sup> @ 25°C	Précision plage normale <sup>1,2</sup> -20°C à 60°C
J	-210,0°C	-150,0°C	1200,0°C	1450,0°C	± 0,3°C	± 0,6°C
K	-270,0°C	-200,0°C	1372,0°C	1622,0°C	± 0,4°C	± 1,0°C
T	-270,0°C	-200,0°C	400,0°C	540,0°C	± 0,5°C	± 1,0°C
E	-270,0°C	-200,0°C	1000,0°C	1200,0°C	± 0,3°C	± 0,6°C
R & S	-50,0°C	100,0°C	1768,0°C	2019,0°C	± 1,0°C	± 2,5°C
N	-270,0°C	-200,0°C	1300,0°C	1550,0°C	± 1,0°C	± 1,6°C
C	0,0°C	100,0°C	2315,0°C	2500,0°C	± 0,7°C	± 2,7°C
TXK/XK(L)	-200,0°C	-150,0°C	800,0°C	1050,0°C	± 0,6°C	± 1,2°C
Tension	-32512	-27648 -80mV	27648 80mV	32511	±0.05%	±0.1%

- <sup>1</sup> Les valeurs de thermocouple inférieures à la valeur de dépassement bas minimum sont signalées en tant que -32768.
- <sup>2</sup> Les valeurs de thermocouple supérieures à la valeur de dépassement haut maximum sont signalées en tant que 32767.
- <sup>3</sup> L'erreur de soudure froide interne est de ±1,5°C pour toutes les plages. Cette valeur s'ajoute à l'erreur indiquée dans ce tableau. Le module requiert au minimum 30 minutes d'échauffement pour atteindre cette spécification.
- <sup>4</sup> La précision du SM 1231 AI 4 x 16 bits TC peut être réduite en présence de radiofréquence rayonnée entre 970 MHz et 990 MHz.

Tableau A- 145 Réduction des bruits et temps d'actualisation pour le SM 1231 Thermocouple

Sélection de la fréquence de réjection	Temps d'intégration	Temps d'actualisation module 4 voies (secondes)	Temps d'actualisation module 8 voies (secondes)
400 Hz (2,5 ms)	10 ms <sup>1</sup>	0.143	0.285
60 Hz (16,6 ms)	16,67 ms	0.223	0.445
50 Hz (20 ms)	20 ms	0.263	0.525
10 Hz (100 ms)	100 ms	1.225	2.450

- <sup>1</sup> Le temps d'intégration est de 10 ms pour conserver la résolution et la précision du module lorsque la réjection 400 Hz est sélectionnée. Cette sélection correspond également à une réjection des bruits 100 Hz et 200 Hz.

Il est recommandé d'utiliser un temps d'intégration de 100 ms pour la mesure des thermocouples. Le choix de temps d'intégration inférieurs augmentera l'erreur de répétabilité des températures lues.

### Remarque

Après sa mise sous tension, le module exécute un calibrage interne pour le convertisseur analogique-numérique. Pendant ce temps, il émet une valeur de 32767 sur chaque voie jusqu'à ce que des données valables soient disponibles sur cette voie. Il peut s'avérer nécessaire que votre programme utilisateur prévoie ce temps d'initialisation. Comme la configuration du module peut faire varier la longueur du temps d'initialisation, vous devez vérifier le comportement du module dans votre configuration. Si nécessaire, vous pouvez inclure de la logique dans votre programme utilisateur afin de prendre en compte le temps d'initialisation du module.

## Représentation des valeurs analogiques des thermocouples de type J

Le tableau ci-dessous donne la représentation des valeurs analogiques des thermocouples de type J.

Tableau A- 146 Représentation des valeurs analogiques des thermocouples de type J

Type J en °C	Unités		Type J en °F	Unités		Plage
	Décimal	Hexadécimal		Décimal	Hexadécimal	
> 1450.0	32767	7FFF	> 2642.0	32767	7FFF	Débordement haut
1450.0	14500	38A4	2642.0	26420	6734	Dépassement haut
:	:	:	:	:	:	
1200.1	12001	2EE1	2192.2	21922	55A2	Plage nominale
1200.0	12000	2EE0	2192.0	21920	55A0	
:	:	:	:	:	:	
-150.0	-1500	FA24	-238.0	-2380	F6B4	Débordement bas <sup>1</sup>
< -150.0	-32768	8000	< -238.0	-32768	8000	

<sup>1</sup> Le module thermocouple pourrait signaler un débordement bas en cas de câblage défectueux (par exemple, inversion de polarité ou entrées ouvertes) ou d'erreur de capteur dans la plage négative (par exemple, mauvais type de thermocouple).

## A.8.2 SM 1231 RTD

### Caractéristiques des SM 1231 RTD

Tableau A- 147 Caractéristiques générales

Caractéristiques techniques	SM 1231 AI 4 x RTD x 16 bits	SM 1231 AI 8 x RTD x 16 bits
Numéro de référence	6ES7 231-5PD30-0XB0	6ES7 231-5PD30-0XB0
Dimensions L x H x P (mm)	45 x 100 x 75	70 x 100 x 75
Poids	220 grammes	270 grammes
Dissipation de courant	1,5 W	1,5 W
Consommation de courant (bus SM)	80 mA	90 mA
Consommation de courant (24 V-) <sup>1</sup>	40 mA	40 mA

<sup>1</sup> 20,4 à 28,8 V- (classe 2, puissance limitée, ou alimentation de capteur de la CPU)

Tableau A- 148 Entrées analogiques

Caractéristiques techniques	SM 1231 AI 4 x RTD x 16 bits	SM 1231 AI 8 x RTD x 16 bits
Nombre d'entrées	4	8
Type	RTD référencée au module et ohms	RTD référencée au module et ohms
Plage	Voir le tableau de sélection des capteurs RTD (Page 833).	Voir le tableau de sélection des capteurs RTD (Page 833).
Plage nominale (mot de données)		
Plage de dépassement haut/bas (mot de données)		
Débordement haut / bas (mot de données)		
Résolution		
Température	0,1° C/0,1° F	0,1° C/0,1° F
Résistance	15 bits plus signe	15 bits plus signe
Tension de tenue maximum	± 35 V	± 35 V
Réjection des bruits	85 dB pour la réduction des bruits sélectionnée (10 Hz, 50 Hz, 60 Hz ou 400 Hz)	85 dB pour la réduction des bruits sélectionnée (10 Hz, 50 Hz, 60 Hz ou 400 Hz)
Réjection en mode commun	> 120 dB	> 120 dB
Impédance	≥ 10 MΩ	≥ 10 MΩ
Isolation		
Site à logique	500 V~	500 V~
Site à 24 V-	500 V~	500 V~
24 V- à logique	500 V~	500 V~
Isolation voie à voie	aucune	aucune
Précision	Voir le tableau de sélection des capteurs RTD (Page 833).	Voir le tableau de sélection des capteurs RTD (Page 833).
Répétabilité	± 0,05% pleine échelle	± 0,05% pleine échelle
Dissipation maximale du capteur	0,5 mW	0,5 mW
Principe de mesure	Intégration	Intégration

Caractéristiques techniques	SM 1231 AI 4 x RTD x 16 bits	SM 1231 AI 8 x RTD x 16 bits
Temps d'actualisation du module	Voir le tableau de sélection pour la réduction des bruits (Page 833).	Voir le tableau de sélection pour la réduction des bruits (Page 833).
Longueur de câble (mètres)	max. 100 mètres au capteur	max. 100 mètres au capteur
Résistance de fil	20 $\Omega$ , 2,7 $\Omega$ pour RTD 10 $\Omega$ max.	20 $\Omega$ , 2,7 $\Omega$ pour RTD 10 $\Omega$ max.

Tableau A- 149 Diagnostic

Caractéristiques techniques	SM 1231 AI 4 x RTD x 16 bits	SM 1231 AI 8 x RTD x 16 bits
Débordement haut/bas <sup>1,2</sup>	Oui	Oui
Rupture de fil <sup>3</sup>	Oui	Oui
Basse tension 24 V- <sup>1</sup>	Oui	Oui

<sup>1</sup> Les informations d'alarmes de diagnostic débordement haut, débordement bas et basse tension sont notifiées dans les valeurs de données analogiques même si les alarmes sont désactivées dans la configuration du module.

<sup>2</sup> La détection de débordement bas n'est jamais activée pour les plages de résistance.

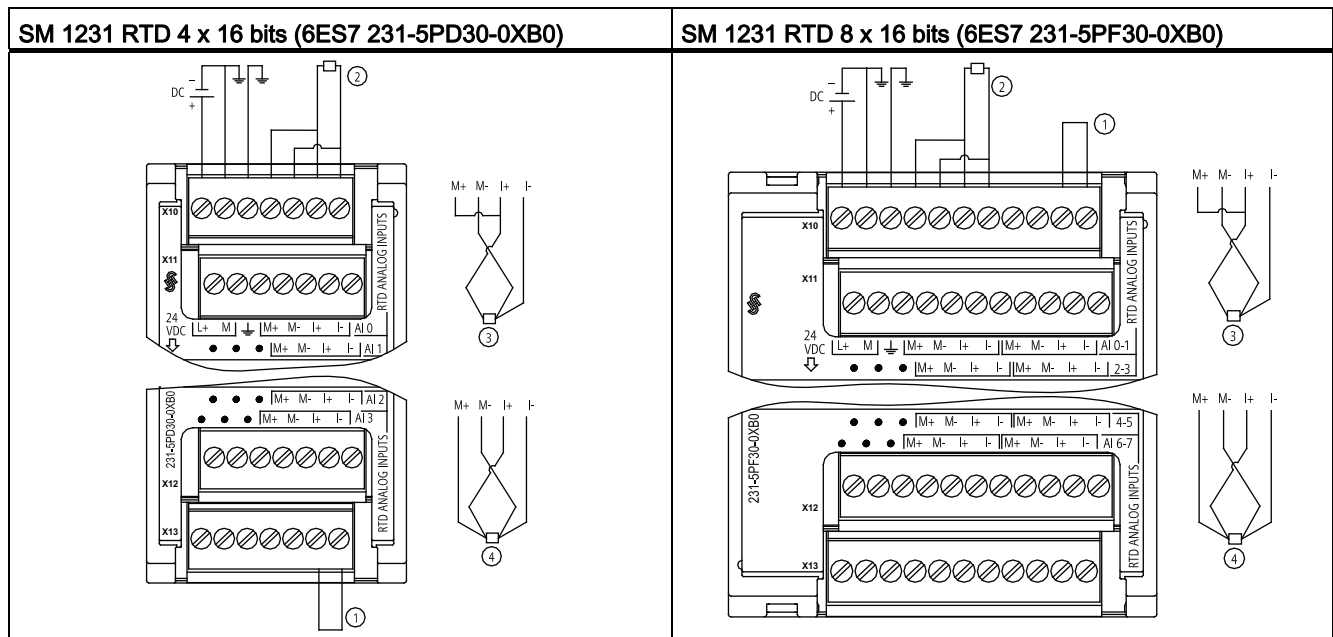
<sup>3</sup> Lorsque l'alarme de rupture de fil est désactivée et qu'un fil est ouvert dans le câblage du capteur, le module peut émettre des valeurs aléatoires.

Le module d'entrées-sorties analogiques SM 1231 RTD mesure la valeur de résistance connectée aux entrées du module. Il est possible de sélectionner "Résistance" ou "Thermistance" comme type de mesure.

- "Résistance" : La valeur pleine échelle de la plage nominale sera 27648 en décimal.
- "Thermistance" : La valeur sera notifiée en degrés multipliés par dix (par exemple, 25,3 degrés seront indiqués sous la forme 253 en décimal).

Le module SM 1231 RTD prend en charge des montages 2 fils, 3 fils et 4 fils avec la résistance du capteur.

Tableau A- 150 Schémas de câblage pour les SM RTD



- ① Bouclage sur les entrées RTD inutilisées
- ② RTD 2 fils ③ RTD 3 fils ④ RTD 4 fils

REMARQUE : Remarque : Les connecteurs doivent être en or. Vous trouverez les numéros de référence à l'annexe C, "Pièces de rechange".

Tableau A- 151 Brochage pour le SM 1231 RTD 4 x 16 bits (6ES7 231-5PD30-0XB0)

Broche	X10 (or)	X11 (or)	X12 (or)	X13 (or)
1	L+ / 24 V-	Pas de connexion	Pas de connexion	Pas de connexion
2	M / 24 V-	Pas de connexion	Pas de connexion	Pas de connexion
3	GND	Pas de connexion	Pas de connexion	Pas de connexion
4	AI 0 M+ /RTD	AI 1 M+ /RTD	AI 2 M+ /RTD	AI 3 M+ /RTD
5	AI 0 M- /RTD	AI 1 M- /RTD	AI 2 M- /RTD	AI 3 M- /RTD
6	AI 0 I+ /RTD	AI 1 I+ /RTD	AI 2 I+ /RTD	AI 3 I+ /RTD
7	AI 0 I- /RTD	AI 1 I- /RTD	AI 2 I- /RTD	AI 3 I- /RTD

Tableau A- 152 Brochage pour le SM 1231 RTD 8 x 16 bits (6ES7 231-5PF30-0XB0)

Broche	X10 (or)	X11 (or)	X12 (or)	X13 (or)
1	L+ / 24 V-	Pas de connexion	Pas de connexion	Pas de connexion
2	M / 24 V-	Pas de connexion	Pas de connexion	Pas de connexion
3	GND	Pas de connexion	Pas de connexion	Pas de connexion
4	AI 0 M+ /RTD	AI 2 M+ /RTD	AI 4 M+ /RTD	AI 6 M+ /RTD
5	AI 0 M- /RTD	AI 2 M- /RTD	AI 4 M- /RTD	AI 6 M- /RTD



Broche	X10 (or)	X11 (or)	X12 (or)	X13 (or)
6	AI 0 I+ /RTD	AI 2 I+ /RTD	AI 4 I+ /RTD	AI 6 I+ /RTD
7	AI 0 I- /RTD	AI 2 I- /RTD	AI 4 I- /RTD	AI 6 I- /RTD
8	AI 1 M+ /RTD	AI 3 M+ /RTD	AI 5 M+ /RTD	AI 7 M+ /RTD
9	AI 1 M- /RTD	AI 3 M- /RTD	AI 5 M- /RTD	AI 7 M- /RTD
10	AI 1 I+ /RTD	AI 3 I+ /RTD	AI 5 I+ /RTD	AI 7 I+ /RTD
11	AI 1 I- /RTD	AI 3 I- /RTD	AI 5 I- /RTD	AI 7 I- /RTD

**Remarque**

Les voies RTD inutilisées peuvent être désactivées. Aucune erreur ne se produira si une voie inutilisée est désactivée.

La boucle de courant du module RTD doit être continue pour éliminer le temps de stabilisation supplémentaire qui est automatiquement ajouté à une voie inutilisée qui n'est pas désactivée. Pour des raisons de cohérence, le module RTD doit avoir une résistance connectée (comme le montage RTD 2 fils).

**A.8.2.1 Tableaux de sélection pour les SM 1231 RTD**

Tableau A- 153 Plages et précision pour les différents capteurs pris en charge par les modules RTD

Coefficient de température	Type de RTD	Dépassement bas minimum <sup>1</sup>	Limite basse plage nominale	Limite haute plage nominale	Dépassement haut maximum <sup>2</sup>	Précision de plage normale à 25°C	Précision de plage normale de -20°C à 60°C
Pt 0.003850 ITS90 DIN EN 60751	Pt 10	-243,0°C	-200,0°C	850,0°C	1000,0°C	± 1,0°C	± 2,0°C
	Pt 50	-243,0°C	-200,0°C	850,0°C	1000,0°C	± 0,5°C	± 1,0°C
	Pt 100						
	Pt 200						
	Pt 500						
	Pt 1000						
Pt 0.003902 Pt 0.003916 Pt 0.003920	Pt 100	-243,0°C	-200,0°C	850,0°C	1000,0°C	± 0,5°C	± 1,0°C
	Pt 200	-243,0°C	-200,0°C	850,0°C	1000,0°C	± 0,5°C	± 1,0°C
	Pt 500						
	Pt 1000						
Pt 0,003910	Pt 10	-273,2°C	-240,0°C	1100,0°C	1295°C	± 1,0°C	± 2,0°C
	Pt 50	-273,2°C	-240,0°C	1100,0°C	1295°C	± 0,8°C	± 1,6°C
	Pt 100						
	Pt 500						
Ni 0.006720 Ni 0.006180	Ni 100	-105,0°C	-60,0°C	250,0°C	295,0°C	± 0,5°C	± 1,0°C
	Ni 120						
	Ni 200						

Coefficient de température	Type de RTD	Dépassement bas minimum <sup>1</sup>	Limite basse plage nominale	Limite haute plage nominale	Dépassement haut maximum <sup>2</sup>	Précision de plage normale à 25°C	Précision de plage normale de -20°C à 60°C
	Ni 500						
	Ni 1000						
LG-Ni 0.005000	LG-Ni 1000	-105,0°C	-60,0°C	250,0°C	295,0°C	± 0,5°C	± 1,0°C
Ni 0.006170	Ni 100	-105,0°C	-60,0°C	180,0°C	212,4°C	± 0,5°C	± 1,0°C
Cu 0.004270	Cu 10	-240,0°C	-200,0°C	260,0°C	312,0°C	± 1,0°C	± 2,0°C
Cu 0.004260	Cu 10	-60,0°C	-50,0°C	200,0°C	240,0°C	± 1,0°C	± 2,0°C
	Cu 50	-60,0°C	-50,0°C	200,0°C	240,0°C	± 0,6°C	± 1,2°C
	Cu 100						
Cu 0.004280	Cu 10	-240,0°C	-200,0°C	200,0°C	240,0°C	± 1,0°C	± 2,0°C
	Cu 50	-240,0°C	-200,0°C	200,0°C	240,0°C	± 0,7°C	± 1,4°C
	Cu 100						

<sup>1</sup> Les valeurs RTD inférieures à la valeur de dépassement bas minimum sont signalées en tant que -32768.

<sup>2</sup> Les valeurs RTD supérieures à la valeur de dépassement haut maximum sont signalées en tant que +32767.

Tableau A- 154 Résistance

Plage	Dépassement bas minimum	Limite basse plage nominale	Limite haute plage nominale	Dépassement haut maximum <sup>1</sup>	Précision de plage normale à 25°C	Précision de plage normale de -20°C à 60°C
150 Ω	Sans objet	0 (0 Ω)	27648 (150 Ω)	176,383 Ω	±0.05%	±0.1%
300 Ω	Sans objet	0 (0 Ω)	27648 (300 Ω)	352,767 Ω	±0.05%	±0.1%
600 Ω	Sans objet	0 (0 Ω)	27648 (600 Ω)	705,534 Ω	±0.05%	±0.1%

<sup>1</sup> Les valeurs de résistance supérieures à la valeur de dépassement haut maximum sont signalées comme +32767.

### Remarque

Le module signale 32767 sur toute voie activée à laquelle aucun capteur n'est connecté. Si la détection de fil ouvert est également activée, le module fait clignoter les DEL appropriées.

Si des plages RTD 500 Ω et 1000 Ω sont utilisées avec d'autres résistances à valeur inférieure, l'erreur peut atteindre deux fois l'erreur indiquée.

La meilleure précision est obtenue pour les plages RTD 10 Ω avec des montages 4 fils.

La résistance des fils de connexion en montage 2 fils provoquera une erreur dans la lecture du capteur. La précision ne peut donc pas être garantie.

Tableau A- 155 Réduction des bruits et temps d'actualisation pour les modules RTD

Sélection de la fréquence de réjection	Temps d'intégration	Temps d'actualisation (secondes)	
		module 4 voies	module 8 voies
400 Hz (2,5 ms)	10 ms <sup>1</sup>	4 / 2 fils : 0.142 3 fils : 0.285	4 / 2 fils : 0.285 3 fils : 0.525
60 Hz (16,6 ms)	16,67 ms	4 / 2 fils : 0.222 3 fils : 0.445	4 / 2 fils : 0.445 3 fils : 0.845
50 Hz (20 ms)	20 ms	4 / 2 fils : 0.262 3 fils : .505	4 / 2 fils : 0.524 3 fils : 1.015
10 Hz (100 ms)	100 ms	4 / 2 fils : 1.222 3 fils : 2.445	4 / 2 fils : 2.425 3 fils : 4.845

- <sup>1</sup> Le temps d'intégration est de 10 ms pour conserver la résolution et la précision du module lorsque le filtre 400 Hz est sélectionné. Cette sélection correspond également à une réjection des bruits 100 Hz et 200 Hz.

**IMPORTANT**

Après sa mise sous tension, le module exécute un calibrage interne pour le convertisseur analogique-numérique. Pendant ce temps, il émet une valeur de 32767 sur chaque voie jusqu'à ce que des données valables soient disponibles sur cette voie. Il peut s'avérer nécessaire que votre programme utilisateur prévoie ce temps d'initialisation. Comme la configuration du module peut faire varier la longueur du temps d'initialisation, vous devez vérifier le comportement du module dans votre configuration. Si nécessaire, vous pouvez inclure de la logique dans votre programme utilisateur afin de prendre en compte le temps d'initialisation du module.

**Représentation des valeurs analogiques pour RTD**

Les tableaux ci-dessous donnent la représentation de la valeur mesurée numérisée pour les sondes RTD dans la plage de température standard.

Tableau A- 156 Représentation des valeurs analogiques pour les thermomètres à résistance PT 100, 200, 500, 1000 et PT 10, 50, 100, 500 GOST (0.003850) standard

Pt x00 standard en °C (1 chiffre = 0,1 °C)	Unités		Pt x00 standard en °F (1 chiffre = 0,1 °F)	Unités		Plage
	Décimal	Hexadécimal		Décimal	Hexadécimal	
> 1000.0	32767	7FFF	> 1832.0	32767	7FFF	Débordement haut
1000.0	10000	2710	1832.0	18320	4790	Dépassement haut
:	:	:	:	:	:	
850.1	8501	2135	1562.1	15621	3D05	
850.0	8500	2134	1562.0	15620	3D04	Plage nominale
:	:	:	:	:	:	
-200.0	-2000	F830	-328.0	-3280	F330	

Pt x00 standard en °C (1 chiffre = 0,1 °C)	Unités		Pt x00 standard en °F (1 chiffre = 0,1 °F)	Unités		Plage
	Décimal	Hexadécimal		Décimal	Hexadécimal	
-200.1 :	-2001 :	F82F :	-328.1 :	-3281 :	F32F :	Dépasseme nt bas
-243.0	-2430	F682	-405.4	-4054	F02A	
< -243.0	-32768	8000	< -405.4	-32768	8000	Débordemen t bas

## A.9 Signal Boards (SB) TOR

### A.9.1 Caractéristiques des SB 1221 entrées TOR, 200 kHz

Tableau A- 157 Caractéristiques générales

Caractéristiques techniques	SB 1221 DI 4 x 24 VDC, 200 kHz	SB 1221 DI 4 x 5 VDC, 200 kHz
Numéro de référence	6ES7 221-3BD30-0XB0	6ES7 221-3AD30-0XB0
Dimensions L x H x P (mm)	38 x 62 x 21	38 x 62 x 21
Poids	35 grammes	35 grammes
Dissipation de courant	1,5 W	1,0 W
Consommation de courant (bus SM)	40 mA	40 mA
Consommation de courant (24 V-)	7 mA / entrée + 20 mA	15 mA / entrée + 15 mA

Tableau A- 158 Entrées TOR

Caractéristiques techniques	SB 1221 DI 4 x 24 VDC, 200 kHz	SB 1221 DI 4 x 5 VDC, 200 kHz
Nombre d'entrées	4	4
Type	N	N
Tension nominale	24 V- à 7 mA, nominal	5 V- à 15 mA, nominal
Tension continue admise	28,8 V-	6 V-
Tension de choc	35 V- pour 0,5 s	6 V
Signal 1 logique (min.)	L+ moins 10 V- à 2,9 mA	L+ moins 2,0 V- à 5,1 mA
Signal 0 logique (max.)	L+ moins 5 V- à 1,4 mA	L+ moins 1,0 V- à 2,2 mA
Fréquences d'entrée d'horloge HSC (max.)	Monophasé : 200 kHz Quadrature de phase : 160 kHz	Monophasé : 200 kHz Quadrature de phase : 160 kHz
Isolation (site à logique)	500 V~ pour 1 minute	500 V~ pour 1 minute
Groupes d'isolation	1	1
Temps de filtre	0,2, 0,4, 0,8, 1,6, 3,2, 6,4 et 12,8 ms, sélectionnables par groupes de 4	0,2, 0,4, 0,8, 1,6, 3,2, 6,4 et 12,8 ms, sélectionnables par groupes de 4

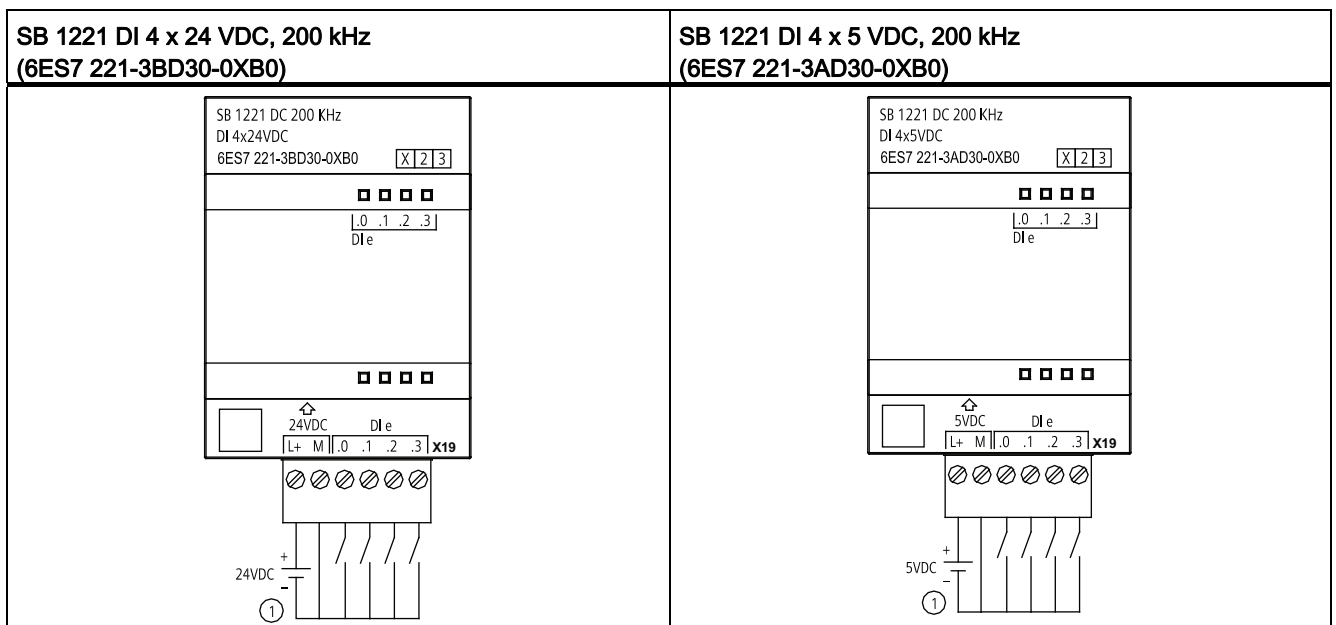
Caractéristiques techniques	SB 1221 DI 4 x 24 VDC, 200 kHz	SB 1221 DI 4 x 5 VDC, 200 kHz
Nombre d'entrées simultanément à 1	<ul style="list-style-type: none"> <li>2 (pas d'E/S adjacentes) à 60 °C horizontal ou 50 °C vertical</li> <li>4 à 55 °C horizontal ou 45 °C vertical</li> </ul>	4
Longueur de câble (mètres)	50, paire torsadée blindée	50, paire torsadée blindée

### IMPORTANT

En cas de commutation de fréquences au-delà de 20 kHz, il est important que les entrées TOR reçoivent une onde carrée. Tenez compte des possibilités suivantes pour améliorer la qualité du signal arrivant aux entrées :

- Diminuez la longueur du câble.
- Remplacez un étage P seul par un étage P et N.
- Choisissez un câble de qualité supérieure.
- Réduisez le circuit/les composants de 24 V à 5 V.
- Ajoutez une charge externe à l'entrée.

Tableau A- 159 Schémas de câblage des SB entrées TOR, 200 kHz



① Prise en charge d'entrées de type N uniquement

Tableau A- 160 Brochage pour le SB 1221 DI 4 x 24 VDC, 200 kHz (6ES7 221-3BD30-0XB0)

Broche	X19
1	L+ / 24 V-
2	M / 24 V-
3	DI e.0
4	DI e.1
5	DI e.2
6	DI e.3

Tableau A- 161 Brochage pour le SB 1221 DI 4 x 5 VDC, 200 kHz (6ES7 221-3AD30-0XB0)

Broche	X19
1	L+ / 5 V-
2	M / 5 V-
3	DI e.0
4	DI e.1
5	DI e.2
6	DI e.3

## A.9.2 Caractéristiques des SB 1222 sorties TOR, 200 kHz

Tableau A- 162 Caractéristiques générales

Caractéristiques techniques	SB 1222 DQ 4 x 24 VDC, 200 kHz	SB 1222 DQ 4 x 5 VDC, 200 kHz
Numéro de référence	6ES7 222-1BD30-0XB0	6ES7 222-1AD30-0XB0
Dimensions L x H x P (mm)	38 x 62 x 21	38 x 62 x 21
Poids	35 grammes	35 grammes
Dissipation de courant	0,5 W	0,5 W
Consommation de courant (bus SM)	35 mA	35 mA
Consommation de courant (24 V-)	15 mA	15 mA

Tableau A- 163 Sorties TOR

Caractéristiques techniques	SB 1222 DQ 4 x 24 VDC, 200 kHz	SB 1222 DQ 4 x 5 VDC, 200 kHz
Nombre de sorties	4	4
Type des sorties	Transistor à effet de champ MOS (P et N) <sup>1</sup>	Transistor à effet de champ MOS (P et N) <sup>1</sup>
Plage de tension	20,4 à 28,8 V-	4,25 à 6,0 V-
Signal 1 logique à courant max.	L+ moins 1,5 V	L+ moins 0,7 V
Signal 0 logique à courant max.	1,0 V-, max.	0,2 V-, max.

Caractéristiques techniques	SB 1222 DQ 4 x 24 VDC, 200 kHz	SB 1222 DQ 4 x 5 VDC, 200 kHz
Courant (max.)	0,1 A	0,1 A
Charge de lampe	--	--
Résistance état activé (contact)	11 $\Omega$ max.	7 $\Omega$ max.
Résistance état désactivé	6 $\Omega$ max.	0,2 $\Omega$ max.
Courant de fuite par sortie	--	--
Fréquence de sortie de trains d'impulsions	200 kHz max., 2 Hz min.	200 kHz max., 2 Hz min.
Courant de choc	0,11 A	0,11 A
Protection contre la surcharge	Non	Non
Isolation (site à logique)	500 V~ pour 1 minute	500 V~ pour 1 minute
Groupes d'isolation	1	1
Courants par commun	0,4 A	0,4 A
Tension de blocage inductive	Aucune	Aucune
Retard de commutation	1,5 $\mu$ s + 300 ns à la montée 1,5 $\mu$ s + 300 ns à la retombée	200 ns + 300 ns à la montée 200 ns + 300 ns à la retombée
Comportement au passage de MARCHE à ARRÊT	Dernière valeur ou valeur de remplacement (valeur par défaut 0)	Dernière valeur ou valeur de remplacement (valeur par défaut 0)
Nombre de sorties simultanément à 1	<ul style="list-style-type: none"> <li>• 2 (pas d'E/S adjacentes) à 60 °C horizontal ou 50 °C vertical</li> <li>• 4 à 55 °C horizontal ou 45 °C vertical</li> </ul>	4
Longueur de câble (mètres)	50, paire torsadée blindée	50, paire torsadée blindée

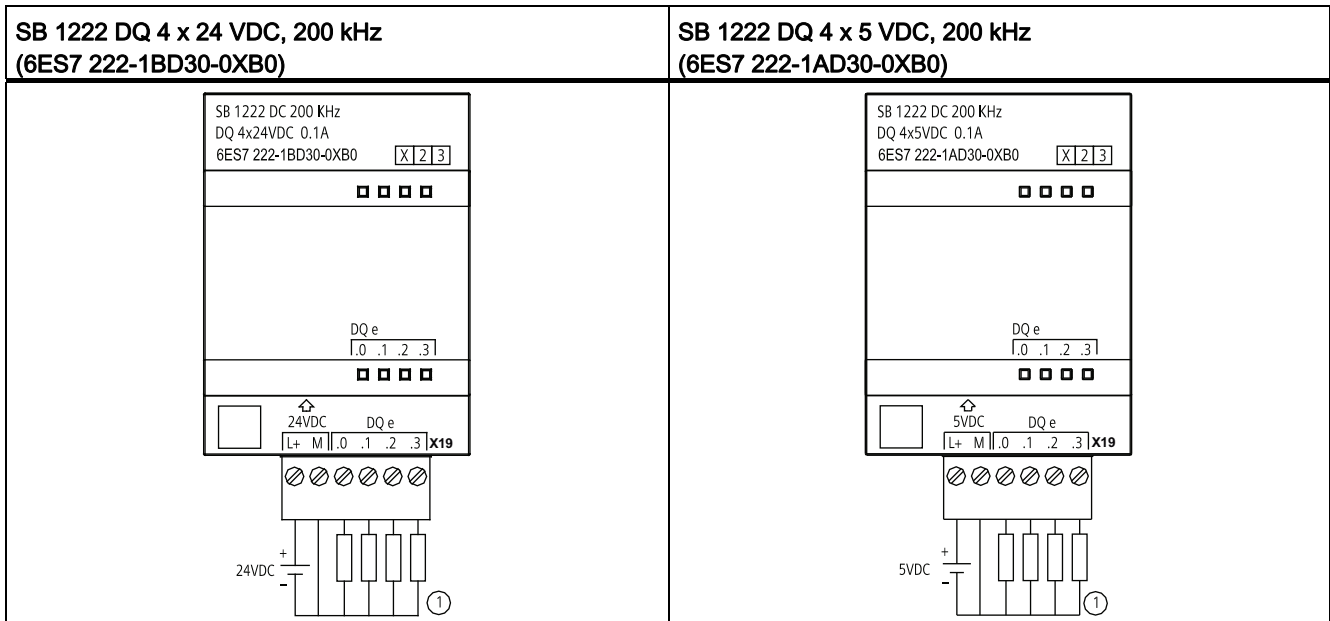
- <sup>1</sup> Comme les deux configurations "sink" et "source" (P et N) sont prises en charge par la même circuiterie, l'état actif d'une charge "source" est l'opposé de celui d'une charge "sink". Une sortie "source" fournit une logique positive (bit Q et DEL activés lorsque la charge présente un flux de courant) alors qu'une sortie "sink" fournit une logique négative (bit Q et DEL désactivés lorsque la charge présente un flux de courant). Si le module est enfiché sans programme utilisateur, le paramétrage par défaut pour ce module est 0 V, ce qui signifie qu'une charge "sink" sera activée.

#### IMPORTANT

En cas de commutation de fréquences au-delà de 20 kHz, il est important que les entrées TOR reçoivent une onde carrée. Tenez compte des possibilités suivantes pour améliorer la qualité du signal arrivant aux entrées :

- Diminuez la longueur du câble.
- Remplacez un étage P seul par un étage P et N.
- Choisissez un câble de qualité supérieure.
- Réduisez le circuit/les composants de 24 V à 5 V.
- Ajoutez une charge externe à l'entrée.

Tableau A- 164 Schémas de câblage des SB sorties TOR, 200 kHz



① Pour les sorties de type N, connectez "Charge" à "-" (comme illustré). Pour les sorties de type P, connectez "Charge" à "+". Comme les deux configurations "sink" et "source" (P et N) sont prises en charge par la même circuiterie, l'état actif d'une charge "source" est l'opposé de celui d'une charge "sink". Une sortie "source" fournit une logique positive (bit Q et DEL activés lorsque la charge présente un flux de courant) alors qu'une sortie "sink" fournit une logique négative (bit Q et DEL désactivés lorsque la charge présente un flux de courant). Si le module est enfiché sans programme utilisateur, le paramétrage par défaut pour ce module est 0 V, ce qui signifie qu'une charge "sink" sera activée.

Tableau A- 165 Brochage pour le SB 1222 DQ 4 x 24 VDC, 200 kHz (6ES7 222-1BD30-0XB0)

Broche	X19
1	L+ / 24 V-
2	M / 24 V-
3	DQ e.0
4	DQ e.1
5	DQ e.2
6	DQ e.3

Tableau A- 166 Brochage pour le SB 1222 DQ 4 x 5 VDC, 200 kHz (6ES7 222-1AD30-0XB0)

Broche	X19
1	L+ / 5 V-
2	M / 5 V-
3	DQ e.0
4	DQ e.1



Broche	X19
5	DQ e.2
6	DQ e.3

### A.9.3 Caractéristiques des SB 1223 entrées/sorties TOR, 200 kHz

Tableau A- 167 Caractéristiques générales

Caractéristiques techniques	SB 1223 DI 2 x 24 VDC / DQ 2 x 24 VDC, 200 kHz	SB 1223 DI 2 x 5 VDC / DQ 2 x 5 VDC, 200 kHz
Numéro de référence	6ES7 223-3BD30-0XB0	6ES7 223-3AD30-0XB0
Dimensions L x H x P (mm)	38 x 62 x 21	38 x 62 x 21
Poids	35 grammes	35 grammes
Dissipation de courant	1,0 W	0,5 W
Consommation de courant (bus SM)	35 mA	35 mA
Consommation de courant (24 V-)	7 mA / entrée + 30 mA	15 mA / entrée + 15 mA

Tableau A- 168 Entrées TOR

Caractéristiques techniques	SB 1223 DI 2 x 24 VDC / DQ 2 x 24 VDC, 200 kHz	SB 1223 DI 2 x 5 VDC / DQ 2 x 5 VDC, 200 kHz
Nombre d'entrées	2	2
Type	N	N
Tension nominale	24 V- à 7 mA, nominal	5 V- à 15 mA, nominal
Tension continue admise	28,8 V-	6 V-
Tension de choc	35 V- pour 0,5 s	6 V
Signal 1 logique (min.)	L+ moins 10 V- à 2,9 mA	L+ moins 2,0 V- à 5,1 mA
Signal 0 logique (max.)	L+ moins 5 V- à 1,4 mA	L+ moins 1,0 V- à 2,2 mA
Fréquences d'entrée d'horloge HSC (max.)	Monophasé : 200 kHz Quadrature de phase : 160 kHz	Monophasé : 200 kHz Quadrature de phase : 160 kHz
Isolation (site à logique)	500 V~ pour 1 minute	500 V~ pour 1 minute
Groupes d'isolation	1 (sans isolation vers les sorties)	1 (sans isolation vers les sorties)
Temps de filtre	0,2, 0,4, 0,8, 1,6, 3,2, 6,4 et 12,8 ms, sélectionnables par groupes de 4	0,2, 0,4, 0,8, 1,6, 3,2, 6,4 et 12,8 ms, sélectionnables par groupes de 4
Nombre d'entrées simultanément à 1	2	2
Longueur de câble (mètres)	50, paire torsadée blindée	50, paire torsadée blindée

Tableau A- 169 Sorties TOR

Caractéristiques techniques	SB 1223 DI 2 x 24 VDC / DQ 2 x 24 VDC, 200 kHz	SB 1223 DI 2 x 5 VDC / DQ 2 x 5 VDC, 200 kHz
Nombre de sorties	2	2
Type des sorties	Transistor à effet de champ MOS (P et N) <sup>1</sup>	Transistor à effet de champ MOS (P et N) <sup>1</sup>
Plage de tension	20,4 à 28,8 V-	4,25 à 6,0 V-
Valeur nominale	24 V-	5 V-
Signal 1 logique à courant max.	L+ moins 1,5 V	L+ moins 0,7 V
Signal 0 logique à courant max.	1,0 V-, max.	0,2 V-, max.
Courant (max.)	0,1 A	0,1 A
Charge de lampe	--	--
Résistance état activé (contact)	11 Ω max.	7 Ω max.
Résistance état désactivé	6 Ω max.	0,2 Ω max.
Courant de fuite par sortie	--	--
Fréquence de sortie de trains d'impulsions	200 kHz max., 2 Hz min.	200 kHz max., 2 Hz min.
Courant de choc	0,11 A	0,11 A
Protection contre la surcharge	Non	Non
Isolation (site à logique)	500 V~ pour 1 minute	500 V~ pour 1 minute
Groupes d'isolation	1 (sans isolation vers les entrées)	1 (sans isolation vers les entrées)
Courants par commun	0,2 A	0,2 A
Tension de blocage inductive	Aucune	Aucune
Retard de commutation	1,5 µs + 300 ns à la montée 1,5 µs + 300 ns à la retombée	200 ns + 300 ns à la montée 200 ns + 300 ns à la retombée
Comportement au passage de MARCHE à ARRÊT	Dernière valeur ou valeur de remplacement (valeur par défaut 0)	Dernière valeur ou valeur de remplacement (valeur par défaut 0)
Nombre de sorties simultanément à 1	2	2
Longueur de câble (mètres)	50, paire torsadée blindée	50, paire torsadée blindée

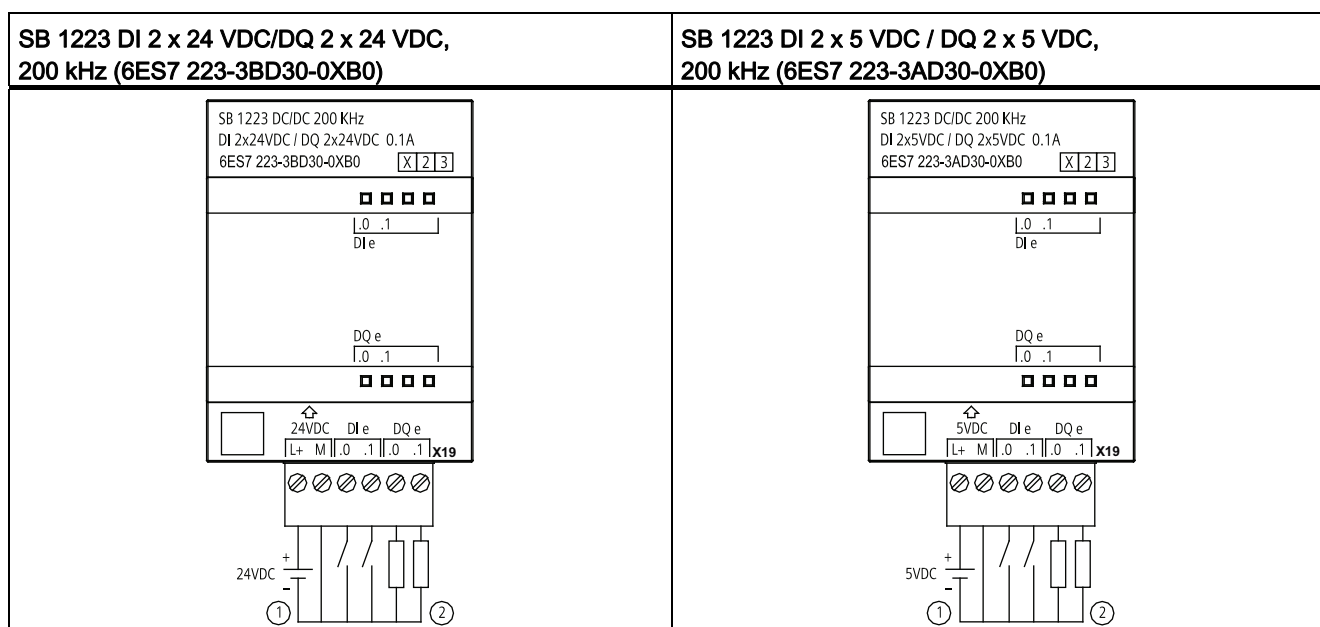
<sup>1</sup> Comme les deux configurations "sink" et "source" (P et N) sont prises en charge par la même circuiterie, l'état actif d'une charge "source" est l'opposé de celui d'une charge "sink". Une sortie "source" fournit une logique positive (bit Q et DEL activés lorsque la charge présente un flux de courant) alors qu'une sortie "sink" fournit une logique négative (bit Q et DEL désactivés lorsque la charge présente un flux de courant). Si le module est enfiché sans programme utilisateur, le paramétrage par défaut pour ce module est 0 V, ce qui signifie qu'une charge "sink" sera activée.

**IMPORTANT**

En cas de commutation de fréquences au-delà de 20 kHz, il est important que les entrées TOR reçoivent une onde carrée. Tenez compte des possibilités suivantes pour améliorer la qualité du signal arrivant aux entrées :

- Diminuez la longueur du câble.
- Remplacez un étage P seul par un étage P et N.
- Choisissez un câble de qualité supérieure.
- Réduisez le circuit/les composants de 24 V à 5 V.
- Ajoutez une charge externe à l'entrée.

Tableau A- 170 Schémas de câblage des SB entrées / sorties TOR, 200 kHz



- ① Prise en charge d'entrées de type N uniquement
- ② Pour les sorties de type N, connectez "Charge" à "-" (comme illustré). Pour les sorties de type P, connectez "Charge" à "+". 1 Comme les deux configurations "sink" et "source" (P et N) sont prises en charge par la même circuiterie, l'état actif d'une charge "source" est l'opposé de celui d'une charge "sink". Une sortie "source" fournit une logique positive (bit Q et DEL activés lorsque la charge présente un flux de courant) alors qu'une sortie "sink" fournit une logique négative (bit Q et DEL désactivés lorsque la charge présente un flux de courant). Si le module est enfiché sans programme utilisateur, le paramétrage par défaut pour ce module est 0 V, ce qui signifie qu'une charge "sink" sera activée.

Tableau A- 171 Brochage pour le SB 1223 DI 2 x 24 VDC/DQ 2 x 24 VDC, 200 kHz (6ES7 223-3BD30-0XB0)

Broche	X19
1	L+ / 24 V-
2	M / 24 V-
3	DI e.0
4	DI e.1
5	DQ e.0
6	DQ e.1

Tableau A- 172 Brochage pour le SB 1223 DI 2 x 5 VDC / DQ 2 x 5 VDC, 200 kHz (6ES7 223-3AD30-0XB0)

Broche	X19
1	L+ / 5 V-
2	M / 5 V-
3	DI e.0
4	DI e.1
5	DQ e.0
6	DQ e.1

#### A.9.4 Caractéristiques du SB 1223 2 entrées 24 VDC / 2 sorties 24 VDC

Tableau A- 173 Caractéristiques générales

Caractéristiques techniques	SB 1223 DI 2 x 24 VDC, DQ 2 x 24 VDC
Numéro de référence	6ES7 223-0BD30-0XB0
Dimensions L x H x P (mm)	38 x 62 x 21
Poids	40 grammes
Dissipation de courant	1,0 W
Consommation de courant (bus SM)	50 mA
Consommation de courant (24 V-)	4 mA/entrée utilisée

Tableau A- 174 Entrées TOR

Caractéristiques techniques	SB 1223 DI 2 x 24 VDC, DQ 2 x 24 VDC
Nombre d'entrées	2
Type	CEI type 1 en mode P
Tension nominale	24 V- à 4 mA, nominal
Tension continue admise	30 V- max.

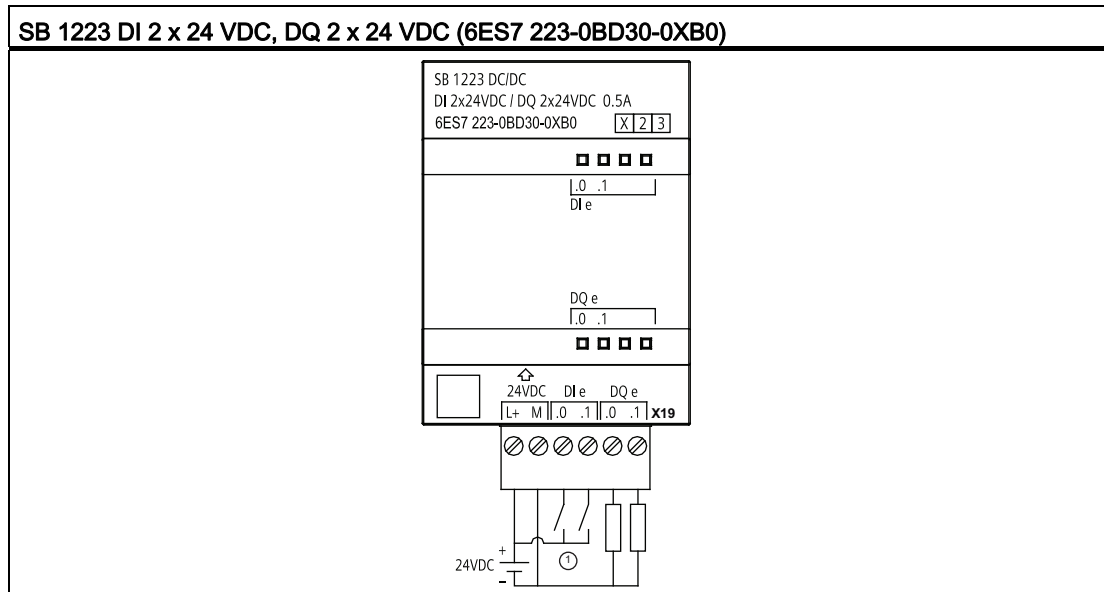
Caractéristiques techniques	SB 1223 DI 2 x 24 VDC, DQ 2 x 24 VDC
Tension de choc	35 V- pour 0,5 s
Signal 1 logique (min.)	15 V- à 2,5 mA
Signal 0 logique (max.)	5 V- à 1 mA
Fréquences d'entrée d'horloge HSC (max.)	20 kHz (15 à 30 V-) 30 kHz (15 à 26 V-)
Isolation (site à logique)	500 V~ pour 1 minute
Groupes d'isolation	1
Temps de filtre	0,2, 0,4, 0,8, 1,6, 3,2, 6,4 et 12,8 ms Sélectionnable en groupes de 2
Nombre d'entrées simultanément à 1	2
Longueur de câble (mètres)	500 blindé, 300 non blindé

Tableau A- 175 Sorties TOR

Caractéristiques techniques	SB 1223 DI 2 x 24 VDC, DQ 2 x 24 VDC
Nombre de sorties	2
Type des sorties	Transistor à effet de champ MOS (mode N)
Plage de tension	20,4 à 28,8 V-
Signal 1 logique à courant max.	20 V- min.
Signal 0 logique avec charge 10 k $\Omega$	0,1 V- max.
Courant (max.)	0,5 A
Charge de lampe	5 W
Résistance état activé (contact)	0,6 $\Omega$ max.
Courant de fuite par sortie	10 $\mu$ A max.
Fréquence de sortie de trains d'impulsions (PTO)	20 kHz max., 2 Hz min. <sup>1</sup>
Courant de choc	5 A pour 100 ms max.
Protection contre la surcharge	Non
Isolation (site à logique)	500 V~ pour 1 minute
Groupes d'isolation	1
Courants par commun	1 A
Tension de blocage inductive	L+ moins 48 V, dissipation 1 W
Retard de commutation	2 $\mu$ s max. de 0 à 1 10 $\mu$ s max. de 1 à 0
Comportement au passage de MARCHE à ARRÊT	Dernière valeur ou valeur de remplacement (valeur par défaut 0)
Nombre de sorties simultanément à 1	2
Longueur de câble (mètres)	500 m blindé, 150 m non blindé

<sup>1</sup> Selon vos câble et récepteur d'impulsions, une résistance de charge supplémentaire (au moins 10% du courant nominal) peut améliorer la qualité du signal d'impulsion et l'immunité aux bruits.

Tableau A- 176 Schéma de câblage du SB entrées / sorties TOR



① Prise en charge d'entrées de type P uniquement

Tableau A- 177 Brochage pour le SB 1223 DI 2 x 24 VDC, DQ 2 x 24 VDC (6ES7 223-0BD30-0XB0)

Broche	X19
1	L+ / 24 V-
2	M / 24 V-
3	DI e.0
4	DI e.1
5	DQ e.0
6	DQ e.1

## A.10 Signal Boards (SB) analogiques

### A.10.1 Caractéristiques du SB 1231 1 sortie analogique

#### Remarque

Le firmware de votre CPU doit être de version V2.0 ou plus pour que vous puissiez utiliser ce SB.

Tableau A- 178 Caractéristiques générales

<b>Caractéristiques techniques</b>	<b>SB 1231 AI 1 x 12 bits</b>
Numéro de référence	6ES7 231-4HA30-0XB0
Dimensions L x H x P (mm)	38 x 62 x 21
Poids	35 grammes
Dissipation de courant	0,4 W
Consommation de courant (bus SM)	55 mA
Consommation de courant (24 V-)	aucune

Tableau A- 179 Entrées analogiques

<b>Caractéristiques techniques</b>	<b>SB 1231 AI 1x12 bits</b>
Nombre d'entrées	1
Type	Tension ou courant (différentiel)
Plage	$\pm 10V$ , $\pm 5V$ , $\pm 2,5V$ ou 0 à 20 mA
Résolution	11 bits + bit de signe
Plage pleine échelle (mot de données)	-27648 à 27648
Dépassement haut/bas (mot de données)	Tension : 32 511 à 27 649 / -27 649 à -32 512 Courant : 32 511 à 27 649 / 0 à -4 864 (voir Représentation des entrées analogiques pour la tension et Représentation des entrées analogiques pour le courant)
Débordement haut / bas (mot de données)	Tension : 32 767 à 32 512 / -32 513 à -32 768 Courant : 32 767 à 32 512 / -4 865 à -32 768 (voir Représentation des entrées analogiques pour la tension et Représentation des entrées analogiques pour le courant)
Tension/courant de tenue maximum	$\pm 35 V$ / $\pm 40 mA$
Lissage	Aucun, faible, moyen ou fort (voir Temps de réponse des entrées analogiques pour le temps de réponse indicielle)
Réjection des bruits	400, 60, 50 ou 10 Hz (voir Temps de réponse des entrées analogiques pour les taux d'échantillonnage)
Précision (25°C / -20 à 60°C)	$\pm 0,3\%$ / $\pm 0,6\%$ de la pleine échelle
Impédance d'entrée	
Mode différentiel	Tension : 220 k $\Omega$ ; courant : 250 $\Omega$
Mode commun	Tension : 55 k $\Omega$ ; courant : 55 k $\Omega$
Comportement au passage de MARCHE à ARRÊT	Dernière valeur ou valeur de remplacement (valeur par défaut 0)
Principe de mesure	Conversion de valeur effective
Réjection en mode commun	400 dB, CC pour 60 Hz
Plage de signal opérationnelle	La tension signal plus mode commun doit être inférieure à +35 V et supérieure à -35 V.
Isolation (site à logique)	Aucune
Longueur de câble (mètres)	100 m, torsadé et blindé

Tableau A- 180 Diagnostic

Caractéristiques techniques	SB 1231 AI x 12 bits
Débordement haut/bas	Oui
Basse tension 24 V-	Non

Tableau A- 181 Schéma de câblage du SB entrée analogique

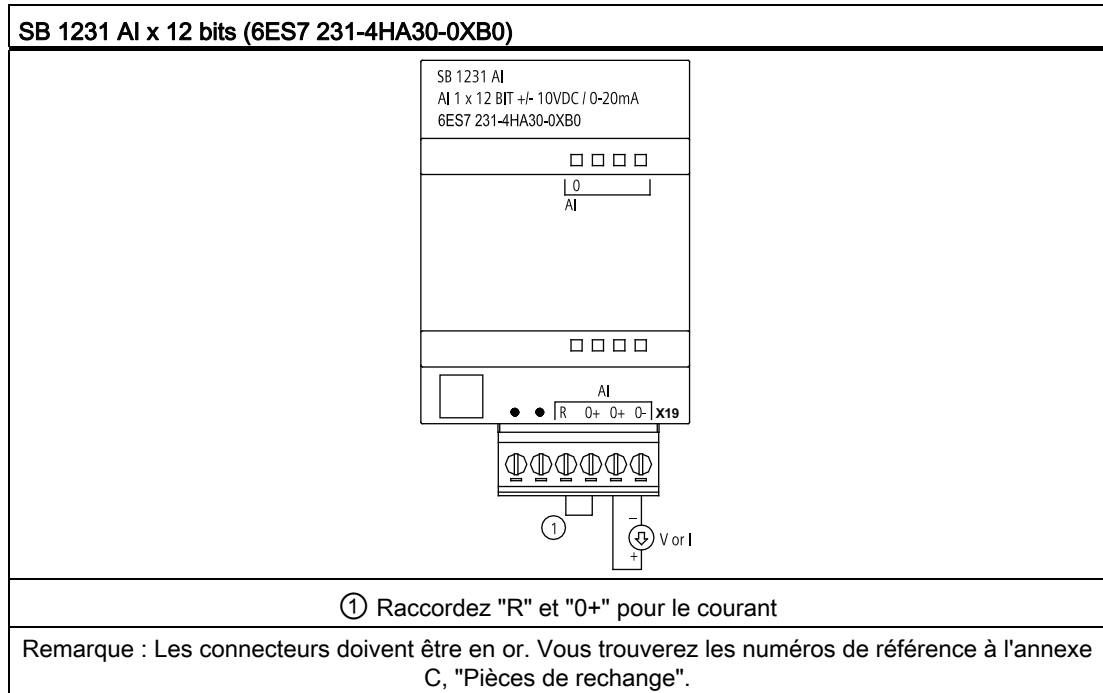


Tableau A- 182 Brochage pour le SB 1231 AI x 12 bits (6ES7 231-4HA30-0XB0)

Broche	X19 (or)
1	Pas de connexion
2	Pas de connexion
3	AI R
4	AI 0+
5	AI 0+
6	AI 0-



## A.10.2 Caractéristiques du SB 1232 1 sortie analogique

Tableau A- 183 Caractéristiques générales

<b>Caractéristiques techniques</b>	<b>SB 1232 AQ 1 x 12 bits</b>
Numéro de référence	6ES7 232-4HA30-0XB0
Dimensions L x H x P (mm)	38 x 62 x 21
Poids	40 grammes
Dissipation de courant	1,5 W
Consommation de courant (bus SM)	15 mA
Consommation de courant (24 V-)	40 mA (sans charge)

Tableau A- 184 Sorties analogiques

<b>Caractéristiques techniques</b>	<b>SB 1232 AQ 1 x 12 bits</b>
Nombre de sorties	1
Type	Tension ou courant
Plage	±10 V ou 0 à 20 mA
Résolution	Tension : 12 bits Courant : 11 bits
Plage pleine échelle (mot de données)	Tension : -27 648 à 27 648
Voir les plages des sorties pour la tension et le courant (Page 853).	Courant : 0 à 27 648
Précision (25°C / -20 à 60°C)	±0,5% / ±1% de la pleine échelle
Temps d'établissement (95% de la nouvelle valeur)	Tension : 300 µs (R), 750 µs (1 µF) Courant : 600 µs (1 mH), 2 ms (10 mH)
Impédance de charge	Tension : ≥ 1000 Ω Courant : ≤ 600 Ω
Comportement au passage de MARCHE à ARRÊT	Dernière valeur ou valeur de remplacement (valeur par défaut 0)
Isolation (site à logique)	Aucune
Longueur de câble (mètres)	100 m, torsadé et blindé

Tableau A- 185 Diagnostic

<b>Caractéristiques techniques</b>	<b>SB 1232 AQ 1 x 12 bits</b>
Débordement haut/bas	Oui
Court-circuit à la terre (mode tension uniquement)	Oui
Rupture de fil (mode courant uniquement)	Oui

Tableau A- 186 Schéma de câblage du SB 1232 AQ 1 x 12 bits

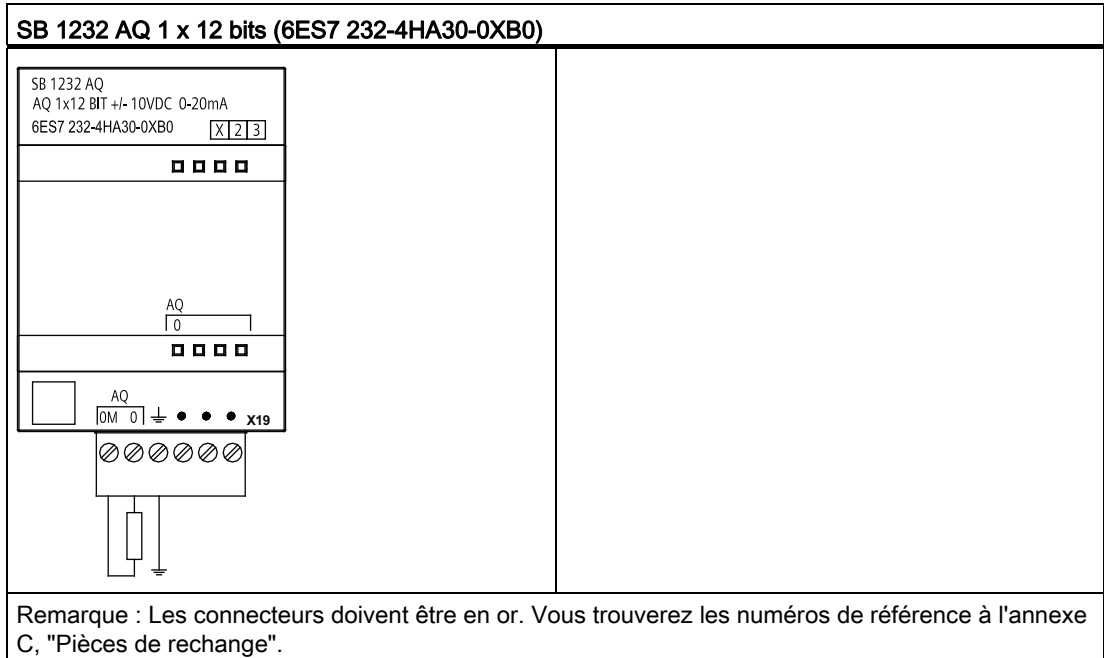


Tableau A- 187 Brochage pour le SB 1232 AQ 1 x 12 bits (6ES7 232-4HA30-0XB0)

Broche	X19 (or)
1	AQ 0M
2	AQ 0
3	GND
4	Pas de connexion
5	Pas de connexion
6	Pas de connexion

## A.10.3 Plages de mesure pour les entrées et sorties analogiques

### A.10.3.1 Réponse indicielle des entrées analogiques

Tableau A- 188 Réponse indicielle (ms), 0 V à 10 V mesuré à 95%

Sélection de lissage (moyennage d'échantillon)	Sélection de temps d'intégration			
	400 Hz (2,5 ms)	60 Hz (16,6 ms)	50 Hz (20 ms)	10 Hz (100 ms)
Aucun (1 cycle) : Pas de moyennage	4,5 ms	18,7 ms	22,0 ms	102 ms
Faible (4 cycles) : 4 échantillons	10,6 ms	59,3 ms	70,8 ms	346 ms
Moyen (16 cycles) : 16 échantillons	33,0 ms	208 ms	250 ms	1240 ms
Fort (32 cycles) : 32 échantillons	63,0 ms	408 ms	490 ms	2440 ms
<b>Temps d'échantillonnage</b>	<b>0,156 ms</b>	<b>1,042 ms</b>	<b>1,250 ms</b>	<b>6,250 ms</b>

### A.10.3.2 Temps d'échantillonnage et temps d'actualisation pour les entrées analogiques

Tableau A- 189 Temps d'échantillonnage et temps d'actualisation

Sélection	Temps d'échantillonnage	Temps d'actualisation du SB
400 Hz (2,5 ms)	0,156 ms	0,156 ms
60 Hz (16,6 ms)	1,042 ms	1,042 ms
50 Hz (20 ms)	1,250 ms	1,25 ms
10 Hz (100 ms)	6,250 ms	6,25 ms

**A.10.3.3 Plages de mesure des entrées analogiques pour la tension**

Tableau A- 190 Représentation des entrées analogiques pour la tension

Technologie		Plage de mesure de tension						
Décimal	Hexadécimal	± 10 V	± 5 V	± 2,5 V	± 1,25V	0 à 10 V		
32767	7FFF	11,851 V	5,926 V	2,963 V	1,481 V	Débordement haut	11,851 V	Débordement haut
32512	7F00							
32511	7EFF	11,759 V	5,879 V	2,940 V	1,470 V	Plage de dépassement haut	11,759 V	Plage de dépassement haut
27649	6C01							
27648	6C00	10 V	5 V	2,5 V	1,250 V	Plage nominale	10 V	Plage nominale
20736	5100	7,5 V	3,75 V	1,875 V	0,938 V		7,5 V	
1	1	361,7 µV	180,8 µV	90,4 µV	45,2 µV		361,7 µV	
0	0	0 V	0 V	0 V	0 V		0 V	
-1	FFFF						Valeurs négatives impossibles	
-20736	AF00	-7,5 V	-3,75 V	-1,875 V	-0,938 V			
-27648	9400	-10 V	-5 V	-2,5 V	-1,250 V			
-27649	93FF							
-32512	8100	-11,759 V	-5,879 V	-2,940 V	-1,470 V	Plage de dépassement bas		
-32513	80FF					Débordement bas		
-32768	8000	-11,851 V	-5,926 V	-2,963 V	-1,481 V			

**A.10.3.4 Plages de mesure des entrées analogiques pour le courant**

Tableau A- 191 Représentation des entrées analogiques pour le courant

Technologie		Plage de mesure de courant			
Décimal	Hexadécimal	0 mA à 20 mA	4 mA à 20 mA		
32767	7FFF	23,70 mA	22,96 mA	Débordement haut	
32512	7F00				
32511	7EFF	23,52 mA	22,81 mA	Plage de dépassement haut	
27649	6C01				
27648	6C00	20 mA	20 mA	Plage nominale	
20736	5100	15 mA	16 mA		
1	1	723,4 nA	4 mA + 578,7 nA		
0	0	0 mA	4 mA		
-1	FFFF				Plage de dépassement bas
-4864	ED00	-3,52 mA	1,185 mA		
-4865	ECFF			Débordement bas	
-32768	8000				

### A.10.3.5 Plages de mesures de sortie (AQ) pour tension et courant (SB et SM)

Tableau A- 192 Représentation des sorties analogiques pour la tension

Technologie		Plage de sortie de tension	
Décimal	Hexadécimal	± 10 V	
32767	7FFF	Voir note 1	Débordement haut
32512	7F00	Voir note 1	
32511	7EFF	11,76 V	Plage de dépassement haut
27649	6C01		
27648	6C00	10 V	Plage nominale
20736	5100	7,5 V	
1	1	361,7 µV	
0	0	0 V	
-1	FFFF	-361,7 µV	
-20736	AF00	-7,5 V	
-27648	9400	-10 V	
-27649	93FF		
-32512	8100	-11,76 V	Plage de dépassement bas
-32513	80FF	Voir note 1	
-32768	8000	Voir note 1	

<sup>1</sup> En cas de dépassement haut ou bas, les sorties analogiques se comporteront selon les propriétés de configuration de l'appareil établies pour le module d'entrées-sorties analogique. Pour le paramètre "Réaction à l'arrêt de la CPU", sélectionnez : Appliquer valeur de remplacement ou Conserver dernière valeur.

Tableau A- 193 Représentation des sorties analogiques pour le courant

Technologie		Plage de sortie de courant	
Décimal	Hexadécimal	0 mA à 20 mA	
32767	7FFF	Voir note 1	Débordement haut
32512	7F00	Voir note 1	
32511	7EFF	23,52 mA	Plage de dépassement haut
27649	6C01		
27648	6C00	20 mA	Plage nominale
20736	5100	15 mA	
1	1	723,4 nA	
0	0	0 mA	

<sup>1</sup> En cas de dépassement haut ou bas, les sorties analogiques se comporteront selon les propriétés de configuration de l'appareil établies pour le module d'entrées-sorties analogique. Pour le paramètre "Réaction à l'arrêt de la CPU", sélectionnez : Appliquer valeur de remplacement ou Conserver dernière valeur.

## A.10.4 SB Thermocouple

### A.10.4.1 Caractéristiques du SB 1231 1 entrée analogique Thermocouple

#### Remarque

Le firmware de votre CPU doit être de version V2.0 ou plus pour que vous puissiez utiliser ce SB.

Tableau A- 194 Caractéristiques générales

Caractéristiques techniques	SB 1231 AI 1 x 16 bits Thermocouple
Numéro de référence	6ES7 231-5QA30-0XB0
Dimensions L x H x P (mm)	38 x 62 x 21
Poids	35 grammes
Dissipation de courant	0,5 W
Consommation de courant (bus SM)	5 mA
Consommation de courant (24 V-)	20 mA

Tableau A- 195 Entrées analogiques

Caractéristiques techniques	SB 1231 AI 1x16 bits Thermocouple
Nombre d'entrées	1
Type	TC flottant et mV
Plage	Voir le tableau de sélection de filtre pour thermocouples (Page 855).
<ul style="list-style-type: none"> <li>• Plage nominale (mot de données)</li> <li>• Dépassement haut/bas (mot de données)</li> <li>• Débordement haut / bas (mot de données)</li> </ul>	
Résolution	0,1° C / 0,1° F
	Tension 15 bits plus signe
Tension de tenue maximum	± 35 V
Réjection des bruits	85 dB pour le réglage de filtre sélectionné (10 Hz, 50 Hz, 60 Hz, 400 Hz)
Réjection en mode commun	> 120 dB à 120 V~
Impédance	≥ 10 M Ω
Précision	Voir le tableau de sélection pour thermocouples (Page 855).
Répétabilité	± 0,05% pleine échelle
Principe de mesure	Intégration
Temps d'actualisation du module	Voir le tableau de sélection de filtre pour thermocouples (Page 855).
Erreur de soudure froide	±1,5° C

Caractéristiques techniques	SB 1231 AI 1x16 bits Thermocouple
Isolation (site à logique)	500 V~
Longueur de câble (mètres)	max. 100 m au capteur
Résistance de fil	100 $\Omega$ max.

Tableau A- 196 Diagnostic

Caractéristiques techniques	SB 1231 AI 1 x 16 bits Thermocouple
Débordement haut/bas <sup>1</sup>	Oui
Rupture de fil <sup>2</sup>	Oui

<sup>1</sup> Les informations d'alarmes de diagnostic de débordement haut et bas sont notifiées dans les valeurs de données analogiques même si les alarmes sont désactivées dans la configuration du module.

<sup>2</sup> Lorsque l'alarme de rupture de fil est désactivée et qu'un fil est ouvert dans le câblage du capteur, le module peut émettre des valeurs aléatoires.

Le module d'entrées analogiques SM 1231 Thermocouple (TC) mesure la valeur de tension connectée aux entrées du module.

Le Signal Board analogique SB 1231 Thermocouple mesure la valeur de tension connectée aux entrées du Signal Board. Le type de mesure de température peut être soit "Thermocouple", soit "Tension".

- "Thermocouple" : La valeur sera notifiée en degrés multipliés par dix (par exemple, 25,3 degrés seront indiqués sous la forme 253 en décimal).
- "Tension" : La valeur pleine échelle de la plage nominale sera 27648 en décimal.

#### A.10.4.2 Principe de fonctionnement des thermocouples

Un thermocouple est composé de deux conducteurs métalliques de nature différente, reliés électriquement par soudure l'un à l'autre. Une tension proportionnelle à la température de la soudure froide est générée. Cette tension est faible ; un microvolt peut représenter beaucoup de degrés. Mesurer la tension provenant d'un thermocouple, compenser les soudures supplémentaires, puis linéariser le résultat constituent le principe fondamental des mesures de température à l'aide de thermocouples.

Lorsque vous raccordez un thermocouple au module SM 1231 Thermocouple, les deux fils métalliques de nature différente sont reliés au module, au niveau du connecteur de signaux du module. L'endroit où les deux fils sont reliés l'un à l'autre constitue le point de mesure du thermocouple.

Deux autres thermocouples sont constitués à l'endroit où les deux fils de nature différente sont raccordés au connecteur de signaux. La température du connecteur provoque une tension qui s'ajoute à la tension du thermocouple capteur. Si l'on ne corrige pas cette tension, la température indiquée sera différente de la température du capteur.

On utilise donc la compensation de soudure froide pour compenser le thermocouple du connecteur. Les tables de données de thermocouples se basent sur une température de soudure de référence, 0 degré Celsius en général. La compensation de soudure froide compense le connecteur à 0 degré Celsius. Elle restaure la tension ajoutée par les thermocouples du connecteur. La température du module est mesurée en interne, puis convertie en une valeur à ajouter à la conversion de capteur. On linéarise ensuite la conversion de capteur corrigée à l'aide des tables de thermocouples.

Pour un fonctionnement optimal de la compensation de soudure froide, le module Thermocouple doit se trouver dans un environnement thermique stable. Une variation lente (inférieure à 0,1 °C/minute) à une température ambiante du module est correctement compensée dans les limites des spécifications du module. Un déplacement d'air à travers le module entraînera également des erreurs de compensation de la soudure froide.

Si une meilleure compensation d'erreur de soudure froide est souhaitée, un bornier isotherme externe peut être utilisé. Le module Thermocouple met à disposition un bornier référencé à 0 °C ou à 50 °C.

### Tableau de sélection pour le SB 1231 Thermocouple

Les plages et la précision des différents types de thermocouple pris en charge par le Signal Board SB 1231 Thermocouple sont présentées dans le tableau ci-dessous.

Tableau A- 197 Tableau de sélection du SB 1231 Thermocouple

Type de thermocouple	Dépassement bas minimum <sup>1</sup>	Limite basse plage nominale	Limite haute plage nominale	Dépassement haut maximum <sup>2</sup>	Précision <sup>3</sup> plage normale à 25°C	Précision <sup>3</sup> plage normale de -20°C à 60°C
J	-210,0°C	-150,0°C	1200,0°C	1450,0°C	± 0,3°C	± 0,6°C
K	-270,0°C	-200,0°C	1372,0°C	1622,0°C	± 0,4°C	± 1,0°C
T	-270,0°C	-200,0°C	400,0°C	540,0°C	± 0,5°C	± 1,0°C
E	-270,0°C	-200,0°C	1000,0°C	1200,0°C	± 0,3°C	± 0,6°C
R & S	-50,0°C	100,0°C	1768,0°C	2019,0°C	± 1,0°C	± 2,5°C
N	-270,0°C	0,0°C	1300,0°C	1550,0°C	± 1,0°C	± 1,6°C
C	0,0°C	100,0°C	2315,0°C	2500,0°C	± 0,7°C	± 2,7°C
TXK/XK(L)	-200,0°C	-150,0°C	800,0°C	1050,0°C	± 0,6°C	± 1,2°C
Tension	-32511	-27648 -80mV	27648 80mV	32511	±0.05%	±0.1%

<sup>1</sup> Les valeurs de thermocouple inférieures à la valeur de dépassement bas minimum sont signalées en tant que -32768.

<sup>2</sup> Les valeurs de thermocouple supérieures à la valeur de dépassement haut maximum sont signalées en tant que 32767.

<sup>3</sup> L'erreur de soudure froide interne est de ±1,5°C pour toutes les plages. Cette valeur s'ajoute à l'erreur indiquée dans ce tableau. Le Signal Board requiert au minimum 30 minutes d'échauffement pour atteindre cette spécification.



Tableau A- 198 Tableau de sélection de filtre pour le SB 1231 Thermocouple

Fréquence de réjection (Hz)	Temps d'intégration (ms)	Temps d'actualisation du Signal Board (secondes)
10	100	0.306
50	20	0.066
60	16.67	0.056
400 <sup>1</sup>	10	0.036

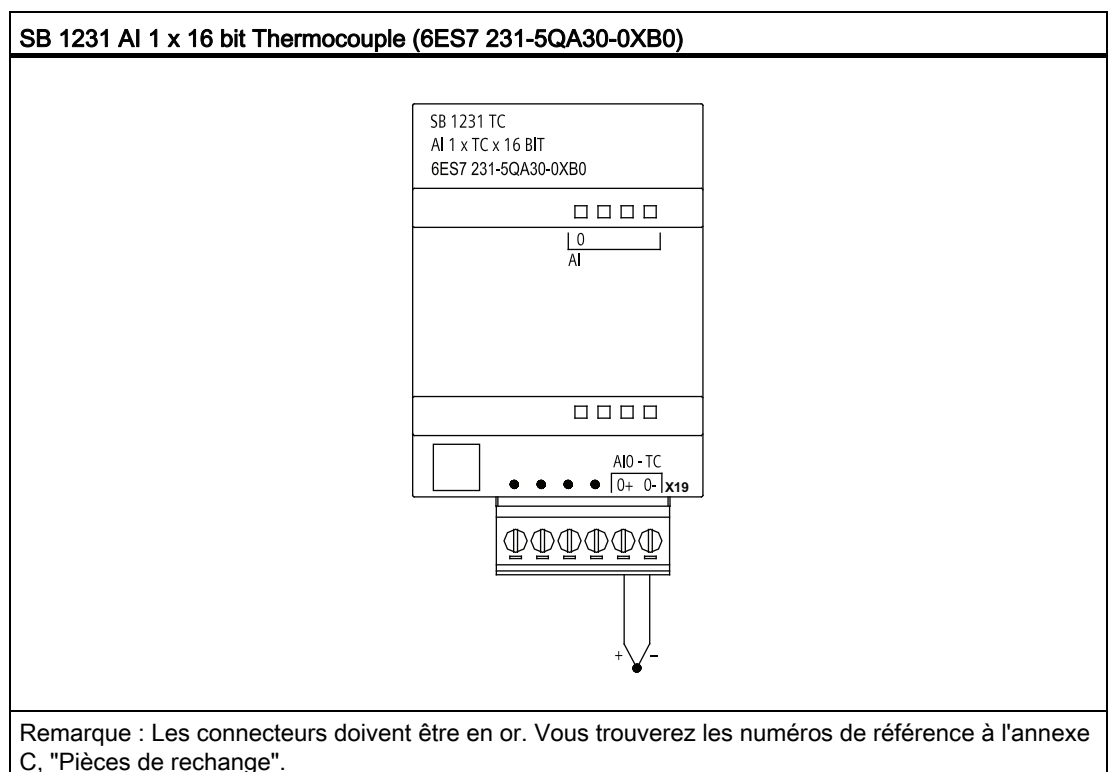
<sup>1</sup> Le temps d'intégration est de 10 ms pour conserver la résolution et la précision du module lorsque la réjection 400 Hz est sélectionnée. Cette sélection correspond également à une réjection des bruits 100 Hz et 200 Hz.

Il est recommandé d'utiliser un temps d'intégration de 100 ms pour la mesure des thermocouples. Le choix de temps d'intégration inférieurs augmentera l'erreur de répétabilité des températures lues.

**Remarque**

Après sa mise sous tension, le module exécute un calibrage interne pour le convertisseur analogique-numérique. Pendant ce temps, il émet une valeur de 32767 sur chaque voie jusqu'à ce que des données valables soient disponibles sur cette voie. Il peut s'avérer nécessaire que votre programme utilisateur prévoie ce temps d'initialisation.

Tableau A- 199 Schéma de câblage du SB entrée analogique Thermocouple



A.10 Signal Boards (SB) analogiques

Tableau A- 200 Brochage pour le SB 1231 AI 1 x 16 bits Thermocouple (6ES7 231-5QA30-0XB0)

Broche	X19 (or)
1	Pas de connexion
2	Pas de connexion
3	Pas de connexion
4	Pas de connexion
5	AI 0- /TC
6	AI 0+ /TC

A.10.5 Signal Boards (SB) RTD

A.10.5.1 Caractéristiques du SB 1231 1 entrée analogique RTD

Remarque

Le firmware de votre CPU doit être de version V2.0 ou plus pour que vous puissiez utiliser ce SB.

Tableau A- 201 Caractéristiques générales

Caractéristiques techniques	SB 1231 AI 1 x 16 bits RTD
Numéro de référence	6ES7 231-5PA30-0XB0
Dimensions L x H x P (mm)	38 x 62 x 2
Poids	35 grammes
Dissipation de courant	0,7 W
Consommation de courant (bus SM)	5 mA
Consommation de courant (24 V-)	25 mA

Tableau A- 202 Entrées analogiques

Caractéristiques techniques	SB 1231 AI 1 x 16 bits RTD
Nombre d'entrées	1
Type	RTD référencée au module et ohms
Plage	Voir les tableaux de sélection (Page 861).
<ul style="list-style-type: none"> <li>• Plage nominale (mot de données)</li> <li>• Dépassement haut/bas (mot de données)</li> <li>• Débordement haut / bas (mot de données)</li> </ul>	
Résolution	Température 0,1° C / 0,1° F

<b>Caractéristiques techniques</b>	<b>SB 1231 AI 1 x 16 bits RTD</b>
Tension	15 bits plus signe
Tension de tenue maximum	± 35 V
Réjection des bruits	85 dB (10 Hz, 50 Hz, 60 Hz, 400 Hz)
Réjection en mode commun	> 120 dB
Impédance	≥ 10 MΩ
Précision	Voir les tableaux de sélection (Page 861).
Répétabilité	± 0,05% pleine échelle
Dissipation maximale du capteur	0,5 mW
Principe de mesure	Intégration
Temps d'actualisation du module	Voir le tableau de sélection (Page 861).
Isolation (site à logique)	500 V~
Longueur de câble (mètres)	max. 100 m au capteur
Résistance de fil	20 Ω, 2,7 pour RTD 10 Ω max.

Tableau A- 203 Diagnostic

<b>Caractéristiques techniques</b>	<b>SB 1231 AI 1 x 16 bits RTD</b>
Dépassement haut / bas <sup>1,2</sup>	Oui
Rupture de fil <sup>3</sup>	Oui

<sup>1</sup> Les informations d'alarmes de diagnostic de débordement haut et bas sont notifiées dans les valeurs de données analogiques même si les alarmes sont désactivées dans la configuration du module.

<sup>2</sup> La détection de débordement bas n'est jamais activée pour les plages de résistance.

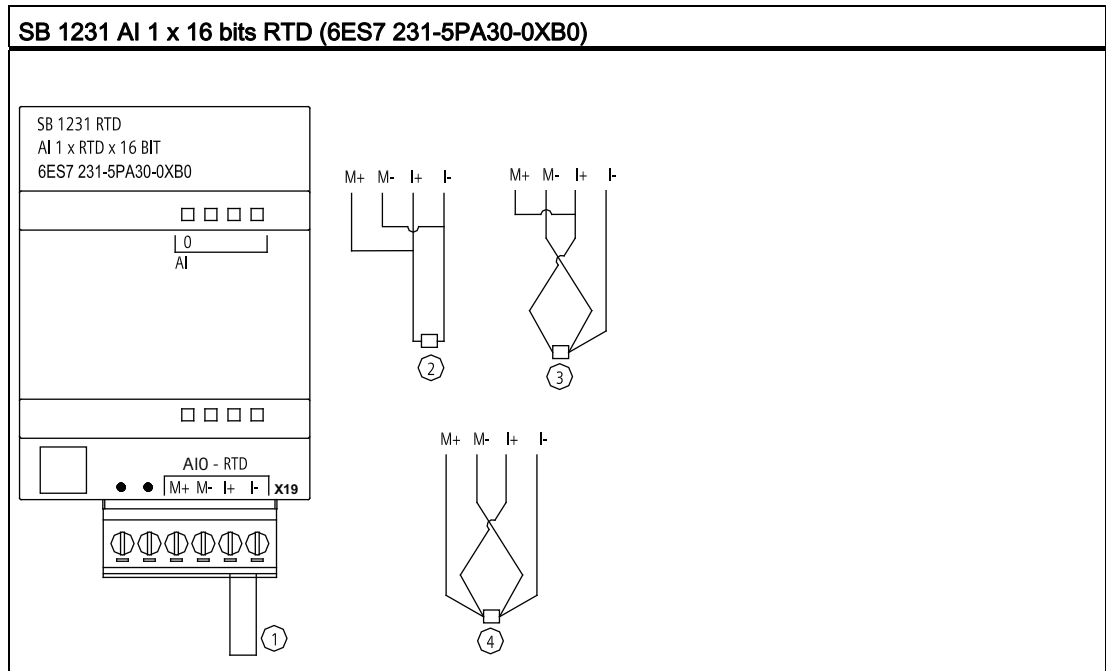
<sup>3</sup> Lorsque l'alarme de rupture de fil est désactivée et qu'un fil est ouvert dans le câblage du capteur, le module peut émettre des valeurs aléatoires.

Le Signal Board analogique SB 1231 RTD mesure la valeur de résistance connectée aux entrées du Signal Board. Il est possible de sélectionner "Résistance" ou "Thermistance" comme type de mesure.

- "Résistance" : La valeur pleine échelle de la plage nominale sera 27648 en décimal.
- "Thermistance" : La valeur sera notifiée en degrés multipliés par dix (par exemple, 25,3 degrés seront indiqués sous la forme 253 en décimal).

Le Signal Board SB 1231 RTD prend en charge des montages 2 fils, 3 fils et 4 fils avec la résistance du capteur.

Tableau A- 204 Schéma de câblage du SB 1231 AI 1 x 16 bits RTD



- ① Bouclage sur l'entrée RTD inutilisée
- ② RTD 2 fils
- ② RTD 3 fils
- ② RTD 4 fils

Remarque : Les connecteurs doivent être en or. Vous trouverez les numéros de référence à l'annexe C, "Pièces de rechange".

Tableau A- 205 Brochage pour le SB 1231 AI 1 x 16 bits RTD (6ES7 231-5PA30-0XB0)

Broche	X19 (or)
1	Pas de connexion
2	Pas de connexion
3	AI 0 M+ /RTD
4	AI 0 M- /RTD
5	AI 0 I+ /RTD
6	AI 0 I- /RTD

## A.10.5.2 Tableaux de sélection pour le SB 1231 RTD

Tableau A- 206 Plages et précision pour les différents capteurs pris en charge par les modules RTD

Coefficient de température	Type de RTD	Dépassement bas minimum <sup>1</sup>	Limite basse plage nominale	Limite haute plage nominale	Dépassement haut maximum <sup>2</sup>	Précision de plage normale à 25°C	Précision de plage normale de -20°C à 60°C
Pt 0.003850 ITS90 DIN EN 60751	Pt 10	-243,0°C	-200,0°C	850,0°C	1000,0°C	± 1,0°C	± 2,0°C
	Pt 50	-243,0°C	-200,0°C	850,0°C	1000,0°C	± 0,5°C	± 1,0°C
	Pt 100						
	Pt 200						
	Pt 500						
Pt 0.003902 Pt 0.003916 Pt 0.003920	Pt 100	-243,0°C	-200,0°C	850,0°C	1000,0°C	± 0,5°C	± 1,0°C
	Pt 200						
	Pt 500						
	Pt 1000						
Pt 0,003910	Pt 10	-273,2°C	-240,0°C	1100,0°C	1295°C	± 1,0°C	± 2,0°C
	Pt 50	-273,2°C	-240,0°C	1100,0°C	1295°C	± 0,8°C	± 1,6°C
	Pt 100						
	Pt 500						
Ni 0.006720 Ni 0.006180	Ni 100	-105,0°C	-60,0°C	250,0°C	295,0°C	± 0,5°C	± 1,0°C
	Ni 120						
	Ni 200						
	Ni 500						
	Ni 1000						
LG-Ni 0.005000	LG-Ni 1000	-105,0°C	-60,0°C	250,0°C	295,0°C	± 0,5°C	± 1,0°C
Ni 0.006170	Ni 100	-105,0°C	-60,0°C	180,0°C	212,4°C	± 0,5°C	± 1,0°C
Cu 0.004270	Cu 10	-240,0°C	-200,0°C	260,0°C	312,0°C	± 1,0°C	± 2,0°C
Cu 0.004260	Cu 10	-60,0°C	-50,0°C	200,0°C	240,0°C	± 1,0°C	± 2,0°C
	Cu 50	-60,0°C	-50,0°C	200,0°C	240,0°C	± 0,6°C	± 1,2°C
	Cu 100						
Cu 0.004280	Cu 10	-240,0°C	-200,0°C	200,0°C	240,0°C	± 1,0°C	± 2,0°C
	Cu 50	-240,0°C	-200,0°C	200,0°C	240,0°C	± 0,7°C	± 1,4°C
	Cu 100						

<sup>1</sup> Les valeurs RTD inférieures à la valeur de dépassement bas minimum sont signalées en tant que -32768.

<sup>2</sup> Les valeurs RTD supérieures à la valeur de dépassement haut maximum sont signalées en tant que +32768.

Tableau A- 207 Résistance

Plage	Dépassement bas minimum	Limite basse plage nominale	Limite haute plage nominale	Dépassement haut maximum <sup>1</sup>	Précision de plage normale à 25°C	Précision de plage normale de -20°C à 60°C
150 Ω	Sans objet	0 (0 Ω)	27648 (150 Ω)	176,383 Ω	±0.05%	±0.1%
300 Ω	Sans objet	0 (0 Ω)	27648 (300 Ω)	352,767 Ω	±0.05%	±0.1%
600 Ω	Sans objet	0 (0 Ω)	27648 (600 Ω)	705,534 Ω	±0.05%	±0.1%

<sup>1</sup> Les valeurs de résistance supérieures à la valeur de dépassement haut maximum sont signalées comme 32767.

**Remarque**

Le module signale 32767 sur toute voie activée à laquelle aucun capteur n'est connecté. Si la détection de fil ouvert est également activée, le module fait clignoter les DEL appropriées.

Si des plages RTD 500 Ω et 1000 Ω sont utilisées avec d'autres résistances à valeur inférieure, l'erreur peut atteindre deux fois l'erreur indiquée.

La meilleure précision est obtenue pour les plages RTD 10 Ω avec des montages 4 fils.

La résistance des fils de connexion en montage 2 fils provoquera une erreur dans la lecture du capteur. La précision ne peut donc pas être garantie.

Tableau A- 208 Réduction des bruits et temps d'actualisation pour les modules RTD

Sélection de la fréquence de réjection	Temps d'intégration	Module 1 voie, 4 / 2 fils Temps d'actualisation (secondes)	Module 1 voie, 3 fils Temps d'actualisation (secondes)
400 Hz (2,5 ms)	10 ms <sup>1</sup>	0.036	0.071
60 Hz (16,6 ms)	16,67 ms	0.056	0.111
50 Hz (20 ms)	20 ms	0.066	1.086
10 Hz (100 ms)	100 ms	0.306	0.611

<sup>1</sup> Le temps d'intégration est de 10 ms pour conserver la résolution et la précision du module lorsque le filtre 400 Hz est sélectionné. Cette sélection correspond également à une réjection des bruits 100 Hz et 200 Hz.

**IMPORTANT**

Après sa mise sous tension, le module exécute un calibrage interne pour le convertisseur analogique-numérique. Pendant ce temps, il émet une valeur de 32767 sur chaque voie jusqu'à ce que des données valables soient disponibles sur cette voie. Il peut s'avérer nécessaire que votre programme utilisateur prévoie ce temps d'initialisation. Comme la configuration du module peut faire varier la longueur du temps d'initialisation, vous devez vérifier le comportement du module dans votre configuration. Si nécessaire, vous pouvez inclure de la logique dans votre programme utilisateur afin de prendre en compte le temps d'initialisation du module.

## A.11 Battery Board BB 1297

### Battery Board BB 1297

Le Battery Board BB 1297 S7-1200 est conçu pour une sauvegarde à long terme de l'horloge temps réel. Il s'enfiche dans le logement pour Signal Board de la CPU S7-1200 (firmware 3.0 et versions ultérieures). Pour que le BB 1297 soit opérationnel, vous devez l'ajouter à la configuration des appareils et charger dans la CPU la configuration matérielle pour le BB.

La pile (de type CR1025) n'est pas fournie avec le BB 1297 et doit être achetée par l'utilisateur.

---

#### Remarque

La conception mécanique du BB 1297 est adaptée aux CPU de firmware 3.0 et plus.

N'utilisez pas le BB 1297 avec des CPU de version antérieure, car son connecteur ne rentrera pas dans la CPU.

---

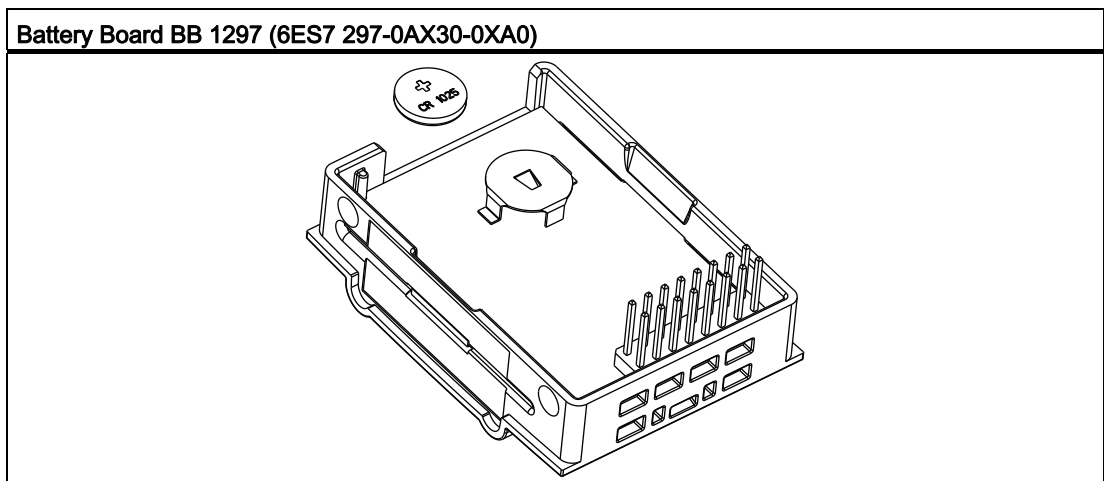
Tableau A- 209 Caractéristiques générales

Caractéristiques techniques	Battery Board BB 1297
Numéro de référence	6ES7 297-0AX30-0XA0
Dimensions L x H x P (mm)	38 x 62 x 21
Poids	28 grammes
Dissipation de courant	0,5 W
Consommation de courant (bus SM)	11 mA
Consommation de courant (24 V-)	aucune

Pile (non incluse)	Battery Board BB 1297
Temps de conservation	Environ 1 an
Type de pile	CR1025 ; voir Installation ou remplacement de la pile dans le Battery Board BB 1297 (Page 53)
Tension nominale	3 V
Capacité nominale	Au moins 30 mAh

Diagnostic	Battery Board BB 1297
Niveau de pile critique	< 2,5 V
Diagnostic de pile	Indicateur de tension basse : <ul style="list-style-type: none"> <li>• Lorsque la tension de la pile est faible, la DEL de maintenance de la CPU s'allume en continu (orange).</li> <li>• Événement de tampon de diagnostic : 16#06:2700 "Requête de maintenance de sous-module : Au moins une pile est épuisée (BATTF)"</li> </ul>
Etat de la pile	Bit d'état de pile fourni 0 = pile OK 1 = pile faible
Actualisation de l'état de la pile	L'état de la pile est actualisé à la mise sous tension, puis une fois par jour lorsque la CPU est à l'état MARCHE.

Tableau A- 210 Schéma d'insertion pour le Battery Board BB 1297



## A.12 Interfaces de communication

### A.12.1 PROFIBUS

#### A.12.1.1 CM 1242-5

Tableau A- 211 Caractéristiques techniques du CM 1242-5

Caractéristiques techniques	
Numéro de référence	6GK7 242-5DX30-0XE0
Interfaces	



### Caractéristiques techniques

Connexion à PROFIBUS	Connecteur femelle Sub-D 9 points
Courant maximal absorbé par l'interface PROFIBUS lors de la connexion de constituants de réseau (constituants de réseau optiques par exemple)	15 mA à 5 V (uniquement pour terminaison du bus) *)

### Conditions ambiantes admissibles

Température ambiante	
<ul style="list-style-type: none"> <li>• pendant le stockage</li> <li>• pendant le transport</li> <li>• en fonctionnement, installé verticalement (sur rail symétrique horizontal)</li> <li>• en fonctionnement, installé horizontalement (sur rail symétrique vertical)</li> </ul>	<ul style="list-style-type: none"> <li>• -40 °C ... 70 °C</li> <li>• -40 °C ... 70 °C</li> <li>• 0 °C ... 55 °C</li> <li>• 0 °C ... 45 °C</li> </ul>

Humidité relative maximale de l'air à 25 °C, en fonctionnement, sans condensation	95 %
-----------------------------------------------------------------------------------	------

Degré de protection	IP20
---------------------	------

### Alimentation électrique, consommation de courant, puissance dissipée

Type d'alimentation	DC
Alimentation par le bus de fond de panier	5 V
Consommation de courant (typique)	150 mA
Puissance dissipée (typique)	0,75 W

### Dimensions et poids

<ul style="list-style-type: none"> <li>• Largeur</li> <li>• Hauteur</li> <li>• Profondeur :</li> </ul>	<ul style="list-style-type: none"> <li>• 30 mm</li> <li>• 100 mm</li> <li>• 75 mm</li> </ul>
--------------------------------------------------------------------------------------------------------	----------------------------------------------------------------------------------------------

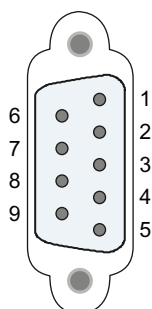
### Poids

<ul style="list-style-type: none"> <li>• Poids net</li> <li>• Poids emballage compris</li> </ul>	<ul style="list-style-type: none"> <li>• 115 g</li> <li>• 152 g</li> </ul>
--------------------------------------------------------------------------------------------------	----------------------------------------------------------------------------

\*) Le courant absorbé par un récepteur externe connecté entre VP (broche 6) et DGND (broche 5) ne doit pas dépasser, pour des raisons de terminaison du bus, 15 mA (résistance aux courts-circuits).

## Interface PROFIBUS

Tableau A- 212 Brochage du connecteur femelle Sub-D



Broche	Description	Broche	Description
1	- non affectée -	6	P5V2 : Alimentation électrique +5V
2	- non affectée -	7	- non affectée -
3	RxD/TxD-P : conducteur de données B	8	RxD/TxD-N : conducteur de données A
4	RTS	9	- non affectée -
5	M5V2 : potentiel de référence des données (Masse DGND)	Boîtier	Mise à la terre

**A.12.1.2 CM 1243-5**

Tableau A- 213 Caractéristiques techniques du CM 1243-5

<b>Caractéristiques techniques</b>	
Numéro de référence	6GK7 243-5DX30-0XE0
<b>Interfaces</b>	
Connexion à PROFIBUS	Connecteur femelle Sub-D 9 points
Courant maximal absorbé par l'interface PROFIBUS lors de la connexion de constituants de réseau (constituants de réseau optiques par exemple)	15 mA à 5 V (uniquement pour terminaison du bus) *)
<b>Conditions ambiantes admissibles</b>	
Température ambiante	
<ul style="list-style-type: none"> <li>• pendant le stockage</li> <li>• pendant le transport</li> <li>• en fonctionnement, installé verticalement (sur rail symétrique horizontal)</li> <li>• en fonctionnement, installé horizontalement (sur rail symétrique vertical)</li> </ul>	<ul style="list-style-type: none"> <li>• -40 °C ... 70 °C</li> <li>• -40 °C ... 70 °C</li> <li>• 0 °C ... 55 °C</li> <li>• 0 °C ... 45 °C</li> </ul>
Humidité relative maximale de l'air à 25 °C, en fonctionnement, sans condensation	95 %
Degré de protection	IP20
<b>Alimentation électrique, consommation de courant, puissance dissipée</b>	
Type d'alimentation	DC
Tension d'alimentation / externe	24 V
<ul style="list-style-type: none"> <li>• minimale</li> <li>• maximale</li> </ul>	<ul style="list-style-type: none"> <li>• 19,2 V</li> <li>• 28,8 V</li> </ul>
Consommation de courant (typique)	
<ul style="list-style-type: none"> <li>• sous DC 24 V</li> <li>• au niveau du bus de fond de panier du S7-1200</li> </ul>	<ul style="list-style-type: none"> <li>• 100 mA</li> <li>• 0 mA</li> </ul>
Puissance dissipée (typique)	
<ul style="list-style-type: none"> <li>• sous DC 24 V</li> <li>• au niveau du bus de fond de panier du S7-1200</li> </ul>	<ul style="list-style-type: none"> <li>• 2,4 W</li> <li>• 0 W</li> </ul>
Alimentation DC 24 V / externe	
<ul style="list-style-type: none"> <li>• Section de conducteur min.</li> <li>• Section de conducteur max.</li> <li>• Couple de serrage des bornes à vis</li> </ul>	<ul style="list-style-type: none"> <li>• min. : 0,14 mm<sup>2</sup> (AWG 25)</li> <li>• max. : 1,5 mm<sup>2</sup> (AWG 15)</li> <li>• 0,45 Nm (4 lb.in.)</li> </ul>
<b>Dimensions et poids</b>	
<ul style="list-style-type: none"> <li>• Largeur</li> <li>• Hauteur</li> <li>• Profondeur :</li> </ul>	<ul style="list-style-type: none"> <li>• 30 mm</li> <li>• 100 mm</li> <li>• 75 mm</li> </ul>

### Caractéristiques techniques

#### Poids

- |                           |         |
|---------------------------|---------|
| • Poids net               | • 134 g |
| • Poids emballage compris | • 171 g |

\*) Le courant absorbé par un récepteur externe connecté entre VP (broche 6) et DGND (broche 5) ne doit pas dépasser, pour des raisons de terminaison du bus, 15 mA (résistance aux courts-circuits).

#### Remarque

Le CM 1243-5 (module maître PROFIBUS) doit être alimenté en courant par l'alimentation capteur 24 V- de la CPU.

## Interface PROFIBUS

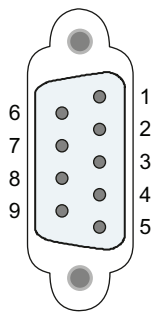


Tableau A- 214 Brochage du connecteur femelle Sub-D

Broche	Description	Broche	Description
1	- non affectée -	6	VP : Alimentation +5 V uniquement pour résistances de terminaison de bus ; pas pour l'alimentation d'appareils externes
2	- non affectée -	7	- non affectée -
3	RxD/TxD-P : conducteur de données B	8	RxD/TxD-N : conducteur de données A
4	CNTR-P : RTS	9	- non affectée -
5	DGND : Masse des signaux de données et VP	Boîtier	Mise à la terre

## Câble PROFIBUS

### IMPORTANT

#### Connexion du blindage du câble PROFIBUS

Le blindage du câble PROFIBUS doit être connecté.

Dénudez pour ce faire l'extrémité du câble PROFIBUS et connectez le blindage à la terre fonctionnelle.

## A.12.2 GPRS

---

### Remarque

**Le CP GPRS n'est pas homologué pour le domaine maritime**

Le module suivant ne dispose pas de l'homologation pour le domaine maritime :

- Module CP 1242-7 GPRS
- 

### Remarque

Le firmware de votre CPU doit être de version V2.0 ou plus pour que vous puissiez utiliser ces modules.

---

### A.12.2.1 CP 1242-7

Tableau A- 215 Caractéristiques techniques du CP 1242-7

<b>Caractéristiques techniques</b>	
Numéro de référence	6GK7 242-7KX30-0XE0
<b>Interface radio</b>	
Connexion d'antenne	connecteur femelle SMA
Impédance nominale	50 ohms
<b>Liaison radio</b>	
Puissance émettrice maximale	<ul style="list-style-type: none"><li>• GSM 850, Class 4: +33 dBm ±2dBm</li><li>• GSM 900, Class 4: +33 dBm ±2dBm</li><li>• GSM 1800, Class 1: +30 dBm ±2dBm</li><li>• GSM 1900, Class 1: +30 dBm ±2dBm</li></ul>
GPRS	Classe Multislot 10 Classe d'équipement terminal B Schéma de codage 1...4 (GMSK)
SMS	Mode opératoire partant : MO Service : point à point
<b>Conditions ambiantes admissibles</b>	
Température ambiante	
<ul style="list-style-type: none"><li>• pendant le stockage</li><li>• pendant le transport</li><li>• en fonctionnement, installé verticalement (sur rail symétrique horizontal)</li><li>• en fonctionnement, installé horizontalement (sur rail symétrique vertical)</li></ul>	<ul style="list-style-type: none"><li>• -40 °C ... 70 °C</li><li>• -40 °C ... 70 °C</li><li>• 0 °C ... 55 °C</li><li>• 0 °C ... 45 °C</li></ul>
Humidité relative maximale de l'air à 25 °C, en fonctionnement, sans condensation	95 %
Degré de protection	IP20

---

### Caractéristiques techniques

---

#### Alimentation électrique, consommation de courant, puissance dissipée

---

Type d'alimentation	DC
---------------------	----

Tension d'alimentation / externe	24 V
----------------------------------	------

- |                                                                                  |                                                                              |
|----------------------------------------------------------------------------------|------------------------------------------------------------------------------|
| <ul style="list-style-type: none"> <li>• minimale</li> <li>• maximale</li> </ul> | <ul style="list-style-type: none"> <li>• 19,2 V</li> <li>• 28,8 V</li> </ul> |
|----------------------------------------------------------------------------------|------------------------------------------------------------------------------|

Consommation de courant (typique)	
-----------------------------------	--

- |                                                                                                                           |                                                                            |
|---------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------|
| <ul style="list-style-type: none"> <li>• sous DC 24 V</li> <li>• au niveau du bus de fond de panier du S7-1200</li> </ul> | <ul style="list-style-type: none"> <li>• 100 mA</li> <li>• 0 mA</li> </ul> |
|---------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------|

Puissance dissipée (typique)	
------------------------------	--

- |                                                                                                                           |                                                                          |
|---------------------------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------|
| <ul style="list-style-type: none"> <li>• sous DC 24 V</li> <li>• au niveau du bus de fond de panier du S7-1200</li> </ul> | <ul style="list-style-type: none"> <li>• 2,4 W</li> <li>• 0 W</li> </ul> |
|---------------------------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------|

Alimentation DC 24 V	
----------------------	--

- |                                                                                                                                                                    |                                                                                                                                                                     |
|--------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <ul style="list-style-type: none"> <li>• Section de conducteur min.</li> <li>• Section de conducteur max.</li> <li>• Couple de serrage des bornes à vis</li> </ul> | <ul style="list-style-type: none"> <li>• min. : 0,14 mm<sup>2</sup> (AWG 25)</li> <li>• max. : 1,5 mm<sup>2</sup> (AWG 15)</li> <li>• 0,45 Nm (4 lb.in.)</li> </ul> |
|--------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------|

#### Dimensions et poids

---

- |                                                                                                        |                                                                                              |
|--------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------|
| <ul style="list-style-type: none"> <li>• Largeur</li> <li>• Hauteur</li> <li>• Profondeur :</li> </ul> | <ul style="list-style-type: none"> <li>• 30 mm</li> <li>• 100 mm</li> <li>• 75 mm</li> </ul> |
|--------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------|

Poids	
-------	--

- |                                                                                                  |                                                                            |
|--------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------|
| <ul style="list-style-type: none"> <li>• Poids net</li> <li>• Poids emballage compris</li> </ul> | <ul style="list-style-type: none"> <li>• 133 g</li> <li>• 170 g</li> </ul> |
|--------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------|
- 

### Caractéristiques de l'antenne GSM/GPRS ANT794-4MR

---

#### ANT794-4MR

---

Numéro de référence	6NH9860-1AA00
---------------------	---------------

Réseaux de téléphonie mobile	GSM / GPRS
------------------------------	------------

Bandes passantes	<ul style="list-style-type: none"> <li>• 824...960 MHz (GSM 850, 900)</li> <li>• 1 710...1 880 MHz (GSM 1 800)</li> <li>• 1 900...2 200 MHz (GSM / UMTS)</li> </ul>
------------------	---------------------------------------------------------------------------------------------------------------------------------------------------------------------

Caractéristique	omnidirectionnel
-----------------	------------------

Gain d'antenne	0 dB
----------------	------

Impédance	50 ohm
-----------	--------

Rapport d'ondes stationnaires (ROS)	< 2,0
-------------------------------------	-------

Puissance max.	20 W
----------------	------

Polarité	linéaire verticale
----------	--------------------

Connecteur	SMA
------------	-----

---

<b>ANT794-4MR</b>	
Longueur du câble d'antenne	5 m
Enveloppe	PVC dur, résistant aux UV
Degré de protection	IP20
Conditions ambiantes admissibles	
• Température de service	• -40 °C à +70 °C
• Température de transport/stockage	• -40 °C à +70 °C
• Humidité relative	• 100 %
Enveloppe	PVC dur, résistant aux UV
Caractéristiques de conception	Antenne équipée d'un câble de 5 m et d'un connecteur mâle SMA
Dimensions (D x H) en mm :	25 x 193
Poids	
• Antenne avec câble	• 310 g
• Matériel de montage	• 54 g
Montage	Sur équerre fournie

### Caractéristiques de l'antenne plate ANT794-3M

Numéro de référence	6NH9870-1AA00	
Réseaux de téléphonie mobile	<b>GSM 900</b>	<b>GSM 1800/1900</b>
Bandes passantes	890 - 960 MHz	1710 - 1990 MHz
Rapport d'ondes stationnaires (ROS)	≤ 2:1	≤ 1,5:1
Affaiblissement de régularité (Tx)	≈ 10 dB	≈ 14 dB
Gain d'antenne	0 dB	
Impédance	50 ohm	
Puissance max.	10 W	
Câble d'antenne	Câble HF RG 174 (raccordé) avec connecteur SMA	
Longueur de câble	1,2 m	
Degré de protection	IP 64	
Plage de température admissible	-40°C à +75°C	
Inflammabilité	UL 94 V2	
Enveloppe	ABS Polylac PA-765, gris clair (RAL 7035)	
Dimensions (P x L x H) en mm	70,5 x 146,5 x 20,5	
Poids	130 g	

## A.12.3 CM 1243-2 maître AS-i

### A.12.3.1 Données techniques pour l'interface AS-i master CM 1243-2

Tableau A- 216 Caractéristiques techniques de l'interface AS-i master CM 12432

<b>Caractéristiques techniques</b>	
Numéro de référence	3RK7243-2AA30-0XB0
<b>Interfaces</b>	
Consommation maximale de courant Du bus de fond de panier SIMATIC	Max. 250 mA, Tension d'alimentation du bus interne SIMATIC 5 V CC
Du câble AS-i	Max. 100 mA
Affectation des broches	Reportez-vous à la rubrique Connexions électriques de l'interface AS-i master CM 1243-2 (Page 872)
Diamètre du conducteur	0,2 mm <sup>2</sup> (AWG 24) ... 3,3 mm <sup>2</sup> (AWG 12)
Couple de serrage du connecteur ASI	0,56 Nm
<b>Conditions ambiantes admissibles</b>	
Température ambiante	
Pendant le stockage	-40 °C... 70 °C
Pendant le transport	-40 °C ... 70 °C
Durant le fonctionnement, avec une installation verticale (rail de montage standard horizontal)	0 °C ... 55 °C
Durant le fonctionnement, avec une installation horizontale (rail de montage standard vertical)	0 °C... 45 °C
Humidité relative à 25 °C durant le fonctionnement, aucune condensation, maximum	95 %
Degré de protection	IP20
<b>Alimentation, consommation de courant, dissipation de courant</b>	
Type d'alimentation	CC
Consommation de courant (typ.)	
Du bus interne du S7-1200	200 mA
Perte de courant (typ.)	2,4 W de AS-i
Du bus interne du S7-1200	0,5 W
<b>Dimensions et poids</b>	
Largeur	30 mm
Hauteur	100 mm
Profondeur	75 mm
Poids	
Poids net	122 g
Poids avec l'emballage	159 g

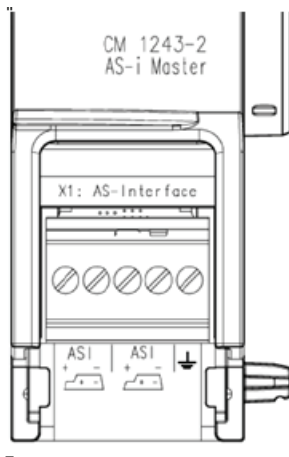
### A.12.3.2 Connexions électriques de l'interface AS-i master CM 1243-2

#### Alimentation du maître AS-i CM 1243-2

Le maître AS-i CM 1243-2 est alimenté par le bus de communication S7-1200. Ainsi, un message de diagnostic peut être envoyé au S7-2100 en cas de disparition de la tension AS-i. Le raccordement au bus de communication se trouve sur la droite du maître AS-i CM 1243-2.

#### Bornes de raccordement AS-Interface

Le bornier débrochable pour le raccordement du câble AS-i se trouve sous le cache-bornes inférieur à l'avant du maître AS-i CM 1243-2.



Si le raccordement est effectué à l'aide d'un câble profilé AS-i, la position de ce dernier (et donc la polarité) est indiquée par le symbole

Pour la dépose et le réenfichage du bornier débrochable, voir remarque du Manuel système "SIMATIC S7-1200 Système d'automatisation" (N° de référence : 6ES7298-8FA30-8AH0).

#### IMPORTANT


##### Charge maximale aux bornes

La charge maximale admissible aux bornes est de 8 A. Si cette valeur est dépassée sur le câble AS-i, le maître AS-i CM 1243-2 ne peut pas être monté en boucle sur le câble AS-i, mais doit être monté en dérivation (utiliser une seule des paires de contacts du maître AS-i CM 1243-2).

Vous trouverez de plus amples informations sur le raccordement du câble AS-i au chapitre "Montage, raccordement et mise en service des modules" du manuel "AS-i Master CM 1243-2 et module AS-i de découplage des données DCM 1271 pour SIMATIC S7-1200".



## Affectation des bornes

Marquage	Signification
ASI+	Connexion AS-i – pôle positif
ASI-	Connexion AS-i – pôle négatif
	Terre fonctionnelle

## A.12.4 RS232, RS422 et RS485

### A.12.4.1 Caractéristiques RS485 CB 1241

#### Remarque

Le firmware de votre CPU doit être de version V2.0 ou plus pour que vous puissiez utiliser ce CB.

Tableau A- 217 Caractéristiques générales

Caractéristiques techniques	CB 1241 RS485
Numéro de référence	6ES7 241-1CH30-1XB0
Dimensions L x H x P (mm)	38 x 62 x 21
Poids	40 grammes

Tableau A- 218 Emetteur et récepteur

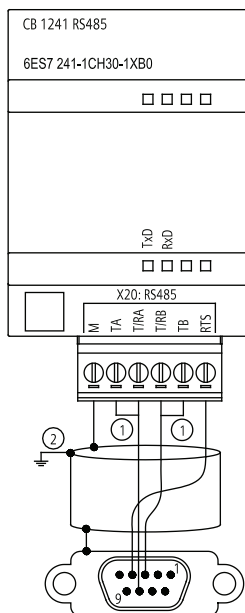
Caractéristiques techniques	CB 1241 RS485
Type	RS485 (semi-duplex 2 fils)
Plage de tension en mode commun	-7 V à +12 V, 1 seconde, 3 V eff. continu
Tension de sortie différentielle émetteur	2 V min. pour $R_L=100 \Omega$ 1,5 V min. pour $R_L=54 \Omega$
Terminaison et polarisation	10K vers +5 V sur B, RS485 Broche 3 10K vers GND sur A, RS485 Broche 4
Terminaison optionnelle	Court-circuiter broche TB sur broche T/RB, impédance de terminaison effective 127 $\Omega$ , connexion à RS485 broche 3 Court-circuiter broche TA sur broche T/RA, impédance de terminaison effective 127 $\Omega$ , connexion à RS485 broche 4
Impédance d'entrée récepteur	5,4 k $\Omega$ min., terminaison incluse
Seuil/sensibilité récepteur	+/- 0,2 V min., hystérésis typique 60 mV

Caractéristiques techniques	CB 1241 RS485
Isolation signal RS485 à la terre du châssis signal RS485 au commun logique CPU	500 V~, 1 minute
Longueur de câble, blindé	1000 m max.
Débit en bauds	300 bauds, 600 bauds, 1,2 kbits, 2,4 kbits, 4,8 kbits, 9,6 kbits (par défaut), 19,2 kbits, 38,4 kbits, 57,6 kbits, 76,8 kbits, 115,2 kbits
Parité	Sans parité (par défaut), parité paire, parité impaire, parité marque (bit de parité toujours à 1), parité espace (bit de parité toujours à 0)
Nombre de bits d'arrêt	1 (par défaut), 2
Contrôle du flux	Non pris en charge
Temps d'attente	0 à 65 535 ms

Tableau A- 219 Alimentation

Caractéristiques techniques	CB 1241 RS485
Perte de puissance (dissipation)	1,5 W
Consommation de courant (bus SM), max.	50 mA
Consommation de courant (24 V-) max.	80 mA

CB 1241 RS485 (6ES7 241-1CH30-1XB0)



- ① Connectez "TA" et "TB" comme indiqué pour terminer le réseau. La terminaison ne concerne que les appareils aux extrémités du réseau RS485.
- ② Utilisez un câble à paire torsadée blindée et connectez le blindage du câble à la terre.

La terminaison ne concerne que les deux extrémités du réseau RS485. Il n'y a pas de terminaison ni de polarisation des appareils situés entre les deux appareils d'extrémité. Reportez-vous au paragraphe "Polarisation et terminaison d'un connecteur de réseau RS485".

Tableau A- 220 Brochage pour le CB 1241 RS485 (6ES7 241-1CH30-1XB0)

Broche	Connecteur à 9 broches	X20
1	RS485 / Terre logique	--
2	RS485 / Non utilisé	--
3	RS485 / TxD+	3 - T/RB
4	RS485 / RTS	1 - RTS
5	RS485 / Terre logique	--
6	RS485 / Alim. 5V	--
7	RS485 / Non utilisé	--
8	RS485 / TxD-	4 - T/RA
9	RS485 / Non utilisé	--
Boîtier		7 - M

### Voir aussi

Polarisation et terminaison d'un connecteur de réseau RS485 (Page 600)

### A.12.4.2 Caractéristiques du module CM 1241 RS232

Tableau A- 221 Caractéristiques générales

Caractéristiques techniques	CM 1241 RS232
Numéro de référence	6ES7 241-1AH30-0XB0
Dimensions (mm)	30 x 100 x 75
Poids	150 grammes

Tableau A- 222 Emetteur et récepteur

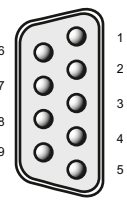
Caractéristiques techniques	CM 1241 RS232
Type	RS232 (duplex intégral)
Tension de sortie émetteur	+/- 5 V min. pour $R_L = 3 \text{ k}\Omega$
Tension de sortie émetteur	+/- 15 V- max.
Impédance d'entrée de récepteur	3 $\text{k}\Omega$ min.
Seuil/sensibilité récepteur	0,8 V minimum bas, 2,4 V maximum haut, 0,5 V hystérésis typique
Tension d'entrée récepteur	+/- 30 V- max.

Caractéristiques techniques	CM 1241 RS232
Isolation signal RS232 à la terre du châssis signal RS232 au commun logique CPU	500 V~, 1 minute
Longueur de câble, blindé	10 m max.
Débit	300 bauds, 600 bauds, 1,2 kbits, 2,4 kbits, 4,8 kbits, 9,6 kbits (par défaut), 19,2 kbits, 38,4 kbits, 57,6 kbits, 76,8 kbits, 115,2 kbits
Parité	Sans parité (par défaut), parité paire, parité impaire, parité marque (bit de parité toujours à 1), parité espace (bit de parité toujours à 0)
Nombre de bits d'arrêt	1 (par défaut), 2
Contrôle du flux	Matériel, logiciel
Temps d'attente	0 à 65535 ms

Tableau A- 223 Alimentation

Caractéristiques techniques	CM 1241 RS232
Perte de puissance (dissipation)	1,1 W
De +5 V-	220 mA

Tableau A- 224 Connecteur RS232 (mâle)

Broche	Description	Connecteur (mâle)	Broche	Description
1 DCD	Détection de porteuse : Entrée		6 DSR	Modem prêt : Entrée
2 RxD	Données reçues de DCE : Entrée		7 RTS	Demande pour émettre : Sortie
3 TxD	Données émises vers DCE : Sortie		8 CTS	Prêt à émettre : Entrée
4 DTR	Terminal de données prêt : Sortie		9 RI	Indicateur d'appel (inutilisé)
5 GND	Terre logique		Boîtier	Terre du châssis

### A.12.4.3 Caractéristiques du module CM 1241 RS422/485

#### Caractéristiques du module CM 1241 RS422/485

Tableau A- 225 Caractéristiques générales

<b>Caractéristiques techniques</b>	<b>CM 1241 RS422/485</b>
Numéro de référence	6ES7 241-1CH31-0XB0
Dimensions L x H x P (mm)	30 x 100 x 75
Poids	155 grammes

Tableau A- 226 Emetteur et récepteur

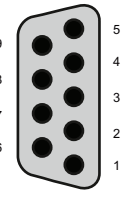
<b>Caractéristiques techniques</b>	<b>CM 1241 RS422/485</b>
Type	RS422 ou RS485, connecteur femelle subD 9 broches
Plage de tension en mode commun	-7 V à +12 V, 1 seconde, 3 V eff. continu
Tension de sortie différentielle émetteur	2 V min. pour $R_L=100 \Omega$ 1,5 V min. pour $R_L=54 \Omega$
Terminaison et polarisation	10 k $\Omega$ vers +5V sur B, PROFIBUS broche 3 10 k $\Omega$ vers GND sur A, PROFIBUS broche 8 Options de polarisation interne fournies ou pas de polarisation interne. Une terminaison externe est nécessaire dans tous les cas, voir Polarisation et terminaison d'un connecteur réseau RS485 (Page 600) et Configuration de RS422 et RS485 dans le Manuel système S7-1200 (Page 641).
Impédance d'entrée récepteur	5,4 k $\Omega$ min., terminaison incluse
Seuil/sensibilité récepteur	+/- 0,2 V min., hystérésis typique 60 mV
Isolation signal RS485 à la terre du châssis signal RS485 au commun logique CPU	500 V~, 1 minute
Longueur de câble, blindé	1000 m max. (en fonction du débit)
Débit en bauds	300 bauds, 600 bauds, 1,2 kbits, 2,4 kbits, 4,8 kbits, 9,6 kbits (par défaut), 19,2 kbits, 38,4 kbits, 57,6 kbits, 76,8 kbits, 115,2 kbits
Parité	Sans parité (par défaut), parité paire, parité impaire, parité marque (bit de parité toujours à 1), parité espace (bit de parité toujours à 0)
Nombre de bits d'arrêt	1 (par défaut), 2
Contrôle du flux	XON/XOFF pris en charge pour le mode RS422
Temps d'attente	0 à 65 535 ms

A.13 TeleService (TS Adapter et TS Adapter modulaire)

Tableau A- 227 Alimentation

Caractéristiques techniques	CM 1241 RS422/485
Perte de puissance (dissipation)	1,2 W
De +5 V-	240 mA

Tableau A- 228 Connecteur RS485 ou RS422 (femelle)

Broche	Description	Connecteur (femelle)	Broche	Description
1	Terre logique ou communication		6 PWR	+5V avec résistance série 100 ohms : sortie
2 TxD+ <sup>1</sup>	Connecté pour RS422 Non utilisé pour RS485 : sortie		7	Non connecté
3 TxD+	Signal B (RxD/TxD+) : entrée/sortie		8 TXD-	Signal A (RxD/TxD-) : entrée/sortie
4 RTS <sup>2</sup>	Demande pour émettre (niveau TTL) : sortie		9 TXD- <sup>1</sup>	Connecté pour RS422 Non utilisé pour RS485 : sortie
5 GND	Terre logique ou communication		Boîtier	Terre du châssis

<sup>1</sup> Les broches 2 et 9 sont utilisées uniquement comme signaux d'émission pour RS422.

<sup>2</sup> RTS est un signal de niveau TTL et peut être utilisé pour piloter un autre appareil semi-duplex en fonction de ce signal. Il est actif lorsque vous émettez et inactif le reste du temps.

### A.13 TeleService (TS Adapter et TS Adapter modulaire)

Les manuels suivants contiennent les caractéristiques techniques des adaptateurs TS Adapter IE Basic et TS Adapter modulaire :

- Industrial Software Engineering Tools  
TS Adapter modulaire
- Industrial Software Engineering Tools  
TS Adapter IE Basic

### A.14 Cartes mémoire SIMATIC

Numéro de référence	Capacité
6ES7 954-8LF01-0AA0	24 Mo
6ES7 954-8LE01-0AA0	12 Mo
6ES7 954-8LB01-0AA0	2 Mo

## A.15 Simulateurs d'entrées

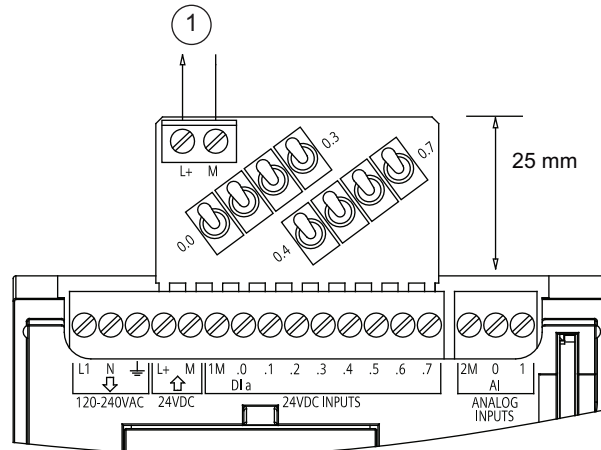
Tableau A- 229 Caractéristiques générales

Caractéristiques techniques	Simulateur 8 positions	Simulateur 14 positions
Numéro de référence	6ES7 274-1XF30-0XA0	6ES7 274-1XH30-0XA0
Dimensions L x H x P (mm)	43 x 35 x 23	67 x 35 x 23
Poids	20 grammes	30 grammes
Entrées	8	14
Utilisé avec la CPU	CPU 1211C, CPU 1212C	CPU 1214C

### ⚠ ATTENTION

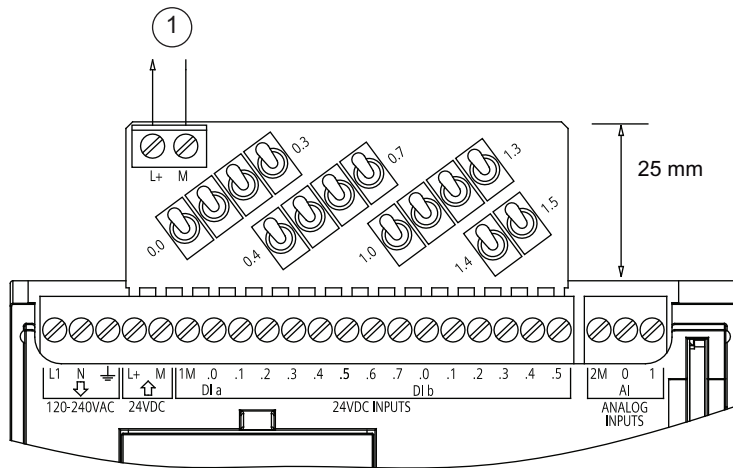
Ces simulateurs d'entrées ne sont pas homologués pour une utilisation dans des lieux dangereux de classe I DIV 2 ou de classe I zone 2. En effet, les commutateurs présentent un risque potentiel d'étincelle/d'explosion en cas d'utilisation dans un site de classe I DIV 2 ou de classe I zone 2.

Simulateur 8 positions (6ES7 274-1XF30-0XA0)



① Sortie alimentation capteur  
24 V-

Simulateur 14 positions (6ES7 274-1XF30-0XA0)



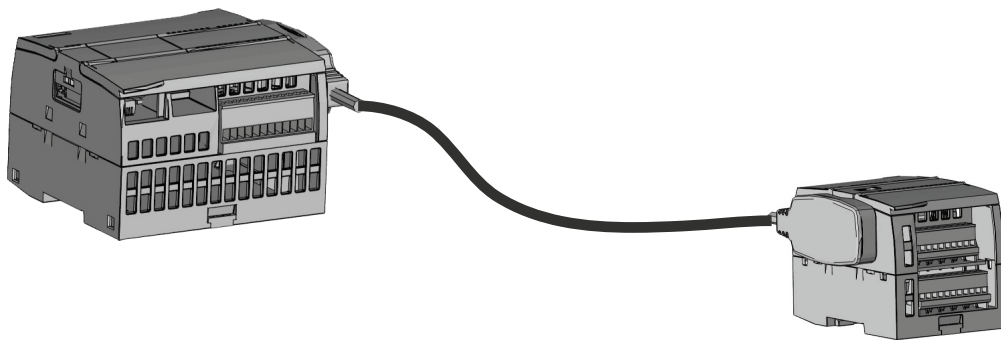
① Sortie alimentation capteur 24 V-

## A.16 Câble d'extension d'E/S

### Caractéristiques techniques

Numéro de référence	6ES7 290-6AA30-0XA0
Longueur de câble	2 m
Poids	200 g

Vous trouverez dans la partie sur l'installation (Page 59) des informations sur l'installation et la désinstallation du câble d'extension S7-1200.





## A.17 Produits connexes

### A.17.1 Module d'alimentation PM 1207

Le PM 1207 est un module d'alimentation pour le SIMATIC S7-1200. Il présente les caractéristiques suivantes :

- Entrée 120/230 V~, sortie 24 V-/2,5A
- Numéro de référence 6ESP 332-1SH71

Pour plus d'informations sur ce produit et pour obtenir la documentation du produit, reportez-vous au site Web du service client (<http://www.siemens.com/automation/>).

### A.17.2 Module commutateur compact CSM 1277

Le CSM1277 est un module commutateur compact Industrial Ethernet. Il peut être utilisé pour multiplier l'interface Ethernet du S7-1200 pour permettre des communications simultanées avec les pupitres opérateurs, les consoles de programmation et autres contrôleurs. Il présente les caractéristiques suivantes :

- 4 prises RJ45 pour connexion à l'Industrial Ethernet
- 3 prises polarisées dans le bornier pour connexion à l'alimentation externe 24 V- sur le dessus
- Des DEL pour l'affichage de diagnostic et d'état des ports Industrial Ethernet
- Numéro de référence 6GK7 277-1AA00-0AA0

Pour plus d'informations sur ce produit et pour obtenir la documentation du produit, reportez-vous au site Web du service client (<http://www.siemens.com/automation/>).



## Calcul d'un bilan de consommation

La CPU possède une alimentation interne fournissant du courant à la CPU elle-même, aux modules d'extension, ainsi qu'à d'autres équipements consommant du courant 24 V-.

Il existe quatre types de modules d'extension :

- Les modules d'entrées-sorties (SM) sont installés sur le côté droit de la CPU. Chaque CPU autorise un nombre maximum de modules d'entrées-sorties sans prise en compte du bilan de consommation.
  - Les CPU 1214C et 1215C autorisent 8 modules d'entrées-sorties.
  - La CPU 1212C autorise 2 modules d'entrées-sorties.
  - La CPU 1211C n'autorise pas de module d'entrées-sorties.
- Les modules de communication (CM) sont installés sur le côté gauche de la CPU. Trois modules de communication au maximum sont autorisés quelle que soit la CPU sans prise en compte du bilan de consommation.
- Les Signal Boards (SB), Communication Boards (CB) et Battery Boards (BB) sont installés sur le dessus de la CPU. Un Signal Board, un Communication Board ou un Battery Board au maximum est autorisé quelle que soit la CPU.

Les informations ci-après doivent vous aider à déterminer combien d'énergie ou de courant la CPU peut mettre à la disposition de votre configuration.

Chaque CPU fournit du courant 5 V- et 24 V- :

- La CPU fournit du courant 5V- pour les modules d'extension lorsqu'un tel module est connecté. Si les besoins en courant 5 V- des modules d'extension dépassent le courant fourni par la CPU, vous devez supprimer des modules d'extension jusqu'à ce que leurs besoins soient couverts.
- Chaque CPU a une alimentation de capteur 24 V- pouvant fournir du courant continu en 24 V aux entrées locales ou aux bobines de relais sur les modules d'extension. Si les besoins en courant 24 V CC dépassent le courant fourni par la CPU, vous pouvez ajouter une alimentation 24 V CC externe afin de fournir ce courant aux modules d'extension. Vous devez connecter à la main l'alimentation 24 V- aux entrées ou aux bobines de relais.

 **ATTENTION**

Connecter une alimentation 24 V- externe en parallèle avec l'alimentation de capteur en courant continu peut entraîner un conflit entre les deux alimentations, chacune cherchant à établir son propre niveau de tension de sortie préféré.

Ce conflit peut réduire la durée de vie ou provoquer une défaillance immédiate de l'une ou des deux alimentations, ayant pour effet un fonctionnement imprévisible du système d'automatisation pouvant entraîner la mort, des blessures graves et des dommages matériels importants.

L'alimentation de capteur CC sur la CPU et toute alimentation externe doivent fournir du courant à des points différents. Une seule connexion des conducteurs communs est autorisée.

Certains ports d'entrée d'alimentation 24 V dans le système d'automatisation sont interconnectés, avec un circuit commun logique connectant plusieurs bornes M. L'entrée d'alimentation 24 V de la CPU, l'entrée d'alimentation de bobine de relais des SM et une entrée d'alimentation analogique non isolée signalées comme non isolées dans les fiches techniques sont des exemples de circuits interconnectés. Toutes les bornes M non isolées doivent être connectées au même potentiel de référence externe.

 **ATTENTION**

Connecter des bornes M non isolées à des potentiels de référence différents provoque des flux de courant indésirables qui peuvent être à l'origine de dégâts ou d'un fonctionnement imprévisible dans l'automate et les équipements connectés

pouvant entraîner la mort, des blessures graves et des dommages matériels importants.

Vérifiez toujours que toutes les bornes M non isolées dans un système d'automatisation sont connectées au même potentiel de référence.

Les caractéristiques techniques (Page 747) fournissent des informations sur les courants fournis par les CPU et sur les besoins en courant des modules d'entrées-sorties.

#### Remarque

Si le bilan de consommation CPU est déficitaire, vous ne pourrez peut-être pas connecter le nombre maximal de modules autorisés pour votre CPU.

### Exemple de bilan de consommation

L'exemple suivant montre un exemple de calcul du bilan de consommation pour une configuration comprenant une CPU 1214C AC/DC/Relais, un SB 1223 2 entrées 24 VDC / 2 sorties 24 VDC, un CM 1241, trois SM 1223 8 entrées DC/8 sorties relais et un SM 1221 8 entrées DC. Ce système comporte 48 entrées et 36 sorties au total.

#### Remarque

La CPU a déjà alloué le courant nécessaire pour piloter les bobines de relais internes. Vous n'avez donc pas besoin d'inclure les besoins en courant des bobines de relais internes dans votre bilan de consommation.

Dans cet exemple, la CPU fournit suffisamment de courant continu 5 V- pour les SM, mais pas suffisamment de courant continu 24 V- à partir de l'alimentation de capteur pour toutes les entrées et bobines de relais d'extension. Les E/S requièrent 456 mA alors que la CPU ne fournit que 400 mA. Il faut donc une source supplémentaire d'au moins 56 mA à 24 V- pour opérer toutes les entrées et sorties 24 V- présentes.

Tableau B- 1 Exemple de bilan de consommation

Courant fourni par la CPU	5 V-	24 V-
CPU 1214C AC/DC/Relais	1600 mA	400 mA
<i>moins</i>		
Besoins du système	5 V-	24 V-
CPU 1214C, 14 entrées	-	14 * 4 mA = 56 mA
1 SB 1223 2 entrées 24 VDC / 2 sorties 24 VDC	50 mA	2 * 4 mA = 8 mA
1 CM 1241 RS422/485, courant 5 V	220 mA	
3 SM 1223, courant 5 V	3 * 145 mA = 435 mA	-
1 SM 1221, courant 5 V	1 * 105 mA = 105 mA	-
3 SM 1223, 8 entrées chacun	-	3 * 8 * 4 mA = 96 mA
3 SM 1223, 8 bobines de relais chacun	-	3 * 8 * 11 mA = 264 mA
1 SM 1221, 8 entrées	-	8 * 4 mA = 32 mA
<b>Total des besoins</b>	810 mA	456 mA
<i>égale</i>		
Bilan de consommation	5 V-	24 V-
Excédent/déficit de courant	790 mA	(56 mA)

**Formulaire pour calculer votre bilan de consommation**

Servez-vous du tableau suivant pour déterminer combien de courant la CPU S7-1200 peut mettre à la disposition de votre configuration. Vous trouverez dans les caractéristiques techniques (Page 747) des informations sur le courant fourni par votre modèle de CPU et sur les besoins en courant de vos modules d'entrées-sorties.

Tableau B- 2 Calcul d'un bilan de consommation

<b>Courant fourni par la CPU</b>	<b>5 V-</b>	<b>24 V-</b>
<i>moins</i>		
<b>Besoins du système</b>	<b>5 V-</b>	<b>24 V-</b>
<b>Total des besoins</b>		
<i>égale</i>		
<b>Bilan de consommation</b>	<b>5 V-</b>	<b>24 V-</b>
Excédent/déficit de courant		

# Numéros de référence



## C.1 Modules CPU

Tableau C- 1 CPU S7-1200

Modèles de CPU		Numéro de référence
CPU 1211C	CPU 1211C DC/DC/DC	6ES7 211-1AE31-0XB0
	CPU 1211C AC/DC/Relais	6ES7 211-1BE31-0XB0
	CPU 1211C DC/DC/Relais	6ES7 211-1HE31-0XB0
CPU 1212C	CPU 1212C DC/DC/DC	6ES7 212-1AE31-0XB0
	CPU 1212C AC/DC/Relais	6ES7 212-1BE31-0XB0
	CPU 1212C DC/DC/Relais	6ES7 212-1HE31-0XB0
CPU 1214C	CPU 1214C DC/DC/DC	6ES7 214-1AG31-0XB0
	CPU 1214C AC/DC/Relais	6ES7 214-1BG31-0XB0
	CPU 1214C DC/DC/Relais	6ES7 214-1HG31-0XB0
CPU 1215C	CPU 1215C DC/DC/DC	6ES7 215-1AG31-0XB0
	CPU 1215C AC/DC/Relais	6ES7 215-1BG31-0XB0
	CPU 1215C DC/DC/Relais	6ES7 215-1HG31-0XB0

## C.2 Modules d'entrées-sorties (SM), Signal Boards (SB) et Battery Boards (BB)

Tableau C- 2 Modules d'entrées-sorties (SM)

Modules d'entrées-sorties		Numéro de référence
Entrées TOR	SM 1221 8 entrées 24 VDC (P/N)	6ES7 221-1BF30-0XB0
	SM 1221 16 entrées 24 VDC (P/N)	6ES7 221-1BH30-0XB0
Sorties TOR	SM 1222 8 sorties 24 VDC (N)	6ES7 222-1BF30-0XB0
	SM 1222 16 sorties 24 VDC (N)	6ES7 222-1BH30-0XB0
	SM 1222 8 sorties relais	6ES7 222-1HF30-0XB0
	SM 1222 8 sorties relais (inverseur)	6ES7 222-1XF30-0XB0
	SM 1222 16 sorties relais	6ES7 222-1HH30-0XB0
Entrées/sorties TOR	SM 1223 8 entrées 24 VDC (P/N) / 8 sorties 24 VDC (N)	6ES7 223-1BH30-0XB0
	SM 1223 16 entrées 24 VDC (P/N) / 16 sorties 24 VDC (N)	6ES7 223-1BL30-0XB0
	SM 1223 8 entrées 24 VDC (P/N) / 8 sorties relais	6ES7 223-1PH30-0XB0
	SM 1223 16 entrées 24 VDC (P/N) / 16 sorties relais	6ES7 223-1PL30-0XB0
	SM 1223 8 entrées 120/230 VAC (P/N) / 8 sorties relais	6ES7 223-1QH30-0XB0
Entrées	SM 1231 4 entrées analogiques	6ES7 231-4HD30-0XB0

Modules d'entrées-sorties		Numéro de référence
analogiques	SM 1231 8 entrées analogiques	6ES7 231-4HF30-0XB0
	SM 1231 4 entrées analogiques x 16 bits (High Feature)	6ES7 231-5ND30-0XB0
Sorties analogiques	SM 1232 2 sorties analogiques	6ES7 232-4HB30-0XB0
	SM 1232 4 sorties analogiques	6ES7 232-4HD30-0XB0
Entrées/sorties analogiques	SM 1234 4 entrées analogiques / 2 sorties analogiques	6ES7 234-4HE30-0XB0
RTD et Thermocouple	SM 1231 TC 4 x 16 bits	6ES7 231-5QD30-0XB0
	SM 1231 TC 8 x 16 bits	6ES7 231-5PD30-0XB0
	SM 1231 RTD 4 x 16 bits	6ES7 231-5PD30-0XB0
	SM 1231 RTD 8 x 16 bits	6ES7 231-5PD30-0XB0

Tableau C- 3 Signal Boards (SB) et Battery Boards (BB)

Signal Boards et Battery Boards		Numéro de référence
Entrées TOR	SB 1221 4 entrées 24 VDC 200 kHz (N)	6ES7 221-3BD30-0XB0
	SB 1221 4 entrées 5 VDC 200 kHz (N)	6ES7 221-3AD30-0XB0
Sorties TOR	SB 1222 4 sorties 24 VDC 200 kHz (P/N)	6ES7 222-1BD30-0XB0
	SB 1222 4 sorties 5 VDC 200 kHz (P/N)	6ES7 222-1AD30-0XB0
Entrées/sorties TOR	SB 1223 2 entrées 24 VDC (P) / 2 sorties 24 VDC (N)	6ES7 223-0BD30-0XB0
	SB 1223 2 entrées 24 VDC (N) / 2 sorties 24 VDC (P/N) 200 kHz	6ES7 223-3BD30-0XB0
	SB 1223 2 entrées 5 VDC (N) / 2 sorties 5 VDC (P/N) 200 kHz	6ES7 223-3AD30-0XB0
Analogiques	SB 1232 1 sortie analogique	6ES7 232-4HA30-0XB0
	SB 1231 1 entrée analogique	6ES7 231-4HA30-0XB0
	SB 1231 1 entrée analogique Thermocouple	6ES7 231-5QA30-0XB0
	SB 1231 1 entrée analogique RTD	6ES7 231-5PA30-0XB0
Pile	Battery Board BB 1297	6ES7 297-0AX30-0XA0

## C.3 Communication

Tableau C- 4 Module de communication (CM)

Module de communication (CM)			Numéro de référence
RS232, RS422 et RS485	CM 1241 RS232	RS232	6ES7 241-1AH30-0XB0
	CM 1241 RS422/485	RS422/485	6ES7 241-1CH31-0XB0
PROFIBUS	CM 1243-5	Maître PROFIBUS	6GK7 243-5DX30-0XE0
	CM 1242-5	Esclave PROFIBUS	6GK7 242-5DX30-0XE0
Maître interface AC	CM 1243-2	Maître interface AC	3RK7 243-2AA30-0XB0



Tableau C- 5 Communication Board (CB)

Communication Board (CB)			Numéro de référence
RS485	CB 1241 RS485	RS485	6ES7 241-1CH30-1XB0

Tableau C- 6 Processeur de communication (CP)

Processeur de communication (CP)		Numéro de référence
CP 1242-7	GPRS	6GK7 242-7KX30-0XE0

Tableau C- 7 TeleService

Adaptateur TS Adapter	Numéro de référence
Adaptateur TS Adapter IE Basic	6ES7 972-0EB00-0XA0
Module TS GSM	6GK7 972-0MG00-0XA0
Module TS RS232	6ES7 792-0MS00-0XA0
Module TS Modem	6ES7 972-0MM00-0XA0
Module TS RNIS	6ES7 972-0MD00-0XA0

Tableau C- 8 Accessoires

Accessoire			Numéro de référence
Antenne	ANT794-4MR	Antenne GSM/GPRS	6NH9 860-1AA00
	ANT794-3M	Antenne en nappe	6NH9 870-1AA00

Tableau C- 9 Connecteurs

Type de connecteur		Numéro de référence
RS485	Sortie de câble 35 degrés, connexion par bornes à vis	6ES7 972-0BA42-0XA0
	Sortie de câble 35 degrés, connexion autodénudante	6ES7 972-0BA60-0XA0

## C.4 Autres modules

Tableau C- 10 Produits connexes

Élément		Numéro de référence
Module d'alimentation	Alimentation PM 1207	6EP1 332-1SH71
Commutateur Ethernet	Commutateur Ethernet CSM 1277, 4 ports	6GK7 277-1AA10-0AA0

## C.5 Cartes mémoire

Tableau C- 11 Cartes mémoire

Cartes mémoire SIMATIC	Numéro de référence
SIMATIC MC 2 Mo	6ES7 954-8LB01-0AA0
SIMATIC MC 12 Mo	6ES7 954-8LE01-0AA0
SIMATIC MC 24 Mo	6ES7 954-8LF01-0AA0

## C.6 Appareils IHM Basic

Tableau C- 12 Appareils IHM

Pupitres HMI Basic Panels	Numéro de référence
KTP400 Basic (monochrome, PN)	6AV6 647-0AA11-3AX0
KTP600 Basic (monochrome, PN)	6AV6 647-0AB11-3AX0
KTP600 Basic (couleur, PN)	6AV6 647-0AD11-3AX0
KTP1000 Basic (couleur, PN)	6AV6 647-0AF11-3AX0
TP1500 Basic (couleur, PN)	6AV6 647-0AG11-3AX0

## C.7 Pièces détachées et autres matériels

Tableau C- 13 Câbles d'extension, simulateurs et borniers de connexion

Elément	Numéro de référence		
Câble d'extension d'E/S	Câble d'extension d'E/S, 2 m	6ES7 290-6AA30-0XA0	
Simulateur d'E/S	Simulateur (1214C/1211C - 8 positions)	6ES7 274-1XF30-0XA0	
	Simulateur (1214C - 14 positions)	6ES7 274-1XH30-0XA0	
Kit volets d'accès	CPU 1211C/1212C	6ES7 291-1AA30-0XA0	
	CPU 1214C	6ES7 291-1AB30-0XA0	
	CPU 1215C	6ES7 291-1AC30-0XA0	
	Module d'entrées-sorties, 45 mm	6ES7 291-1BA30-0XA0	
	Module d'entrées-sorties, 70 mm	6ES7 291-1BB30-0XA0	
	Module de communication	6ES7 291-1CC30-0XA0	
Bornier de connexion	Etain	7 bornes, 4/boîte	6ES7 292-1AG30-0XA0
		8 bornes, 4/boîte	6ES7 292-1AH30-0XA0
		11 bornes, 4/boîte	6ES7 292-1AL30-0XA0
		12 bornes, 4/boîte	6ES7 292-1AM30-0XA0
		14 bornes, 4/boîte	6ES7 292-1AP30-0XA0
		20 bornes, 4/boîte	6ES7 292-1AV30-0XA0

Élément		Numéro de référence	
	Or	3 bornes, 4/boîte (pour CPU analogique)	6ES7 292-1BC30-0XA0
		6 bornes, 4/boîte (pour Signal Board)	6ES7 292-1BF30-0XA0
		6 bornes, 4/boîte (pour CPU analogique)	6ES7 292-1BF30-0XB0
		7 bornes, 4/boîte (pour module d'entrées-sorties analogiques)	6ES7 292-1BG30-0XA0
		11 bornes, 4/boîte (pour module d'entrées-sorties analogiques)	6ES7 292-1BL30-0XA0

## C.8 Logiciel de programmation

Tableau C- 14 Logiciel de programmation

Logiciel SIMATIC		Numéro de référence
Logiciel de programmation	STEP 7 Basic V11	6ES7 822-0AA01-0YA0
	STEP 7 Professional V11	6ES7 822-1AA01-0YA5
Logiciel de visualisation	WinCC Basic V11	6AV2100-0AA01-0AA0
	WinCC Comfort V11	6AV2101-0AA01-0AA5
	WinCC Advanced V11	6AV2102-0AA01-0AA5
	WinCC Professional 512 PowerTags V11	6AV2103-0DA01-0AA5
	WinCC Professional 4096 PowerTags V11	6AV2103-0HA01-0AA5
	WinCC Professional max. PowerTags V11	6AV2103-0XA01-0AA5

## C.9 Documentation

Tableau C- 15 Documentation S7-1200

Documentation imprimée	Langue	Numéro de référence
Automate programmable S7-1200, Manuel système	Allemand	6ES7 298-8FA30-8AH0
	Anglais	6ES7 298-8FA30-8BH0
	Français	6ES7 298-8FA30-8CH0
	Espagnol	6ES7 298-8FA30-8DH0
	Italien	6ES7 298-8FA30-8EH0
	Chinois	6ES7 298-8FA30-8KH0
S7-1200 Easy Book	Allemand	6ES7 298-8FA30-8AQ0
	Anglais	6ES7 298-8FA30-8BQ0
	Français	6ES7 298-8FA30-8CQ0
	Espagnol	6ES7 298-8FA30-8DQ0
	Italien	6ES7 298-8FA30-8EQ0
	Chinois	6ES7 298-8FA30-8KQ0



# Index

## A

ABS (Valeur absolue), 215  
Accès  
    journaux à partir du PC, 548  
    pages Web personnalisées, 573  
ACOS (Arc cosinus), 218  
Acquisition des valeurs dans un DB en ligne, 732  
Actualisation de pages Web personnalisées, 555  
Actualisation des pages Web personnalisées, 555  
Adaptateur TS Adapter, 22  
    carte SIM, 62  
    installation d'un module TS, 61  
    installation sur un profilé support, 64  
    montage mural, 64  
ADD (Addition), 212  
Adressage  
    entrées (I) ou sorties (Q) individuelles, 95  
    mémoire image, 94  
    Valeurs booléennes ou de bit, 95  
    zones de mémoire, 94  
Adresse AS-i, 514  
    configuration, 514  
Adresse IP, 146, 147  
    adresse MAC, 145  
    Affectation, 141, 149  
    affectation en ligne, 144  
    configuration, 145  
    configuration de la CPU en ligne, 726  
    configuration des appareils, 131  
Adresse IP de routeur, 147  
adresse MAC, 145, 150  
Adresse PROFIBUS, 509, 510  
    configuration, 509  
Adresses de mémoire, 94, 96  
Affectation de types Enum, pages Web personnalisées, 564  
Affichage des adresses MAC et IP, 150  
Ajouter des entrées ou sorties aux opérations CONT ou LOG, 35  
Ajouter un appareil  
    CPU, 128  
    CPU non spécifiée, 129  
    détection du matériel existant, 129  
Alarme, 316  
    accès à la périphérie, 501

Alarme temporisée, 314  
Alarmes  
    ATTACH et DETACH, 307  
    CAN\_DINT (Annuler une alarme temporisée), 314  
    présentation, 78  
    SRT\_DINT (Démarrer une alarme temporisée), 314  
    temps d'attente, 79  
Alarmes d'accès à la périphérie, 501  
Alias dans les pages Web personnalisées, 562  
AND (ET), 255  
API  
    affectation d'une adresse IP à une CPU en ligne, 144  
    ajouter des modules, 130  
    bilan de consommation, 45  
    bornier de connexion, 58  
    câble d'extension, 59  
    carte mémoire, 115, 878  
    charge due à la communication, 87  
    chargement dans la CPU, 179  
    comparaison et synchronisation, 729  
    compteur d'heures de fonctionnement, 267  
    Conception de système, 155  
    configuration des appareils, 127  
    Configuration HSC, 369  
    copie des blocs depuis une CPU en ligne, 179  
    CPU 1211C, 754  
    CPU 1212C, 764  
    CPU 1214C, 774  
    CPU 1215C, 784  
    états de fonctionnement, 73  
    forçage permanent, 736, 737  
    installation, 49, 50  
    présentation de la CPU, 19  
    propriété Synchronisation de l'heure, 152  
    protection du savoir-faire, 176  
    RD\_LOC\_T (Lire l'heure locale), 265  
    RD\_SYS\_T (Lire l'heure système), 265  
    table de visualisation, 733  
    temps de cycle, 87  
    temps de cycle, 87  
    traitement de mise en route, 76  
    Utilisation de blocs, 156  
    variables, 94  
    visualisation, 731  
    WR\_SYS\_T (Ecrire l'heure système), 265  
Appareil  
    PROFINET IO, 497

- Appareils IHM
  - configuration de la communication PROFINET, 491
  - liaison réseau, 135
  - présentation, 27
- Appel de blocs de code dans le programme utilisateur, 158
- Architecture d'interrogation, 636
- Architecture d'interrogation esclave, 636
- Architecture d'interrogation maître, 636
- AS-i
  - adresse AS-i, 514
  - affectation système, 515
  - affectation système d'adresses d'esclaves, 515
  - ajout d'un esclave AS-i, 512
  - ajout d'un module maître AS-i CM 1243-2, 512
  - configuration d'esclaves avec STEP 7, 516
  - configuration d'esclaves sans STEP 7, 515
  - instructions de périphérie décentralisée, 291
  - liaison réseau, 513
  - propriétés de l'adresse AS-i, 514
  - RDREC, 292
  - transfert de valeurs analogiques, 516
  - transfert de valeurs TOR, 516
  - WRREC, 292
- ASIN (Arc sinus), 218
- Assistance, 3
- Assistance client, 3
- Assistance technique, 3
- Assistance technique Siemens, 3
- Assistant d'importation du certificat, 596
- ATH (ASCII en hexadécimal), 282
- ATTACH, 307
- AWP\_Enum\_Def, 563
- AWP\_Enum\_Ref, 564
- AWP\_Import\_Fragment, 566
- AWP\_In\_Variable, 557, 561
- AWP\_Out\_Variable, 559
- AWP\_Start\_Fragment, 565
  
- B**
- Barre d'outils Favoris, 32
- Basic panels (HMI), 27
- Besoins du système, 30
- Besoins en courant
  - bilan de consommation, 883
  - calcul d'un bilan de consommation, 885
  - formulaire pour le calcul, 886
- Bibliothèque du protocole USS
  - codes d'état, 658
  - conditions requises pour l'utilisation, 648
  - informations générales sur la configuration d'un entraînement, 660
  - présentation, 646
  - USS\_DRV, 651
  - USS\_PORT, 654
  - USS\_RPM, 655
  - USS\_WPM, 656
- Bibliothèque globale
  - présentation du protocole USS, 646
- Bilan de consommation, 45
  - exemple, 885
  - formulaire pour le calcul, 886
  - présentation, 883
- Bits d'arrêt, 623
- Bloc
  - contrôle de cohérence, 183
- Bloc de code
  - alarmes, 21, 756, 766, 776, 786
  - appel de blocs de code dans le programme utilisateur, 158
  - appels de blocs, 71
  - bloc de données d'instance (DB), 161
  - bloc d'organisation (OB), 159
  - blocs d'organisation (OB), 21, 756, 766, 776, 786
  - compteurs (exigences en termes de quantité et de mémoire), 21, 756, 766, 776, 786
  - DB (bloc de données), 71, 162
  - FB (bloc fonctionnel), 71, 161
  - FC (fonction), 71, 160
  - liaison à une CPU ou carte mémoire, 177
  - nombre de blocs de code, 21, 756, 766, 775, 786
  - nombre d'OB, 21, 756, 766, 776, 786
  - numéros de FC, FB et DB valides, 71
  - profondeur d'imbrication, 21, 756, 766, 776, 786
  - programmes linéaires et structurés, 156
  - protection contre la copie, 177
  - protection du savoir-faire, 176
  - taille du programme utilisateur, 21, 756, 766, 775, 786
  - temporisations (exigences en termes de quantité et de mémoire), 21, 756, 766, 776, 786
  - types de blocs de code, 71
  - valeur initiale d'un FB, 161
  - visualisation, 21, 756, 766, 776, 786
- Bloc de données
  - acquisition des valeurs, 732
  - bloc de données d'instance, 94
  - bloc de données global, 94, 162
  - bloc d'organisation (OB), 159
  - CONF\_DATA, 483
  - FB unique avec plusieurs DB d'instance, 162
  - numéros de DB valides, 71

- présentation, 71, 162
  - redéfinition des valeurs initiales, 732
  - Bloc de données d'instance, 94
  - Bloc de données global, 94, 162
  - Bloc de gestion de données (DHB), 162
  - Bloc de transmission (bloc T), 495
  - Bloc d'organisation
    - appel, 78
    - appel de blocs de code dans le programme utilisateur, 158
    - classes de priorité, 78
    - Configuration du fonctionnement, 160
    - création, 160
    - fonction, 78
    - plusieurs OB de cycle de programme, 160
    - présentation, 71
    - programmation linéaire et structurée, 156
    - protection du savoir-faire, 176
    - traitement, 159
    - traitement de mise en route, 76
  - Bloc fonctionnel (FB)
    - appel de blocs de code dans le programme utilisateur, 158
    - bloc de données d'instance, 161
    - FB unique avec plusieurs DB d'instance, 162
    - numéros de FB valides, 71
    - paramètres de sortie, 161
    - présentation, 71, 161
    - programmes linéaires et structurés, 156
    - protection du savoir-faire, 176
    - valeur initiale, 161
  - Blocs
    - alarmes, 21, 79, 756, 766, 776, 786
    - appel d'un FB ou d'une FC avec SCL, 168
    - appels de blocs, 71
    - bloc de données (DB), 71
    - bloc de données d'instance (DB), 161
    - bloc fonctionnel (FB), 71, 161
    - blocs d'organisation (OB), 21, 71, 78, 79, 756, 766, 776, 786
    - chargement dans la CPU, 179
    - compteurs (exigences en termes de quantité et de mémoire), 21, 756, 766, 776, 786
    - copie des blocs depuis une CPU en ligne, 179
    - DB mono-instance ou multi-instance, 161
    - événements, 79
    - fonction (FC), 71, 160
    - nombre de blocs de code, 21, 756, 766, 775, 786
    - nombre d'OB, 21, 79, 756, 766, 776, 786
    - numéros de FC, FB et DB valides, 71
    - OB de démarrage, 79
    - profondeur d'imbrication, 21, 71, 756, 766, 776, 786
    - programmes linéaires et structurés, 156
    - protection par mot de passe, 176
    - taille du programme utilisateur, 21, 71, 756, 766, 775, 786
    - temporisations (exigences en termes de quantité et de mémoire), 21, 756, 766, 776, 786
    - types de blocs de code, 71
    - valeur initiale d'un FB, 161
    - visualisation, 21, 756, 766, 776, 786
  - Blocs de code, 158
  - Blocs de données pour les pages Web personnalisées
    - importation de fragments, 566
  - Bornes
    - charge max. admissible, 872
  - Bornier de connexion, 58
  - Boutons MARCHE/ARRET, 38
  - BUFFER, paramètre de SEND\_PTP, 615
- C**
- Câble d'extension
    - démontage, 59
    - installation, 59
  - CALCULATE, 211
  - CALCULER, 33
    - mise à l'échelle de valeurs analogiques, 34
  - Calendrier, 263
  - CAN\_DINT (Annuler une alarme temporisée), 314
  - Caractère de début de message, 628
  - Caractère de fin de message, 632
  - Caractères spéciaux
    - pages Web personnalisées, 567
  - Caractéristiques
    - caractéristiques techniques d'ordre général, 747
    - cartes mémoire, 878
    - CB 1241 RS485, 874
    - CM 1241 RS232, 876
    - Compatibilité électromagnétique (CEM), 750
    - Conditions ambiantes, 751
    - CPU 1211C, 754
    - CPU 1212C, 764
    - CPU 1214C, 774
    - CPU 1215C, 784
    - Durée d'utilisation électrique des relais, 754
    - environnements industriels, 750
    - homologation ATEX, 749
    - Homologation CE, 747
    - homologation C-Tick, 749
    - homologation cULus, 748
    - homologation FM, 748
    - homologation pour le domaine maritime, 750

- module d'entrées-sorties SM 1231 AI 4 x 16 bits TC, 824
- module d'entrées-sorties SM 1231 AI 8 x 16 bits TC, 824
- module SM 1231 AI 4 x RTD x 16 bits, 830
- module SM 1231 AI 8 x RTD x 16 bits, 830
- Protection, 752
- représentation des entrées analogiques (courant), 822, 852
- représentation des entrées analogiques (tension), 822, 852
- représentation des sorties analogiques (courant), 823, 853
- représentation des sorties analogiques (tension), 823, 853
- SB 1221 DI 4, 200 kHz, 836
- SB 1222 DQ 4, 200 kHz, 838
- SB 1223 DI 2 / DQ 2, 844
- SB 1223 DI 2 / DQ 2, 200 kHz, 841
- SB 1231 AI 1 x 12 bits, 847
- SB 1231 AI 1 x 16 bits RTD, 858
- SB 1231 AI 1 x 16 bits Thermocouple, 854
- SB 1232 AQ 1x12 bits, 849
- schéma de câblage du SB 1231 AI 1 x 16 bits RTD, 860
- schéma de câblage du SB 1231 AI 1 x 16 bits Thermocouple, 857
- schéma de câblage du SB 1231 AI x 12 bits, 848
- schéma de câblage du SM 1231 AI 4 x 16 bits TC, 826
- schéma de câblage du SM 1231 AI 8 x 16 bits TC, 826
- schéma de câblage du SM 1231 RTD 4 x 16 bits, 832
- schéma de câblage du SM 1231 RTD 8 x 16 bits, 832
- schémas de câblage des SM 1231 d'entrées analogiques, 816
- schémas de câblage des SM 1232 de sorties analogiques, 817
- schémas de câblage du SM 1234 d'entrées/sorties analogiques, 820
- simulateurs d'entrées, 880
- SM 1221, module d'entrées-sorties, 796
- SM 1221, schéma de câblage, 797
- SM 1222, 799, 801
- SM 1222 DQ8 RLY inverseur, 799
- SM 1222, schéma de câblage, 801
- SM 1223, module d'entrées-sorties, 806, 809
- SM 1223, schéma de câblage, 806, 809
- SM 1231 AI 4 x 13 bits, 812
- SM 1232 AQ 2 x 14 bits, 816
- SM 1232 AQ 4 x 14 bits, 816
- SM 1234 AI 4 x 13 bits / AQ 2 x 14 bits, 818
- temps de réponse indicelle (CPU), 760, 770, 780, 790
- temps de réponse indicelle (SB), 851
- temps de réponse indicelle (SM), 821
- tensions nominales, 752
- Caractéristiques CB 1241 RS485, 874
- Caractéristiques des Signal Boards (SB) TOR
  - SB 1221 DI 4, 200 kHz, 836
  - SB 1222 DQ 4, 200 kHz, 838
  - SB 1223 DI 2 / DQ 2, 844
  - SB 1223 DI 2 / DQ 2, 200 kHz, 841
- Caractéristiques techniques, 747
- Caractéristiques techniques d'ordre général, 747
- Carte mémoire
  - caractéristiques, 878
  - carte programme, 121
- Carte transfert, 118
  - carte transfert vide en cas d'oubli du mot de passe, 126
  - configuration des paramètres de mise en route, 118
  - fonctionnement, 115
  - insertion dans la CPU, 116
  - mise à jour du firmware, 123
  - numéro de référence, 878
  - oubli du mot de passe, 126
  - présentation, 115
- Carte programme
  - configuration des paramètres de mise en route, 118
  - création, 121
  - fonctionnement, 115
  - insertion dans la CPU, 116
  - numéro de référence, 878
  - présentation, 115
- Carte transfert, 118
  - carte transfert vide en cas d'oubli du mot de passe, 126
  - configuration des paramètres de mise en route, 118
  - fonctionnement, 115
  - insertion dans la CPU, 116
  - numéro de référence, 878
  - oubli du mot de passe, 126
  - présentation, 115
- CEIL (Arrondir à l'entier supérieur), 232
- Certificat de sécurité Siemens, pages Web, 539, 596
- Chaîne
  - présentation des instructions sur chaîne, 284
  - S\_MOVE (Copier chaîne), 271
- Chaîne de caractères
  - STRING, type de données, 106



- Changement de langue pour les pages Web personnalisées, 586
- Char (type de données caractère), 105
- Chargement à partir de l'appareil
  - copie des blocs depuis une CPU en ligne, 179
  - programme utilisateur, 179
- Chargement dans la CPU
  - affichage des adresses MAC et IP, 150
  - DB de pages Web personnalisées, 573
  - programme utilisateur, 179
  - projet, 179
- Charges de lampe, 68
- Charges inductives, 69
- Classe de priorité, 78
- Classe de protection, 752
- Classes d'erreurs point à point, 605
- CM 1241
  - caractéristiques RS232, 876
  - caractéristiques RS422/RS485, 877
- Codes d'erreur
  - erreurs communes pour les instructions avancées, 358
  - RALRM, 298
  - RDREC, 298
  - WRREC, 298
- Colonnes et en-têtes dans les Task Cards, 36
- Commande de DB de fragment manuels, 590
- Commande de mouvement
  - configuration de l'axe, 395
  - fins de course matériels et logiciels, 426
  - MC\_ChangeDynamic, 422
  - MC\_CommandTable, 419
  - MC\_Halt, 408
  - MC\_Home, 405
  - MC\_MoveAbsolute, 410
  - MC\_MoveJog, 417
  - MC\_MoveRelative, 412
  - MC\_MoveVelocity, 414
  - MC\_Power, 401
  - MC\_Reset, 404
  - paramètres de configuration du référencement, 431
  - présentation, 391
  - référencement (séquence pour le référencement actif), 433
  - référencement de l'axe, 429
- Commandes AWP, 555
  - combinaison de définitions, 567
  - définition d'un type Enum, 563
  - écriture de variables, 557
  - écriture de variables spéciales, 561
  - génération de fragments, 565
  - importation de fragments, 566
  - lecture de variables spéciales, 559
  - référencement d'un type Enum, 564
  - utilisation d'un alias, 562
- Communication
  - active/passive, 136, 138, 525
  - adresse AS-i, 514
  - adresse IP, 145
  - adresse MAC, 145
  - adresse PROFIBUS, 509
  - Architecture d'interrogation, 636
  - charge due à la communication, 87
  - configuration, 136, 138, 525
  - Contrôle du flux, 624
  - ID de liaison, 455
  - Liaison matérielle, 489
  - liaison réseau, 135
  - nombre de liaisons (PROFINET), 452
  - paramètres, 138, 545
  - paramètres d'émission et de réception, 625
  - PROFINET et PROFIBUS, 451
  - propriété Synchronisation de l'heure (PROFINET), 152
  - Réseau, 489
  - statistiques, 545
  - TCON\_Param, 138
  - temps de cycle, 87
- Communication active/passive
  - configuration des partenaires, 136, 525
  - ID de liaison, 455
  - paramètres, 138
- Communication board (CB)
  - ajouter des modules, 130
  - RS485, 599
- Communication Board (CB)
  - CB 1241 RS485, 874
  - configuration des appareils, 127
  - configuration des paramètres, 133
  - DEL de signalisation, 599, 721
  - démontage, 53
  - installation, 53
  - présentation, 25
  - programmation, 635
  - tableau comparatif, 22
- Communication CPU, 454
- Communication passive/active
  - configuration des partenaires, 136, 525
  - ID de liaison, 455
  - paramètres, 138
- Communication point à point, 602
  - configuration de l'exemple de programme, 638
  - configuration des paramètres, 625
  - configuration des ports, 622

- émulateur de terminal pour l'exemple de programme, 645
- exemple de programme, 637
- exemple de programme, exécution, 646
- exemple de programme, programmation STEP 7, 644
- programmation, 635
- Communication programmable, protocole, 602
- Communication réseau, 489
- Communication S7
  - configuration de la liaison, 137
- Communication série, 602
- Communication TCP/IP, 458
- Communication TeleService
  - TM\_MAIL, 713
- Communication, page Web standard, 545
- Commutation RTS, 624
- Comparaison, 208
- Comparaison des blocs de code, 729
- Comparaison et synchronisation des CPU hors ligne/en ligne, 729
- Compatibilité électromagnétique (CEM), 750
- Compensation de soudure froide
  - thermocouples, 827, 856
- Compteur d'heures de fonctionnement, 267
- Compteur rapide
  - Configuration, 369
  - fonctionnement, 361
  - HSC, 359
- Compteur rapide (HSC)
  - forçage permanent impossible, 738
- Compteurs
  - Configuration HSC, 369
  - fonctionnement HSC, 361
  - HSC (compteur rapide), 359
  - quantité, 21, 756, 766, 776, 786
  - taille, 22, 756, 766, 776, 786
- CONCAT (Concaténer), 285
- Conception d'un système d'automatisation, 155, 156
- Conditions ambiantes
  - conditions de service, 751
  - conditions de transport et stockage, 751
  - environnements industriels, 750
- Conditions de début, 627
- Conditions de fin, 630
- Conditions requises pour l'installation, 30
- Configuration
  - adresse AS-i, 514
  - adresse IP, 145
  - adresse MAC, 145
  - adresse PROFIBUS, 509
  - ajouter des modules, 130
  - AS-i, 514
  - charge due à la communication, 87
  - chargement dans la CPU, 179
  - communication API-API, 493
  - découvrir, 129
  - HSC (compteur rapide), 369
  - Interfaces de communication, 622
  - liaison réseau, 135
  - modules, 133
  - pages Web personnalisées, 569
  - pages Web personnalisées multilingues, 589
  - paramètres de la CPU, 131
  - paramètres de mise en route, 118
  - port AS-i, 513
  - port Ethernet, 145
  - port PROFIBUS, 509
  - Ports, 622
  - PROFIBUS, 509
  - PROFINET, 145
  - propriété Synchronisation de l'heure (PROFINET), 152
  - réception des messages, 627
  - temps de cycle, 85
- Configuration de la transmission de messages, 626
- Configuration de l'émission de messages, 626
  - exemple de programme point à point, 638
- Configuration des appareils, 127, 490
  - ajouter des modules, 130
  - ajouter un appareil, 128
  - AS-i, 514
  - chargement dans la CPU, 179
  - configuration de la CPU, 131
  - configuration des modules, 133
  - découvrir, 129
  - liaison réseau, 135
  - modules retirés, 41
  - port AS-i, 514
  - port Ethernet, 145
  - port PROFIBUS, 509
  - PROFIBUS, 509
  - PROFINET, 145
  - propriété Synchronisation de l'heure (PROFINET), 152
- Configuration des messages
  - émission, 626
  - Instructions, 635
  - réception, 627
- Configuration des paramètres
  - Emission, 496
  - LENGTH et BUFFER pour SEND\_PTP, 615
  - Réception, 497
- Configuration du port, 622

- erreurs, 607
- exemple de programme point à point, 638
- Instructions, 635
- Configuration matérielle, 127
  - ajouter des modules, 130
  - ajouter un appareil, 128
  - AS-i, 514
  - chargement dans la CPU, 179
  - configuration de la CPU, 131
  - configuration des modules, 133
  - découvrir, 129
  - liaison réseau, 135
  - port AS-i, 514
  - port Ethernet, 145
  - port PROFIBUS, 509
  - PROFIBUS, 509
  - PROFINET, 145
- Configuration OPC, 709
- Connecteur
  - montage et démontage, 58
- Connecteur de bus, 25
- Connexions maximum du serveur Web, 594
- Connexions, serveur Web, 594
- Conseils
  - charges de lampe, 68
  - charges inductives, 69
  - conseils de câblage, 65, 67
  - installation, 43
  - installation de la CPU, 50
  - isolation, 66
  - mise à la terre, 67
  - procédures d'installation, 49
- Conseils de câblage, 67
  - conditions requises, 65
  - dégagement pour l'écoulement d'air et le refroidissement, 44
  - mise à la terre, 67
- CONT (schéma à contacts)
  - éditeur de programme, 731
  - état, 731, 736
  - présentation, 165
  - visualisation, 731
- Contraintes
  - pages Web personnalisées, 573
  - serveur Web, 593
- Contrôle de cohérence, 183
- Contrôle de flux logiciel, 625
- Contrôle de flux matériel, 624
- Contrôle du flux, 623, 624
  - Configuration, 623
- Contrôle du programme (SCL), 236
  - CASE, 238
  - CONTINUE, 242
  - EXIT, 243
  - FOR, 239
  - GO TO, 243
  - IF-THEN, 237
  - REPEAT, 241
  - RETURN, 244
  - WHILE, 240
- Contrôler invalidité, instruction, 210
- Contrôler validité, instruction, 210
- CONV (Convertir), 227
- Conversion (instructions SCL), 228
- Cookie siemens\_automation\_language, 587
- Copie des blocs depuis une CPU en ligne, 179
- Copier zone (MOVE\_BLK), 220
- Copier zone contiguë (UMOVE\_BLK), 220
- COS (Cosinus), 218
- Courant alternatif
  - charges inductives, 69
  - conseils de câblage, 65, 67
  - conseils d'isolation, 66
  - mise à la terre, 67
- Courant continu
  - charges inductives, 69
  - conseils de câblage, 65, 67
  - conseils d'isolation, 66
  - mise à la terre, 67
- CPU
  - 1211C, caractéristiques, 754
  - 1211C, schémas de câblage, 763
  - 1212C, caractéristiques, 764
  - 1212C, schémas de câblage, 773
  - 1214C, caractéristiques, 774
  - 1214C, schémas de câblage, 783
  - 1215C, caractéristiques, 784
  - 1215C, schémas de câblage, 795
  - acquisition des valeurs d'un DB, 732
  - adresse AS-i, 514
  - adresse IP, 145
  - adresse MAC, 145
  - Adresse MAC, 145, 150
  - adresse PROFIBUS, 509
  - affectation d'une adresse IP à une CPU en ligne, 144
  - affichage des adresses MAC et IP, 150
  - ajouter des modules, 130
  - ajouter un appareil, 128
  - Appareils IHM, 27
  - AS-i, 513
  - besoins en courant, 883
  - bilan de consommation, 45
  - bornier de connexion, 58

boutons MARCHE/ARRET, 38  
 câble d'extension, 59  
 carte mémoire, 115, 878  
 carte programme, 115, 121  
 carte transfert, 115, 118  
 Carte transfert vide, 126  
 charge due à la communication, 87  
 chargement dans la CPU, 179  
 Charger dans l'appareil, 150  
 charges de lampe, 68  
 charges inductives, 69  
 Communication Board (CB), 25  
 comparaison et synchronisation des blocs, 729  
 compteur d'heures de fonctionnement, 267  
 configuration de la communication avec des IHM, 491  
 configuration de plusieurs CPU, 493  
 configuration des appareils, 127  
 configuration des modules, 133  
 configuration des paramètres, 131  
 configuration des voies d'impulsion, 334  
 configuration du temps de cycle, 87  
 Configuration HSC, 369  
 conseils de câblage, 65, 67  
 conseils d'isolation, 66  
 copie des blocs depuis une CPU en ligne, 179  
 CPU non spécifiée, 129  
 création d'une carte programme, 121  
 création d'une carte transfert, 118  
 déblocage des sorties à l'état ARRET, 736  
 DEL de signalisation, 721  
 en ligne, 731  
 En ligne, 726  
 état MARCHE et ARRET, 727  
 états de fonctionnement, 73  
 exécution du programme, 71  
 forçage permanent, 736, 737  
 insertion de la carte mémoire, 116  
 installation, 49, 50  
 liaison réseau, 135  
 mise à la terre, 67  
 niveaux de sécurité, 175  
 Niveaux de sécurité, 175  
 nombre de liaisons de communication, 452  
 oubli du mot de passe, 126  
 panneau de commande, 38  
 panneau de commande (CPU en ligne), 727  
 paramètres de mise en route, 118  
 passage en ligne, 723  
 port AS-i, 513  
 port Ethernet, 145  
 port PROFIBUS, 509

présentation, 19  
 PROFIBUS, 509  
 PROFINET, 145  
 PROFINET IO, 497  
 propriété Synchronisation de l'heure, 152  
 protection d'accès, 175  
 protection du savoir-faire, 176  
 protection par mot de passe, 175  
 RD\_LOC\_T (Lire l'heure locale), 265  
 RD\_SYS\_T (Lire l'heure système), 265  
 récupération en cas d'oubli du mot de passe, 126  
 redéfinition des valeurs initiales d'un DB, 732  
 représentation des entrées analogiques (tension), 822, 852  
 restauration des réglages d'usine, 726  
 Signal Board (SB), 25  
 sorties d'impulsions, 332  
 surveillance du temps de cycle, 85  
 table de visualisation, 733  
 tableau comparatif, 20  
 temps de réponse indicielle, 760, 770, 780, 790  
 traitement de mise en route, 76  
 traitement des OB, 159  
 types de communication, 451  
 visualisation, 731  
 WR\_SYS\_T (Ecrire l'heure système), 265  
 zone thermique, 44, 48  
 CPU non spécifiée, 129  
 Création de DB de pages Web personnalisées, 571  
 Création de pages Web personnalisées, 554  
 Création d'une liaison réseau, 135  
 CTD (Décomptage), 202  
 CTRL\_PWM, 330  
 CTS, 624  
 CTU (Comptage), 202  
 CTUD (Comptage et décomptage), 202  
 Cycle  
   forçage permanent, 736, 737

## D

### Date

DTL (type de données date et heure long), 104  
 SET\_TIMEZONE (Définir fuseau horaire), 269  
 T\_ADD (Additionner temps), 264  
 T\_COMBINE (Combiner temps), 265  
 T\_CONV (Convertir temps), 263  
 T\_DIFF (Différence de temps), 264  
 T\_SUB (Soustraire temps), 264  
 type de données Date, 104  
 DB (bloc de données), 71, 162  
 acquisition des valeurs, 732

- numéros de DB valides, 71
  - redéfinition des valeurs initiales, 732
  - DB de commande pour les pages Web personnalisées
    - commandes et états de requête, 590
    - commandes globales, 590
    - paramètre pour l'instruction WWW, 571
  - DB de fragment (pages Web personnalisées)
    - génération, 571
  - Début de message, 627
  - DEC (Décrémenter), 215
  - Déclenchement dans la table de visualisation, 735
  - DECO (Décoder), 256
  - Découvrir pour télécharger une CPU en ligne, 129
  - Définition de plusieurs variables AWP, 567
  - Définition de types Enum, pages Web personnalisées, 563
  - Dégagement
    - écoulement d'air et refroidissement, 44
  - DEL de signalisation
    - état de la CPU, 721
    - instruction LED, 318
    - interface de communication, 599, 721
  - Délai inter-caractères, 632
  - DELETE (Insérer des caractères dans une chaîne), 288
  - DELETE (Supprimer des caractères dans une chaîne), 287
  - Démarrage à chaud, 73
  - Démarrage après mise sous tension, 73
    - traitement de mise en route, 76
  - Démarrage, page Web standard, 540
  - Dépannage
    - DEL de signalisation, 721
    - mémoire tampon de diagnostic, 728
  - DETACH, 307
  - DeviceStates, 320
  - Diagnostic
    - DEL de signalisation, 721
    - DeviceStates, 320
    - état, 501
    - GET\_DIAG, 322
    - indicateur d'état, 91
    - instruction LED, 318
    - mémoire tampon de diagnostic, 728
    - ModuleStates, 321
    - Ob d'alarme, 501
    - table de visualisation, 733
    - temps de cycle, 728
    - utilisation de la mémoire, 728
  - Diagnostic, page Web standard, 541
  - DIS\_AIRT (Inhiber le traitement des alarmes), 316
  - DIV (Division), 212
  - Diverses erreurs de paramètres de la communication
    - point à point, 604
  - Documentation, 4
  - Données de diagnostic
    - lues avec GET\_DIAG, 322
  - Dossiers, langues pour les pages Web personnalisées, 586
  - DPNRM\_DG, 304
  - DPRD\_DAT, 302
  - DPWR\_DAT, 302
  - DTL, type de données
    - instructions pour l'horloge système, 265
  - Durée d'utilisation électrique des relais, 754
- E**
- E/S
    - adressage, 99
    - charges inductives, 69
    - erreurs d'accès, PROFINET, 501
    - forçage permanent, 736, 737
    - indicateurs d'état analogiques, 722
    - indicateurs d'état TOR, 722
    - représentation des entrées analogiques (courant), 822, 852
    - représentation des entrées analogiques (tension), 822, 852
    - représentation des sorties analogiques (courant), 823, 853
    - représentation des sorties analogiques (tension), 823, 853
    - temps de réponse indicielle (CPU), 760, 770, 780, 790
    - temps de réponse indicielle (SB), 851
    - temps de réponse indicielle du module d'entrées-sorties (SM), 821
    - visualisation avec une table de visualisation, 733
    - visualisation de l'état dans CONT, 731
  - E/S analogiques
    - configuration, 134
    - conversion en unités physiques, 34, 99, 235
    - indicateurs d'état, 722
    - représentation des entrées (courant), 822, 852
    - représentation des entrées (tension), 822, 852
    - représentation des sorties (courant), 823, 853
    - représentation des sorties (tension), 823, 853
    - temps de réponse indicielle (CPU), 760, 770, 780, 790
    - temps de réponse indicielle (SB), 851
    - temps de réponse indicielle (SM), 821
  - E/S TOR
    - configuration, 134

- indicateurs d'état, 722
- Écoulement d'air, 44
- Editeur de programmes
  - acquisition des valeurs d'un DB, 732
  - redéfinition des valeurs initiales d'un DB, 732
- Éditeur de programmes
  - état, 731
  - visualisation, 731
- Emission de messages, configuration, 626
- Emission, configuration des paramètres, 136, 496, 525
- Emulateur de terminal pour l'exemple de programme point à point, 645
- EN et ENO (flux de courant), 174
- En ligne
  - acquisition des valeurs d'un DB, 732
  - Adresse IP, 726
  - affectation d'une adresse IP, 144
  - boutons MARCHE/ARRET, 38
  - comparaison et synchronisation, 729
  - état, 731
  - forçage permanent, 736, 737
  - heure, 726
  - mémoire tampon de diagnostic, 728
  - outils, 730
  - panneau de commande, 38, 727
  - passage en ligne, 723
  - redéfinition des valeurs initiales d'un DB, 732
  - table de visualisation, 731, 733
  - temps de cycle, 728
  - utilisation de la mémoire, 728
  - visualisation, 731
- EN\_AIRT (Valider le traitement des alarmes), 316
- ENCO (Encoder), 256
- Entrées et sorties
  - visualisation, 731
- Environnement matériel requis, 30
- Erreurs
  - erreurs de diagnostic, 84
  - erreurs de temps, 82
- Erreurs à l'exécution de la réception, 617
- Erreurs à l'exécution de l'émission, 615
- Erreurs de configuration de réception, 613
- Erreurs de configuration d'émission, 608
- Erreurs de gestion des signaux, 620, 622
- Etat
  - DEL de signalisation, 721
  - instruction LED, 318
- État
  - DEL de signalisation (interface de communication), 599
- Etat ARRET, 73, 727
  - boutons de la barre d'outils, 38
  - déblocage des sorties à l'état ARRET, 736
  - forçage permanent, 737
  - panneau de commande, 38
- Etat de fonctionnement, 38
  - changement ARRET/MARCHE, 727
  - états de fonctionnement de la CPU, 73
- Etat des variables, page Web standard, 546
- Etat MARCHE, 73, 77, 727
  - boutons de la barre d'outils, 38
  - forçage permanent, 737
  - panneau de commande, 38
- Etat MISE EN ROUTE
  - forçage permanent, 737
- Ethernet
  - adresse IP, 145
  - adresse MAC, 145
  - DPNRM\_DG, 304
  - DPRD\_DAT, 302
  - DPWR\_DAT, 302
  - GET, 520
  - ID de liaison, 455
  - liaison réseau, 135
  - mode ad hoc, 459
  - nombre de liaisons de communication, 452
  - présentation, 458
  - PUT, 520
  - RALRM, 295
  - RDREC, 292
  - T\_CONFIG, 480
  - TCON, 467
  - TDISCON, 467
  - TRCV, 467
  - TRCV\_C, 460
  - TSEND, 467
  - TSEND\_C, 460
  - TURCV, 475
  - TUSEND, 475
  - types de communication, 451
  - WRREC, 292
- Exécution d'événement, 79
- Exécution du programme, 71
- Exemple
  - communication point à point, 637
  - communication point à point, configuration, 638
  - communication point à point, émulateur de terminal, 645
  - communication point à point, exécution, 646
  - communication point à point, programmation STEP 7, 644
  - esclave Modbus, 701
  - pages Web personnalisées, 574
- EXP (Fonction exponentielle), 218

EXPT (Elever à la puissance), 218  
 Extension des fonctionnalités du S7-1200, 22

## F

FAQ, 4  
 FB (bloc fonctionnel)  
   numéros de FB valides, 71  
   présentation, 71  
 FC (fonction), 71, 160  
   numéros de FC valides, 71  
 FieldRead, 223  
 FieldWrite, 223  
 FILL\_BLK, 224  
 Fin de message, 630  
 FIND (Trouver des caractères dans une chaîne), 290  
 FLOOR (Arrondir à l'entier inférieur), 232  
 Fonction (FC)  
   appel de blocs de code dans le programme  
   utilisateur, 158  
   numéros de FC valides, 71  
   présentation, 71, 160  
   programmes linéaires et structurés, 156  
   protection du savoir-faire, 176  
 Fonctions mathématiques, 33, 211, 212  
 Forçage  
   table de visualisation, 733  
 Forçage de variables depuis un PC, 546  
 Forçage permanent, 736, 737  
   carte mémoire, 115  
   cycle, 737  
   entrées et sorties, 737  
   mémoire I, 737  
   Mémoire I, 736  
   périphérie d'entrée, 736, 737  
   table de visualisation, 733  
 Former le complément à 1 (INV), 256  
 FRAC (Décimales), 218  
 Fragments (pages Web personnalisées)  
   création par une commande AWP, 565  
   importation avec une commande AWP, 566  
 Fréquence, bits de memento de cadence, 92  
 Front descendant, 190  
 Front montant, 190

## G

Génération de DB de pages Web personnalisées, 571  
 gestion des blocs de données, 355  
 GET, 520  
   configuration de la liaison, 137

GET\_DIAG, 322  
 GetError, 251  
 GetErrorID, 253  
 Glisser-déplacer entre les éditeurs, 37  
 Guillemets et apostrophes, conventions pour le serveur  
 Web, 567

## H

Heure  
   configuration de la CPU en ligne, 726  
   DTL (type de données date et heure long), 104  
   RD\_LOC\_T (Lire l'heure locale), 265  
   RD\_SYS\_T (Lire l'heure système), 265  
   SET\_TIMEZONE (Définir fuseau horaire), 269  
   T\_ADD (Additionner temps), 264  
   T\_COMBINE (Combiner temps), 265  
   T\_CONV (Convertir temps), 263  
   T\_DIFF (Différence de temps), 264  
   T\_SUB (Soustraire temps), 264  
   TOD (type de données heure), 104  
   type de données Time, 104  
   WR\_SYS\_T (Ecrire l'heure système), 265  
 Heure locale  
   RD\_LOC\_T (Lire l'heure locale), 265  
 Homologation ATEX, 749  
 homologation CE, 747  
 Homologation C-Tick, 749  
 Homologation cULus, 748  
 Homologation FM, 748  
 Homologation pour le domaine maritime, 750  
 Homologations  
   homologation ATEX, 749  
   homologation CE, 747  
   homologation C-Tick, 749  
   homologation cULus, 748  
   homologation FM, 748  
   homologation pour le domaine maritime, 750  
 Horloge  
   horloge temps réel, 93  
   RD\_LOC\_T (Lire l'heure locale), 265  
   RD\_SYS\_T (Lire l'heure système), 265  
   WR\_SYS\_T (Ecrire l'heure système), 265  
 Horloge système  
   RD\_LOC\_T (Lire l'heure locale), 265  
   RD\_SYS\_T (Lire l'heure système), 265  
   WR\_SYS\_T (Ecrire l'heure système), 265  
 HSC (compteur rapide)  
   configuration, 369  
   fonctionnement, 359, 361  
 HTA (Hexadécimal en ASCII), 283  
 HTTP, connexions HTTP du serveur Web, 594

- I**
- Identification de la CPU, page Web standard, 541
  - Identification, page Web standard, 541
  - Idle line, 626, 628
  - Importation du certificat de sécurité Siemens, 596
  - IN\_RANGE (Valeur dans la plage), 209
  - INC (Incrémenter), 215
  - Indexation de tableaux par des variables, 224
  - Indicateur Premier cycle, 91
  - Indice variable pour un tableau, 224
  - Information sur le programme
    - Dans la structure d'appel, 183
  - Informations pour nous contacter, 3
  - Informations sur les modules, page Web standard, 542
  - Insertion de la carte mémoire dans la CPU, 116
  - Insertion d'instructions
    - Favoris, 32
  - Insertion d'opérations
    - Glisser-déplacer, 32
    - Glisser-déplacer entre les éditeurs, 37
  - Insertion d'un appareil
    - CPU non spécifiée, 129
  - Installation
    - bilan de consommation, 45
    - bornier de connexion, 58
    - câble d'extension, 59
    - carte SIM du TS Adapter, 62
    - charges de lampe, 68
    - charges inductives, 69
    - Communication Board (CB), 53
    - conseils, 43
    - conseils de câblage, 65, 67
    - conseils d'isolation, 66
    - CPU, 50
    - dégagement, 44
    - dimensions de montage, 48
    - écoulement d'air, 44
    - mise à la terre, 67
    - module de communication (CM), 57
    - module d'entrées-sorties (SM), 25, 55
    - montage mural de l'adaptateur TS, 64
    - présentation, 43, 49
    - refroidissement, 44
    - Signal Board (SB), 53
    - TS Adapter et module TS, 61
    - TS Adapter sur un profilé support, 64
    - zone thermique, 44, 48
  - Installation, conditions requises, 30
  - Instructions
    - ABS (Valeur absolue), 215
    - ACOS (Arc cosinus), 218
    - ADD (Addition), 212
    - ajouter des entrées ou sorties aux opérations
      - CONT ou LOG, 35
    - AND, 255
    - ASIN (Arc sinus), 218
    - ATAN (Arc tangente), 218
    - ATH (ASCII en hexadécimal), 282
    - ATTACH, 307
    - CALCULATE, 33, 211
    - calendrier, 263
    - CAN\_DINT (Annuler une alarme temporisée), 314
    - CASE (SCL), 238
    - CEIL (Arrondir à l'entier supérieur), 232
    - Codes d'état USS, 658
    - colonnes et en-têtes, 36, 683
    - comparaison, 208
    - compteur d'heures de fonctionnement, 267
    - compteurs, 202
    - CONCAT (Concaténer), 285
    - CONTINUE (SCL), 242
    - contrôle (SCL), 236
    - Contrôler invalidité, 210
    - Contrôler validité, 210
    - CONV (Convertir), 227
    - Copier zone (MOVE\_BLK), 220
    - Copier zone contiguë (UMOVE\_BLK), 220
    - COS (Cosinus), 218
    - CTD (Décomptage), 202
    - CTRL\_PWM, 330
    - CTU (Comptage), 202
    - CTUD (Comptage et décomptage), 202
    - DataLogClose (Fermer un journal), 342
    - DataLogCreate (Créer un journal), 337
    - DataLogNewFile (Créer un journal sur la base d'un journal existant), 344
    - DataLogOpen (Ouvrir un journal), 340
    - DataLogWrite (Ecrire dans un journal), 343
    - date, 263
    - DEC (Décrémenter), 215
    - DECO (Décoder), 256
    - DELETE (Insérer des caractères dans une chaîne), 288
    - DELETE (Supprimer des caractères dans une chaîne), 287
    - DETACH, 307
    - DeviceStates, 320
    - DIS\_AIRT (Inhiber le traitement des alarmes), 316
    - DIV (Division), 212
    - DPNRM\_DG, 304
    - DPRD\_DAT, 302
    - DPWR\_DAT, 302
    - EN\_AIRT (Valider le traitement des alarmes), 316
    - ENCO (Encoder), 256



- état, 731
- état des DEL, 318
- EXIT (SCL), 243
- EXP (Fonction exponentielle), 218
- EXPT (Elever à la puissance), 218
- Favoris, 32
- FieldRead, 223
- FieldWrite, 223
- FILL\_BLK, 224
- FIND (Trouver des caractères dans une chaîne), 290
- FLOOR (Arrondir à l'entier inférieur), 232
- fonctionnement des temporisations, 197
- FOR (SCL), 239
- forçage permanent, 736, 737
- FRAC (Décimales), 218
- Front descendant, 190
- Front montant, 190
- gestion des blocs de données, 355
- GET, 520
- GET\_DIAG, 322
- GetError, 251
- GetErrorID, 253
- Glisser-déplacer, 32
- Glisser-déplacer entre les éditeurs, 37
- GOTO (SCL), 243
- graduation valeurs analogiques, 34
- heure, 263
- horloge, 265
- HSC (compteur rapide), 359, 361
- HTA (Hexadécimal en ASCII), 283
- IF-THEN (SCL), 237
- IN\_RANGE (Valeur dans la plage), 209
- INC (Incrémenter), 215
- Insertion, 32
- instructions de conversion SCL, 228
- instructions extensibles, 35
- INV (Former le complément à 1), 256
- JMP, 244
- JMP\_LIST, 245
- LEFT (Lire les caractères de gauche d'une chaîne), 286
- LEN (Longueur), 284
- LIMIT, 217
- LN (Logarithme népérien), 218
- Logiques sur bits, 185
- mathématiques à virgule flottante, 218
- MAX (Calculer le maximum), 216
- MB\_CLIENT, 666
- MC\_ChangeDynamic, 422
- MC\_CommandTable, 419
- MC\_Halt, 408
- MC\_Home, 405
- MC\_MoveAbsolute, 410
- MC\_MoveJog, 417
- MC\_MoveRelative, 412
- MC\_MoveVelocity, 414
- MC\_Power, 401
- MC\_Reset, 404
- MID (Lire les caractères du milieu d'une chaîne), 286
- MIN (Calculer le minimum), 216
- Mise à 0, 188
- Mise à 1, 188
- MOD (modulo), 213
- ModuleStates, 321
- MOVE, 220
- MUL (Multiplication), 212
- MUX (Multiplexeur), 258
- N\_TRIG, 191
- NEG (négation), 214
- NORM\_X (Normaliser), 233
- OR, 255
- OUT\_RANGE (Valeur en dehors de la plage), 209
- P\_TRIG, 191
- paramètres communs, 487
- périphérie décentralisée AS-i, 291
- périphérie décentralisée PROFIBUS, 291
- périphérie décentralisée PROFINET, 291
- PID\_Compact, 374
- PORT\_CFG (configuration de port), 605
- PUT, 520
- QRY\_CINT (Interroger alarme cyclique), 312
- RALRM, 295
- RCV\_CFG (Configuration de réception), 609
- RCV\_PTP (Réception point à point), 617
- RCV\_RST (Réinitialiser récepteur), 618
- RD\_LOC\_T (Lire l'heure locale), 265
- RD\_SYS\_T (Lire l'heure système), 265
- RDREC, 292
- RE\_TRIGR, 85, 249
- Renvoyer la valeur (RET), 248
- REPEAT (SCL), 241
- Repère de saut, 244
- REPLACE (Remplacer des caractères dans une chaîne), 289
- RETURN (SCL), 244
- RIGHT (Lire les caractères de droite d'une chaîne), 286
- ROL et ROR (Rotation vers la gauche et Rotation vers la droite), 262
- ROUND, 231
- RT (Réinitialiser temporisation), 193
- S\_CONV (Conversions valeur en chaîne), 271

S\_MOVE (Copier chaîne), 271  
 SCALE\_X (Mettre à l'échelle), 233  
 SEL (Sélectionner), 258  
 SEND\_CFG (Configuration d'émission), 607  
 SEND\_PTP (Emission de données point à point), 613  
 SET\_CINT (Régler alarme cyclique), 310  
 SET\_TIMEZONE, 269  
 SGN\_GET (Lire signaux RS232), 619  
 SGN\_SET (Activer signaux RS232), 621  
 SHL et SHR (Décaler vers la gauche et Décaler vers la droite), 261  
 SIN (Sinus), 218  
 SQR (Carré), 218  
 SQRT (Racine carrée), 218  
 SRT\_DINT (Démarrer une alarme temporisée), 314  
 STP (Arrêter), 251  
 STRG\_VAL (Chaîne en valeur), 271  
 SUB (Soustraction), 212  
 SWAP, 226  
 SWITCH, 246  
 T\_ADD (Additionner temps), 264  
 T\_COMBINE (Combiner temps), 265  
 T\_CONFIG, 480  
 T\_CONV (Convertir temps), 263  
 T\_DIFF (Différence de temps), 264  
 T\_SUB (Soustraire temps), 264  
 TAN (Tangente), 218  
 TCON, 467  
 TDISCON, 467  
 Temporisation, 193  
 TOF (Temporisation "Retard à la retombée"),  
 TON (Temporisation "Retard à la montée"),  
 TONR (Temporisation "Retard à la montée mémorisé"),  
 TP (Temporisation "Impulsion"),  
 TRCV, 467  
 TRCV\_C, 460, 496  
 TRUNC (Arrondir par troncature), 231  
 TSEND, 467  
 TSEND\_C, 460, 495  
 TURCV, 475  
 TUSEND, 475  
 UFILL\_BLK (Compléter zone contiguë), 224  
 USS\_DRV, 651  
 USS\_PORT, 654  
 USS\_RPM, 655  
 USS\_WPM, 656  
 VAL\_STRG (Valeur en chaîne), 271  
 versions des instructions, 36, 683  
 visualisation, 731  
 WHILE (SCL), 240

WR\_SYS\_T (Ecrire l'heure système), 265  
 WRREC, 292  
 WWW (activation des pages Web personnalisées), 571  
 XOR (OU exclusif), 255  
 Instructions extensibles, 35  
 Instructions logiques sur bits, 185  
 Instructions point à point, valeurs en retour, 603  
 Interface utilisateur  
   Vue du portail, 31  
   Vue du projet, 31  
 Interfaces de communication  
   ajouter des modules, 130  
   CB 1241 RS485, 874  
   CM 1241 RS232, 876  
   Configuration, 622  
   configuration des appareils, 127  
   DEL de signalisation, 721  
   programmation, 635  
   RS232 et RS485, 599  
   tableau comparatif des modules, 22  
 Intro, page Web standard, 539  
 ISO sur TCP  
   configuration des liaisons, 136  
   ID de liaison, 455  
   mode ad hoc, 459  
   paramètres, 138  
 ISO sur TCP, protocole, 458  
 Isolation, conseils, 66

## J

JMP, 244  
 JMP\_LIST, 245  
 JMPN, 244  
 Journal de données  
   DataLogClose (Fermer un journal), 342  
   DataLogCreate (Créer un journal), 337  
   DataLogNewFile (Créer un journal sur la base d'un journal existant), 344  
   DataLogOpen (Ouvrir un journal), 340  
   DataLogWrite (Ecrire dans un journal), 343  
   exemple de programme, 351  
   limites de taille, 348  
   présentation des journaux de données, 335  
   structure des enregistrements de données, 336  
   visualisation de journaux de données, 346  
 Journaux, page Web standard, 548

**L**

Langues pour les pages Web personnalisées, 586  
 Lecture de variables HTTP, 559  
 LED (Lire l'état de DEL), 318  
 LEFT (Lire les caractères de gauche d'une chaîne), 286  
 LEN (Longueur), 284  
 LENGTH, paramètre de SEND\_PTP, 615  
 Liaison à une CPU ou carte mémoire, 177  
 Liaison active/passive, 453  
 Liaison appareil local/partenaire, 453  
 Liaison réseau  
   configuration, 135  
   plusieurs CPU, 493, 494, 498, 509, 513  
 Liaisons  
   configuration, 138  
   ID de liaison, 455  
   liaison S7, 524  
   nombre de liaisons (PROFINET), 452  
   partenaires, 136, 525  
   protocoles Ethernet, 524  
   types de communication, 451  
   types de liaisons multinoeuds, 524  
 Liaisons multinoeuds  
   protocoles Ethernet, 524  
   types de liaisons, 524  
 Ligne d'assistance, 3  
 LIMIT, 217  
 Lire l'état de DEL, 318  
 Listage HTML, exemple de pages Web personnalisées, 580  
 LN (Logarithme népérien), 218  
 LOG (logigramme), 166  
 Longueur  
   Message, 633  
 Longueur de message, 632  
 Longueur de message maximale, 632  
 Longueur m, 633  
 Longueur n, 633

**M**

maître AS-i CM 1243-2  
   caractéristiques du module, 511  
 maître AS-i CM 1243-2, 511  
 Manuels, 4  
 Masque de sous-réseau, 146  
 Mathématiques à virgule flottante, 218  
 MAX (Calculer le maximum), 216  
 MB\_CLIENT, 666  
 MB\_COMM\_LOAD, 684

MB\_MASTER, 687  
 MB\_SERVER, 673  
 MB\_SLAVE, 693  
 MC\_ChangeDynamic, 422  
 MC\_CommandTable, 419  
 MC\_Halt, 408  
 MC\_Home, 405  
 MC\_MoveAbsolute, 410  
 MC\_MoveJog, 417  
 MC\_MoveRelative, 412  
 MC\_MoveVelocity, 414  
 MC\_Power, 401  
 MC\_Reset, 404  
 Memento de cadence  
   octet, 92  
 Mémoire  
   adresses de périphérie d'entrée (table de forçage permanent), 736  
   I (mémoire image des entrées), 96  
   L (mémoire locale), 94  
   M (mémentos), 97  
   Memento de cadence, 90  
   Memento système, 90  
   Mémoire de chargement, 88  
   Mémoire de travail, 88  
   Mémoire rémanente, 88  
   Mémoire temporaire, 98  
   Q (mémoire image des sorties), 96  
   visualisation de l'utilisation de la mémoire, 728  
 Mémoire de chargement, 20  
   carte mémoire, 115  
   carte programme, 115  
   carte transfert, 115  
   CPU 1211C, 754  
   CPU 1212C, 764  
   CPU 1214C, 774  
   CPU 1215C, 784  
 Mémoire de chargement, pages Web personnalisées, 573  
 Mémoire de travail, 20  
   CPU 1211C, 754  
   CPU 1212C, 764  
   CPU 1214C, 774  
   CPU 1215C, 784  
 Mémoire I  
   adresses de périphérie d'entrée (table de forçage permanent), 736  
   forçage permanent, 736, 737  
   HSC (compteur rapide), 361  
   table de forçage permanent, 736  
   table de visualisation, 731  
   visualisation, 731

- visualiser CONT, 731
- Mémoire image
  - état, 731, 736
  - forçage permanent, 736, 737
  - visualisation, 731
- Mémoire Q
  - configuration des voies d'impulsion, 334
  - sorties d'impulsions, 332
- Mémoire rémanente, 20, 88
  - CPU 1211C, 754
  - CPU 1212C, 764
  - CPU 1214C, 774
  - CPU 1215C, 784
- Mémoire tampon de diagnostic, 92, 728
- MID (Lire les caractères du milieu d'une chaîne), 286
- MIN (Calculer le minimum), 216
- Mise à 0, 188
- Mise à 1, 188
- Mise à jour du firmware, 123
- Mise à l'échelle de valeurs analogiques, 34, 235
- Mise en file d'attente, 79
- MOD (modulo), 213
- MODBUS
  - exemple d'esclave Modbus, 701
  - MB\_CLIENT, 666
  - MB\_COMM\_LOAD, 684
  - MB\_MASTER, 687
  - MB\_SERVER, 673
  - MB\_SLAVE, 693
  - versions, 36, 683
- Mode ad hoc
  - ISO sur TCP, 459
  - TCP, 459
- Modification
  - état de l'éditeur de programme, 731
- Modification des paramètres de STEP 7, 37
- Module de communication
  - CM 1241 RS422/RS485, caractéristiques, 877
- Module de communication (CM)
  - ajout d'un module CM 1243-5 (maître DP), 508
  - ajout d'un module maître AS-i CM 1243-2, 512
  - ajouter des modules, 130
  - besoins en courant, 883
  - CM 1241 RS232, caractéristiques, 876
  - configuration des appareils, 127
  - configuration des paramètres, 133
  - configuration pour l'exemple de programme point à point, 638
  - DEL de signalisation, 599, 721
  - démontage, 57
  - installation, 57
  - présentation, 25
  - programmation, 635
  - réception de données, 617
  - RS232 et RS485, 599
  - tableau comparatif, 22
- Module d'entrées-sorties (SM)
  - ajouter des modules, 130
  - besoins en courant, 883
  - câble d'extension, 59
  - configuration des paramètres, 133
  - démontage, 56
  - installation, 55
  - présentation, 25
  - représentation des entrées analogiques (courant), 822, 852
  - représentation des entrées analogiques (tension), 822, 852
  - représentation des sorties analogiques (courant), 823, 853
  - représentation des sorties analogiques (tension), 823, 853
  - SM 1221, 796
  - SM 1222, 799, 801
  - SM 1222 DQ8 RLY inverseur, 799
  - SM 1223, 806
  - SM 1223, caractéristiques, 809
  - SM 1231 AI 4 x 13 bits, 812
  - SM 1231 AI 4 x 16 bits TC, 824
  - SM 1231 AI 4 x RTD x 16 bits, 830
  - SM 1231 AI 8 x 16 bits TC, 824
  - SM 1231 AI 8 x RTD x 16 bits, 830
  - SM 1232 AQ 2 x 14 bits, 816
  - SM 1232 AQ 4 x 14 bits, 816
  - SM 1234 AI 4 x 13 bits / AQ 2 x 14 bits, 818
  - temps de réponse indicielle, 821
- Module d'entrées-sorties (SM) analogique
  - SM 1231 AI 4 x RTD x 16 bits, 830
  - SM 1231 AI 8 x 13 bits, 812
  - SM 1231 AI 8 x 16 bits TC, 824
  - SM 1231 AI 8 x RTD x 16 bits, 830
  - SM 1231 AI 4 x 16 bits TC, 824
  - SM 1232 AQ 2 x 14 bits, 816
  - SM 1232 AQ 4 x 14 bits, 816
  - SM 1234 AI 4 x 13 bits / AQ 2 x 14 bits, 818
- Module d'entrées-sorties (SM) TOR
  - SM 1221, 796
  - SM 1222, 799
  - SM 1223, 806
- Module d'entrées-sorties TOR
  - SM 1222, 801
- Modules
  - caractéristiques de la CPU 1211C, 754
  - caractéristiques de la CPU 1212C, 764

- caractéristiques de la CPU 1214C, 774
  - caractéristiques de la CPU 1215C, 784
  - CB 1241 RS485, 874
  - CM 1241 RS232, 876
  - Communication Board (CB), 25
  - configuration des paramètres, 133
  - module de communication (CM), 25
  - module d'entrées-sorties (SM), 25
  - module d'entrées-sorties SM 1231 AI 8 x 16 bits TC, 824
  - processeur de communication (CP), 25
  - SB 1221 DI 4, 200 kHz, 836
  - SB 1222 DQ 4, 200 kHz, 838
  - SB 1223 DI 2 / DQ 2, 844
  - SB 1223 DI 2 / DQ 2, 200 kHz, 841
  - SB 1231 AI 1 x 12 bits, 847
  - SB 1231 AI 1 x 16 bits RTD, 858
  - SB 1232 AQ 1x12 bits, 849
  - Signal Board (SB), 25
  - Signal Board SB 1231 AI 1 x 16 bits
  - Thermocouple, 854
  - SM 1221, 796
  - SM 1222, 799, 801
  - SM 1222 DQ8 RLY inverseur, 799
  - SM 1223, 806
  - SM 1231 AI 4 x 16 bits TC, 824
  - SM 1231 AI 4 x RTD x 16 bits, 830
  - SM 1231 AI 8 x 13 bits, 812
  - SM 1231 AI 8 x RTD x 16 bits, 830
  - SM 1232 AQ 2 x 14 bits, 816
  - SM 1232 AQ 4 x 14 bits, 816
  - SM 1234 AI 4 x 13 bits / AQ 2 x 14 bits, 818
  - tableau comparatif, 22
  - zone thermique, 44, 48
  - Modules de remplacement, 41
  - Modules retirés, 41
  - ModuleStates, 321
  - Montage
    - bornier de connexion, 58
    - câble d'extension, 59
    - charges de lampe, 68
    - charges inductives, 69
    - Communication Board (CB), 53
    - conseils, 43
    - conseils de câblage, 65, 67
    - CPU, 50
    - dégagement, 44
    - dimensions, 48
    - écoulement d'air, 44
    - isolation, 66
    - mise à la terre, 67
    - module de communication (CM), 57
    - module d'entrées-sorties (SM), 55
    - présentation, 49
    - refroidissement, 44
    - Signal Board (SB), 53
    - zone thermique, 44, 48
  - MOVE, 220
  - MRES
    - panneau de commande, 38
  - MUL (Multiplication), 212
  - MUX (Multiplexeur), 258
  - My Documentation Manager, 4
- ## N
- N\_TRIG, 191
  - NEG (négation), 214
  - Niveau de protection
    - bloc de code, 176
    - CPU, 175
    - liaison à une CPU ou carte mémoire, 177
    - Oubli du mot de passe, 126
  - Nombres
    - binaires, 101
    - entiers, 102
    - réels, 103
  - Noms d'appareils
    - PROFINET IO, 498
  - Noms d'appareils en ligne
    - PROFINET IO, 724
  - NORM\_X (Normaliser), 233
  - Normalisation de valeurs analogiques, 235
  - NTP (Network time protocol), 151
  - Numéro de port, 458
  - Numéros de port
    - limités, 489
  - Numéros de référence
    - alimentation PM 1207, 889
    - borniers de connexion, 890
    - câbles d'extension, 890
    - cartes mémoire, 890
    - commutateur Ethernet CSM 1277, 889
    - Connecteurs et bornes, 889
    - CPU, 887
    - documentation, 891
    - Interfaces de communication (CM, CB et CP), 888, 889
    - logiciel de programmation, 891
    - logiciel de visualisation, 891
    - modules d'entrées-sorties (SM), 887
    - pupitres HMI Basic Panels, 890
    - Signal Boards (SB), 888
    - simulateurs, 890

STEP 7, 891  
WinCC, 891

## O

OB d'alarme cyclique, 78  
OB d'alarme de diagnostic, 79  
OB d'alarme de processus, 79  
OB de démarrage, 78  
OB d'erreur de temps, 79  
Objets technologiques  
    HSC (compteur rapide), 361  
Octet de memento système, 91  
Open User Communication, valeurs en retour des instructions, 488  
OR (OU), 255  
Oubli du mot de passe, 126  
OUT\_RANGE (Valeur en dehors de la plage), 209  
Outils en ligne et de diagnostic  
    chargement dans la CPU à l'état MARCHE, 739  
Ouverture/fermeture de session  
    pages Web standard, 538

## P

P\_TRIG, 191  
Pages HTML personnalisées, 553  
    accès aux données du S7-1200, 555  
    actualisation, 555  
    conception, 554  
    emplacements, 569  
    emplacements des langues, 589  
Pages Web  
    STEP 7, 4  
Pages Web personnalisées, 533, 553  
    accès depuis le PC, 573  
    activation de l'instruction WWW, 571  
    activation et désactivation à partir du DB de commande, 590  
    actualisation, 555  
    chargement des DB correspondants dans la CPU, 573  
    commande de DB de fragment manuels, 590  
    commandes AWP pour accéder aux données du S7-1200, 555  
    configuration, 569  
    configuration multilingue, 589  
    contraintes de mémoire de chargement, 573  
    création avec un éditeur HTML, 554  
    création de fragments, 565  
    écriture de variables, 557

    écriture de variables spéciales, 561  
    exemple, 574  
    génération de blocs de programme, 571  
    gestion des caractères spéciaux, 567  
    importation de fragments, 566  
    lecture de variables, 556  
    lecture de variables spéciales, 559  
    listage HTML, 580  
    plusieurs langues, 586  
    programmation dans STEP 7, 571  
    suppression de blocs de programme, 571  
Pages Web standard, 533  
    accès depuis le PC, 535  
    accès sécurisé, 536  
    changement de l'état de fonctionnement, 540  
    communication, 545  
    Démarrage, 540  
    Diagnostic, 541  
    disposition, 537  
    Etat des variables, 546  
    Identification, 541  
    Informations sur les modules, 542  
    Intro, 539  
    Journaux, 548  
    ouverture et fermeture de session, 538  
    restrictions de cookies, 595  
    restrictions JavaScript, 594  
Pages Web STEP 7, 4  
Panneau de commande, 38  
    états de fonctionnement de la CPU, 73  
Paramétrage, 161  
Paramètres, 37  
Paramètres de mise en route, 118  
Paramètres de sortie, 161  
    configuration des voies d'impulsion, 334  
    sorties d'impulsions, 332  
Parité, 623  
Pas de redémarrage, 73  
Passage de MARCHE à ARRET, 93  
Pause, 626, 627  
PID  
    algorithme PID\_3Step, 371, 380  
    algorithme PID\_Compact, 371, 374  
    PID\_3STEP, 380  
    PID\_Compact, 374  
    présentation, 370  
Podcasts, 4  
Pointeur  
    présentation, 108  
Pointeurs  
    type de données Any, 110  
    type de données Pointer, 109

- type de données Variant, 111
- PORT\_CFG (configuration de port), 605
- Portail TIA
  - Vue du portail, 31
  - Vue du projet, 31
- Position de caractère
  - longueur de message, 633
- Priorité
  - classe de priorité, 78
  - priorité dans le traitement, 79
- Processeur de communication (CP)
  - ajouter des modules, 130
  - configuration des appareils, 127
  - configuration des paramètres, 133
  - présentation, 25
  - tableau comparatif, 22
- PROFIBUS
  - adresse PROFIBUS, 509
  - ajout d'un esclave DP, 508
  - ajout d'un module CM 1243-5 (maître DP), 508
  - DPNRM\_DG, 304
  - DPRD\_DAT, 302
  - DPWR\_DAT, 302
  - esclave, 502
  - GET, 520
  - instructions de périphérie décentralisée, 291
  - liaison réseau, 135, 509
  - liaison S7, 524
  - maître, 502
  - module CM 1242-5 (esclave DP), 502
  - module CM 1243-5 (maître DP), 503
  - propriétés de l'adresse PROFIBUS, 510
  - PUT, 520
  - RALRM, 295
  - RDREC, 292
  - WRREC, 292
- Profilé support, 49
- PROFINET
  - adresse IP, 145
  - adresse MAC, 145
  - affectation d'adresse IP, 153
  - communication API-API, 493
  - communication CPU à CPU, 493
  - configuration de la communication entre CPU et IHM, 491
  - configuration de l'adresse IP, 131
  - diagnostic, 501
  - DPRD\_DAT, 302
  - DPWR\_DAT, 302
  - GET, 520
  - ID de liaison, 455
  - instructions de périphérie décentralisée, 291
  - liaison réseau, 135, 493, 494, 498
  - liaison S7, 524
  - mode ad hoc, 459
  - nom et adresse des appareils, 153
  - nombre de liaisons de communication, 452
  - présentation, 458
  - propriété Synchronisation de l'heure, 152
  - propriétés de l'adresse Ethernet, 147
  - PUT, 520
  - RALRM, 295
  - RDREC, 292
  - synchronisation d'horloge, 131
  - T\_CONFIG, 480
  - TCON, 467
  - TDISCON, 467
  - temps de démarrage système, 153
  - Test d'un réseau, 149
  - TRCV, 467
  - TRCV\_C, 460
  - TSEND, 467
  - TSEND\_C, 460
  - TURCV, 475
  - TUSEND, 475
  - types de communication, 451
  - WRREC, 292
- PROFINET IO
  - affectation de noms d'appareils, 498
  - affectation de noms d'appareils en ligne, 724
  - affectation d'une CPU, 498
  - ajout d'un appareil, 497
  - appareils, 497
  - noms d'appareils, 498
  - noms d'appareils en ligne, 724
- PROFINET RT, 458
- Profondeur d'imbrication, 71
- Programmation
  - ajouter des entrées ou sorties aux opérations CONT ou LOG, 35
  - algorithme PID\_3Step, 371, 380
  - algorithme PID\_Compact, 371, 374
  - appel de blocs de code dans le programme utilisateur, 158
  - appels de blocs, 71
  - bloc de données (DB), 71
  - bloc de données d'instance (DB), 161
  - bloc d'organisation (OB), 159
  - bloc fonctionnel (FB), 71, 161
  - classe de priorité, 78
  - comparaison et synchronisation des blocs de code, 729
  - Compteur d'heures de fonctionnement, 267
  - compteurs, 202

- CONT (schéma à contacts), 165
- CPU non spécifiée, 129
- états de fonctionnement de la CPU, 73
- Favoris, 32
- flux de courant (EN et ENO), 174
- fonction (FC), 160
- Glisser-déplacer entre les éditeurs, 37
- Insertion d'opérations, 32
- instructions extensibles, 35
- Instructions point à point, 635
- liaison à une CPU ou carte mémoire, 177
- LOG (logigramme), 166
- modules retirés, 41
- numéros de FC, FB et DB valides, 71
- PID\_3STEP, 380
- PID\_Compact, 374
- présentation de PID, 370
- programme linéaire, 156
- programme structuré, 156
- RD\_LOC\_T (Lire l'heure locale), 265
- RD\_SYS\_T (Lire l'heure système), 265
- SCL (Structured Control Language), 166, 167, 168
- types de blocs de code, 71
- valeur initiale d'un FB, 161
- WR\_SYS\_T (Ecrire l'heure système), 265
- Programmation du changement de langue d'une page Web personnalisée, 587
- Programmation linéaire, 156
- Programmation point à point, 635
- Programmation STEP 7
  - exemple de programme point à point, 644
  - pages Web personnalisées, 571
- Programmation structurée, 156, 158
  - appel de blocs, 158
- Programme
  - acquisition des valeurs d'un DB, 732
  - appel de blocs de code dans le programme utilisateur, 158
  - bloc d'organisation (OB), 159
  - carte mémoire, 115
  - chargement dans la CPU, 179
  - classe de priorité, 78
  - copie des blocs depuis une CPU en ligne, 179
  - liaison à une CPU ou carte mémoire, 177
  - programmes linéaires et structurés, 156
  - protection par mot de passe, 176
  - redéfinition des valeurs initiales d'un DB, 732
- Programme utilisateur
  - ajouter des entrées ou sorties aux opérations CONT ou LOG, 35
  - appel de blocs de code dans le programme utilisateur, 158
- bloc d'organisation (OB), 159
- carte mémoire, 115
- carte programme, 115
- carte transfert, 115
- chargement dans la CPU, 179
- copie des blocs depuis une CPU en ligne, 179
- Favoris, 32
- Glisser-déplacer entre les éditeurs, 37
- Insertion d'opérations, 32
- instructions extensibles, 35
- liaison à une CPU ou carte mémoire, 177
- programmes linéaires et structurés, 156
- protection par mot de passe, 176
- Projet
  - accès restreint à une CPU, 175
  - carte programme, 121
  - carte transfert, 118
  - carte transfert vide, 126
  - chargement dans la CPU, 179
  - comparaison et synchronisation, 729
  - liaison à une CPU ou carte mémoire, 177
  - oubli du mot de passe, 126
  - protection d'accès, 175
  - protéger un bloc de code, 176
- Propriété Synchronisation de l'heure, 152
- Propriétés de la CPU, pages Web personnalisées, 569
- Propriétés de la CPU, pages Web personnalisées multilingues, 589
- Protection contre la copie
  - liaison à une CPU ou carte mémoire, 177
- Protection d'accès
  - CPU, 175
- Protection du savoir-faire
  - protection par mot de passe, 176
- Protection par mot de passe
  - accès à la CPU, 175
  - bloc de code, 176
  - carte transfert vide, 126
  - CPU, 175
  - liaison à une CPU ou carte mémoire, 177
  - Oubli du mot de passe, 126
  - protection contre la copie, 177
- Protocole
  - Communication, 602
  - Communication programmable, 602
  - ISO sur TCP, 458
  - Modbus, 602
  - PROFINET RT, 458
  - TCP, 458
  - UDP, 458
  - USS, 602
- Protocoles Ethernet, 458



liaisons multinoeuds, 524  
 PTO (Sortie de trains d'impulsions)  
   configuration des voies d'impulsion, 334  
   CTRL\_PWM, 330  
   fonctionnement, 332  
   forçage permanent impossible, 738  
 Pupitres (HMI), 27  
 Pupitres opérateur, 27  
 PUT, 520  
   configuration de la liaison, 137  
 PWM (Modulation de largeur d'impulsion)  
   configuration des voies d'impulsion, 334  
   CTRL\_PWM, 330  
   fonctionnement, 332  
   forçage permanent impossible, 738

## Q

QRY\_CINT (Interroger alarme cyclique), 312

## R

RALRM, 295, 298  
 RCV\_CFG (Configuration de réception), 609  
 RCV\_PTP (Réception point à point), 617  
 RCV\_RST (Réinitialiser récepteur), 618  
 RD\_LOC\_T (Lire l'heure locale), 265  
 RD\_SYS\_T (Lire l'heure système), 265  
 RDREC, 292, 298  
 RE\_TRIGR, 249  
 READ\_DBL, 355  
 Réception de messages, configuration, 627  
   exemple de programme point à point, 639  
 Réception, configuration des paramètres, 497  
 Redéfinition des valeurs initiales d'un DB, 732  
 Référencement de types Enum, pages Web personnalisées, 564  
 Références croisées, 183  
   Introduction, 183  
   Usages, 183  
 Refroidissement, 44  
 Réinitialiser temporisation (RT), 193  
 Renvoyer la valeur (RET), 248  
 Repère de saut, 244  
 Répertoires, langues pour les pages Web personnalisées, 586  
 REPLACE (Remplacer des caractères dans une chaîne), 289  
 Restauration des réglages d'usine, 726  
 Restrictions cookies, pages Web standard, 595  
 Restrictions JavaScript, pages Web standard, 594

Retard RTS OFF, 626  
 Retard RTS ON, 626  
 RIGHT (Lire les caractères de droite d'une chaîne), 286  
 ROL et ROR (Rotation vers la gauche et Rotation vers la droite), 262  
 ROUND, 231  
 Routeur IP, 145  
 RS232 et RS485, modules de communication, 599  
 RT (Réinitialiser temporisation), 193  
 RTS, 624  
 RTS toujours activé, 625

## S

S\_CONV (Conversions valeur en chaîne), 271  
 S\_MOVE (Copier chaîne), 271  
 S7-1200  
   acquisition des valeurs d'un DB, 732  
   adaptateur TS Adapter, 22  
   adresse AS-i, 514  
   adresse IP, 145  
   adresse MAC, 145  
   adresse PROFIBUS, 509  
   ajouter des modules, 130  
   ajouter un appareil, 128  
   Appareils IHM, 27  
   AS-i, 514  
   bilan de consommation, 45  
   bornier de connexion, 58  
   boutons MARCHE/ARRET, 38  
   câble d'extension, 59  
   carte mémoire, 878  
   carte programme, 121  
   carte transfert, 118  
   carte transfert vide en cas d'oubli du mot de passe, 126  
   charge due à la communication, 87  
   charges de lampe, 68  
   charges inductives, 69  
   Communication Board (CB), 25  
   compare les blocs de code, 729  
   configuration des appareils, 127  
   configuration des modules, 133  
   configuration des paramètres de la CPU, 131  
   Configuration HSC, 369  
   conseils de câblage, 65, 67  
   conseils d'isolation, 66  
   dégagement, 44  
   dimensions de montage, 48  
   écoulement d'air, 44  
   états de fonctionnement, 73

- fonctionnement, 733
- forçage permanent, 736, 737
- installation, 49
- installation de la CPU, 50
- installation d'un CB, 53
- Installation d'un CM, 57
- Installation d'un SB, 53
- Installation d'un SM, 55
- liaison réseau, 135
- mise à la terre, 67
- module de communication (CM), 25
- module d'entrées-sorties (SM), 25
- modules, 22
- oubli du mot de passe, 126
- panneau de commande, 38
- paramètres de mise en route, 118
- port AS-i, 513
- port Ethernet, 145
- port PROFIBUS, 509
- présentation de la CPU, 19
- processeur de communication (CP), 25
- PROFIBUS, 509
- PROFINET, 145
- protection d'accès, 175
- protection du savoir-faire, 176
- protection par mot de passe, 175
- redéfinition des valeurs initiales d'un DB, 732
- refroidissement, 44
- Signal Board (SB), 25
- sorties d'impulsions, 332
- tableau comparatif des modèles de CPU, 20
- temps de cycle, 87
- traitement de mise en route, 76
- visualisation, 731
- zone thermique, 44, 48
- SB 1221
  - schéma de câblage des SB 1221 DI 4, 200 kHz, 837
- SB 1222
  - schéma de câblage du SB 1222 DQ 4 x 24 VDC, 200 kHz, 840
- SB 1223
  - schéma de câblage des SB 1223 DI 2 / DQ 2, 200 kHz, 843
  - schéma de câblage du SB 1223 DI 2 / DQ 2, 846
- SB 1231 AI 1 x 16 bits Thermocouple
  - tableau de sélection de filtre, 857
- SB 1232
  - schéma de câblage du SB 1232 AQ 1 x 12 bits, 850
- SCALE\_X (Mettre à l'échelle), 233
- Schémas de câblage
  - CPU 1211C, 763
  - CPU 1212C, 773
  - CPU 1214C, 783
  - CPU 1215C, 795
  - SB 1221 DI 4, 200 kHz, 837
  - SB 1222 DQ 4, 200 kHz, 840
  - SB 1223 DI 2 / DQ 2, 846
  - SB 1223 DI 2 / DQ 2, 200 kHz, 843
  - SB 1231 AI 1 x 16 bits RTD, 860
  - SB 1231 AI 1 x 16 bits Thermocouple, 857
  - SB 1231 AI x 12 bits, 848
  - SB 1232 AQ 1 x 12 bits, 850
  - SM 1221, module d'entrées-sorties, 797
  - SM 1222, 801
  - SM 1223, module d'entrées-sorties, 806, 809
  - SM 1231 AI 4 x 16 bits TC, 826
  - SM 1231 AI 8 x 16 bits TC, 826
  - SM 1231 d'entrées analogiques, 816
  - SM 1231 RTD 4 x 16 bits, 832
  - SM 1231 RTD 8 x 16 bits, 832
  - SM 1232 de sorties analogiques, 817
  - SM 1234 d'entrées/sorties analogiques, 820
- SCL (Structured Control Language)
  - ABS (Valeur absolue), 215
  - ACOS (Arc cosinus), 218
  - adressage, 168
  - algorithme PID\_3Step, 371, 380
  - algorithme PID\_Compact, 371, 374
  - AND, 255
  - appel de blocs, 158
  - appel d'un FB ou d'une FC, 168
  - Arrondir, 231
  - Arrondir par troncature, 231
  - ASIN (Arc sinus), 218
  - ATAN (Arc tangente), 218
  - ATH (ASCII en hexadécimal), 282
  - ATTACH et DETACH, 307
  - CAN\_DINT (Annuler une alarme temporisée), 314
  - CASE, 238
  - CEIL (Arrondir à l'entier supérieur), 232
  - Comparaison, 208
  - Compteur d'heures de fonctionnement, 267
  - compteurs, 202
  - CONCAT (Concaténer), 285
  - conditions, 168
  - CONTINUE, 242
  - contrôle du programme, 236
  - Contrôler validité, 210
  - CONV (Convertir), 227
  - Copier valeur (MOVE), 220
  - COS (Cosinus), 218
  - CTD (Décomptage), 202
  - CTU (Comptage), 202

- CTUD (Comptage et décomptage), 202  
 DataLogClose (Fermer un journal), 342  
 DataLogCreate (Créer un journal), 337  
 DataLogNewFile (Créer un journal sur la base d'un journal existant), 344  
 DataLogOpen (Ouvrir un journal), 340  
 DataLogWrite (Ecrire dans un journal), 343  
 DEC (Décrémenter), 215  
 DECO (Décoder), 256  
 DELETE (Insérer des caractères dans une chaîne), 288  
 DELETE (Supprimer des caractères dans une chaîne), 287  
 DeviceStates, 320  
 DIS\_AIRT (Inhiber le traitement des alarmes), 316  
 éditeur de programmes, 167  
 EN et ENO (flux de courant), 174  
 EN\_AIRT (Valider le traitement des alarmes), 316  
 ENCO (Encoder), 256  
 état des DEL, 318  
 EXIT, 243  
 EXP (Fonction exponentielle), 218  
 expressions, 168  
 EXPT (Elever à la puissance), 218  
 FILL\_BLK, 224  
 FIND (Trouver des caractères dans une chaîne), 290  
 FLOOR (Arrondir à l'entier inférieur), 232  
 fonctionnement des temporisations, 197  
 fonctions mathématiques, 212  
 FOR, 239  
 FRAC (Décimales), 218  
 GET\_DIAG, 322  
 GOTO, 243  
 HTA (Hexadécimal en ASCII), 283  
 IF-THEN, 237  
 IN\_RANGE (Valeur dans la plage), 209  
 INC (Incrémenter), 215  
 instructions de contrôle, 168, 236, 237, 238, 239, 240, 241, 242, 243, 244  
 instructions de conversion, 228  
 instructions logiques sur bits, 185  
 INV (Former le complément à 1), 256  
 JMP\_LIST, 245  
 LEFT (Lire les caractères de gauche d'une chaîne), 286  
 LEN (Longueur), 284  
 LIMIT, 217  
 LN (Logarithme népérien), 218  
 mathématiques à virgule flottante, 218  
 MAX (Calculer le maximum), 216  
 MC\_ChangeDynamic, 422  
 MC\_CommandTable, 419  
 MC\_Halt, 408  
 MC\_Home, 405  
 MC\_MoveAbsolute, 410  
 MC\_MoveJog, 417  
 MC\_MoveRelative, 412  
 MC\_MoveVelocity, 414  
 MC\_Power, 401  
 MC\_Reset, 404  
 MID (Lire les caractères du milieu d'une chaîne), 286  
 MIN (Calculer le minimum), 216  
 Mise à 1 et Mise à 0, 188  
 MOD (modulo), 213  
 ModuleStates, 321  
 MUX (Multiplexeur), 258  
 N\_TRIG, 191  
 NEG (négation), 214  
 NORM\_X (Normaliser), 233  
 opérateurs, 168  
 OR, 255  
 OUT\_RANGE (Valeur en dehors de la plage), 209  
 P\_TRIG, 191  
 Permutation, 226  
 PID\_3STEP, 380  
 PID\_Compact, 374  
 présentation, 166  
 présentation de PID, 370  
 priorité des opérateurs, 168  
 QRY\_CINT (Interroger alarme cyclique), 312  
 RD\_LOC\_T (Lire l'heure locale), 265  
 RD\_SYS\_T (Lire l'heure système), 265  
 REPEAT, 241  
 REPLACE (Remplacer des caractères dans une chaîne), 289  
 RETURN, 244  
 RIGHT (Lire les caractères de droite d'une chaîne), 286  
 ROL et ROR (Rotation vers la gauche et Rotation vers la droite), 262  
 S\_CONV (Conversions valeur en chaîne), 271  
 S\_MOVE (Copier chaîne), 271  
 SCALE\_X (Mettre à l'échelle), 233  
 section Var, 167  
 SEL (Sélectionner), 258  
 SET\_CINT (Régler alarme cyclique), 310  
 SET\_TIMEZONE (Définir fuseau horaire), 269  
 SHL et SHR (Décaler vers la gauche et Décaler vers la droite), 261  
 SIN (Sinus), 218  
 SQR (Carré), 218  
 SQRT (Racine carrée), 218

- SRT\_DINT (Démarrer une alarme temporisée), 314
- STRG\_VAL (Chaîne en valeur), 271
- SWITCH, 246
- T\_ADD (Additionner temps), 264
- T\_COMBINE (Combiner temps), 265
- T\_CONV (Convertir temps), 263
- T\_DIFF (Différence de temps), 264
- T\_SUB (Soustraire temps), 264
- TAN (Tangente), 218
- temporisations, 193
- UFILL\_BLK (Compléter zone contiguë), 224
- VAL\_STRG (Valeur en chaîne), 271
- WHILE, 240
- WR\_SYS\_T (Ecrire l'heure système), 265
- XOR (OU exclusif), 255
- Sécurité
  - CPU, 175
  - liaison à une CPU ou carte mémoire, 177
  - oubli du mot de passe, 126
  - protection contre la copie, 177
  - protection d'accès, 175
  - protection du savoir-faire pour un bloc de code, 176
- SEL (Sélectionner), 258
- SEND\_CFG (Configuration d'émission), 607
- SEND\_PTP (Emission de données point à point), 613
  - paramètres LENGTH et BUFFER, 615
- Séquence de caractères
  - début de message, 628
  - Fin de message, 632
- Serveur Web, 533
  - activation, 534
  - connexions HTTP maximum, 594
  - contraintes, 593
  - conventions pour les guillemets et apostrophes, 567
  - pages Web standard, 535
  - vitesse d'actualisation, 534
- Serveur Web, pages Web personnalisées, 553
- Service et assistance, 3
- SET\_CINT (Régler alarme cyclique), 310
- SET\_TIMEZONE (Définir fuseau horaire), 269
- SGN\_GET (Lire signaux RS232), 619
- SGN\_SET (Activer signaux RS232), 621
- SHL et SHR (Décaler vers la gauche et Décaler vers la droite), 261
- siemens\_automation\_language, cookie, 587
- Signal Board (SB)
  - ajouter des modules, 130
  - besoins en courant, 883
  - configuration des paramètres, 133
  - démontage, 53
  - installation, 53
  - présentation, 25
  - représentation des entrées (courant), 822, 852
  - représentation des entrées (tension), 822, 852
  - représentation des sorties analogiques (courant), 823, 853
  - représentation des sorties analogiques (tension), 823, 853
  - SB 1221 DI 4, 200 kHz, 836
  - SB 1222 DQ, 200 kHz, 838
  - SB 1223 DI 2 / DQ 2, 844
  - SB 1223 DI 2 / DQ 2, 200 kHz, 841
  - SB 1231 AI 1 x 12 bits, 847
  - SB 1231 AI 1 x 16 bits RTD, 858
  - SB 1231 AI 1 x 16 bits Thermocouple, 854
  - SB 1232 AQ 1x12 bits, 849
  - temps de réponse indicielle, 851
- Signal Boards (SB) analogiques
  - SB 1231 AI 1 x 12 bits, 847
  - SB 1231 AI 1 x 16 bits RTD, 858
  - SB 1231 AI 1 x 16 bits Thermocouple, 854
  - SB 1232 AQ 1x12 bits, 849
- Simulateurs, 880
- Simulateurs d'entrées, 880
- SIN (Sinus), 218
- SM 1231 RTD
  - tableaux de sélection, 835, 862
- SM et SB
  - configuration des appareils, 127
  - tableau comparatif, 22
- SMS, 707
- Sorties d'impulsions, 332
- Sources d'informations, 4
- SQR (Carré), 218
- SQRT (Racine carrée), 218
- SRT\_DINT (Démarrer une alarme temporisée), 314
- STEP 7
  - acquisition des valeurs d'un DB, 732
  - affectation d'une adresse IP à une CPU en ligne, 144
  - ajout d'un périphérique PROFINET IO, 497
  - ajouter des entrées ou sorties aux opérations CONT ou LOG, 35
  - ajouter des modules, 130
  - ajouter un appareil, 128
  - appel de blocs de code dans le programme utilisateur, 158
  - appels de blocs, 71
  - AS-i, 514
  - bloc de données (DB), 71
  - bloc de données d'instance (DB), 161
  - bloc fonctionnel (FB), 71, 161
  - boutons MARCHE/ARRET, 38
  - carte mémoire, 115, 878

- carte programme, 115
  - carte transfert, 115
  - charge due à la communication, 87
  - chargement dans la CPU, 179
  - classe de priorité (OB), 78
  - comparaison et synchronisation, 729
  - compteur d'heures de fonctionnement, 267
  - compteurs, 202
  - configuration de la CPU, 131
  - configuration des appareils, 127
  - configuration des modules, 133
  - Configuration HSC, 369
  - copie des blocs depuis une CPU en ligne, 179
  - entrées et sorties affichées ou masquées, 35
  - états de fonctionnement, 73
  - Favoris, 32
  - fonction (FC), 160
  - fonctionnement, 733
  - forçage permanent, 736, 737
  - Glisser-déplacer entre les éditeurs, 37
  - Insertion d'opérations, 32
  - liaison réseau, 135
  - Modification des paramètres, 37
  - modules retirés, 41
  - numéros de FC, FB et DB valides, 71
  - panneau de commande, 38
  - port AS-i, 513
  - port Ethernet, 145
  - port PROFIBUS, 509
  - PROFIBUS, 509
  - PROFINET, 145
  - programmes linéaires et structurés, 156
  - propriété Synchronisation de l'heure (PROFINET), 152
  - protection par mot de passe, 176
  - RD\_LOC\_T (Lire l'heure locale), 265
  - RD\_SYS\_T (Lire l'heure système), 265
  - redéfinition des valeurs initiales d'un DB, 732
  - temps de cycle, 87
  - temps de cycle, 87
  - traitement de mise en route, 76
  - types de blocs de code, 71
  - valeur initiale d'un FB, 161
  - visualisation, 731
  - Vue du portail, 31
  - Vue du projet, 31
  - WR\_SYS\_T (Ecrire l'heure système), 265
  - STP (Arrêter), 251
  - STRG\_VAL (Chaîne en valeur), 271
  - String
    - présentation des données chaînes de caractères, 270
  - Structure d'appel, 183
  - Structure du programme, 158
    - appel de blocs de code, 158
  - SUB (Soustraction), 212
  - Surveillance
    - instruction LED, 318
  - Surveillance du temps de cycle, 249
  - SWAP, 226
  - SWITCH, 246
  - Synchronisation
    - propriété Synchronisation de l'heure (PROFINET), 152
- T**
- T\_ADD (Additionner temps), 264
  - T\_COMBINE (Combiner temps), 265
  - T\_CONFIG, 480
  - T\_CONV (Convertir temps), 263
  - T\_DIFF (Différence de temps), 264
  - T\_SUB (Soustraire temps), 264
  - Table de forçage permanent
    - adressage de la périphérie d'entrée, 736
    - forçage permanent, 736, 737
  - Table de visualisation
    - carte mémoire, 115
    - déblocage des sorties à l'état ARRET, 736
    - fonctionnement, 733
    - forçage permanent, 182
    - types de déclenchement, 735
    - visualisation, 731
  - Tableau comparatif
    - Modèles de CPU, 20
    - modules, 22
  - Tableau de comparaison
    - Appareils IHM, 27
  - Tableaux
    - accès aux éléments par une variable, 224
  - TAN (Tangente), 218
  - Task Cards
    - colonnes et en-têtes, 36, 683
  - TCON, 467
    - configuration, 136
    - ID de liaison, 455
    - paramètres de liaison, 138
  - TCON\_Param, 138
  - TCP
    - configuration des liaisons, 136
    - ID de liaison, 455
    - mode ad hoc, 459
    - paramètres, 138
    - protocole, 458

- TDISCON, 467
  - Téléchargement
    - certificat de sécurité Siemens pour le PC, 539, 596
    - mise à jour du firmware, 123
  - Telecontrol, 704
  - TeleService via GPRS, 704
  - Temporisation "Impulsion" (TP),
    - fonctionnement,
  - Temporisation "Retard à la montée mémorisé" (TONR),
    - fonctionnement,
  - Temporisation "Retard à la montée" (TON),
    - fonctionnement,
  - Temporisation "Retard à la retombée" (TOF),
    - fonctionnement,
  - Temporisations
    - fonctionnement, 197
    - quantité, 21, 756, 766, 776, 786
    - RT (Réinitialiser temporisation), 193
    - taille, 21, 756, 766, 776, 786
    - TOF (Temporisation "Retard à la retombée"),
    - TON (Temporisation "Retard à la montée"),
    - TONR (Temporisation "Retard à la montée mémorisé"),
    - TP (Temporisation "Impulsion"),
  - Temps d'attente, 79, 624
  - Temps de cycle
    - configuration, 87
    - présentation, 85
    - visualisation, 728
  - Tensions nominales, 752
  - Test
    - chargement dans la CPU à l'état MARCHE, 739, 745
  - Test du programme, 182
  - Thermocouples
    - compensation de soudure froide, 827, 856
    - principe de fonctionnement, 827, 856
    - tableau de sélection de filtre du module SM 1231
    - Thermocouple, 828
    - tableau de sélection de filtre du SB 1231
    - Thermocouple, 856
    - tableau de sélection du module SM 1231
    - Thermocouple, 828
    - tableau de sélection du SB 1231
    - Thermocouple, 856
  - TM\_MAIL, 713
  - Traitement image
    - état, 731
    - visualisation, 731
  - Transmission des données, déclenchement, 613
  - TRCV, 467
    - ID de liaison, 455
    - mode ad hoc, 459
  - TRCV\_C, 460, 496
    - configuration, 136
    - ID de liaison, 455
    - mode ad hoc, 459
    - paramètres de liaison, 138
  - TRCV\_C, configuration de l'instruction, 497
  - TRUNC (Arrondir par troncature), 231
  - TSAP, 458
    - limités, 489
  - TSAP (points d'accès au service transport), 138, 460, 494, 524
  - TSAP et numéros de port limités, 489
  - TSEND, 467
    - ID de liaison, 455
  - TSEND\_C, 495
    - configuration, 136
    - ID de liaison, 455
    - paramètres de liaison, 138
  - TSEND\_C, 495
  - TSEND\_C, configuration de l'instruction, 496
  - TURCV, 475
    - configuration, 136
    - paramètres de liaison, 138
  - TUSEND, 475
    - configuration, 136
    - paramètres, 138
  - Types de données, 100
    - Any (pointeur), 110
    - Bool, Byte, Word et DWord, 101
    - Char (caractère) et String (chaîne de caractères), 105
    - éditeur de type de données API, 108
    - Pointer (pointeur), 109
    - présentation des types de données pointeur, 108
    - Real, LReal (réels à virgule flottante), 103
    - Struc, 108
    - tableaux, 107
    - Time, Date, TOD (heure), DTL (date et heure long), 104
    - USInt, SInt, UInt, Int, UDIInt, DInt (entiers), 102
    - Variant (pointeur), 111
  - Types Enum dans les pages Web personnalisées, 563, 564
- ## U
- UDP
    - configuration des liaisons, 136
    - paramètres, 138
  - UDP, protocole, 458

UFILL\_BLK (Compléter zone contiguë), 224

## V

VAL\_STRG (Valeur en chaîne), 271

valeurs booléennes ou de bit, 95

Valeurs en retour

instructions point à point, 603

instructions pour la communication ouverte, 488

Valeurs initiales

acquisition et redéfinition des valeurs initiales d'un DB, 732

Variables

état, 731

forçage permanent, 736, 737

visualisation, 731

visualisation et forçage depuis un PC, 546

Versions des instructions, 36, 683

Visualisation

acquisition des valeurs d'un DB, 732

Appareils IHM, 27

état CONT, 731

État CONT, 731

forçage permanent, 737

redéfinition des valeurs initiales d'un DB, 732

table de forçage permanent, 736

table de visualisation, 731, 733

temps de cycle, 728

utilisation de la mémoire, 728

Visualisation de variables depuis un PC, 546

Visualisation du programme, 182

Vitesse de transmission, 623

Vue du portail, 31

Vue du projet, 31

Zones de mémoire

accès direct, 94

adressage des valeurs booléennes ou de bit, 95

mémoire image, 94

## W

WR\_SYS\_T (Ecrire l'heure système), 265

WRIT\_DBL, 355

WRREC, 292, 298

WWW (activation des pages Web personnalisées), 571

## X

XON / XOFF, 625

XOR (OU exclusif), 255

## Z

Zone thermique, 44, 48

