



Programmation du palettiseur en langage ST

Cahier des charges du palettiseur (1/2)

Cahier des charges

Analyse

Déclaration des données

Programmation en ST

Mise au point

Les écrans d'exploitation

Personnalisation

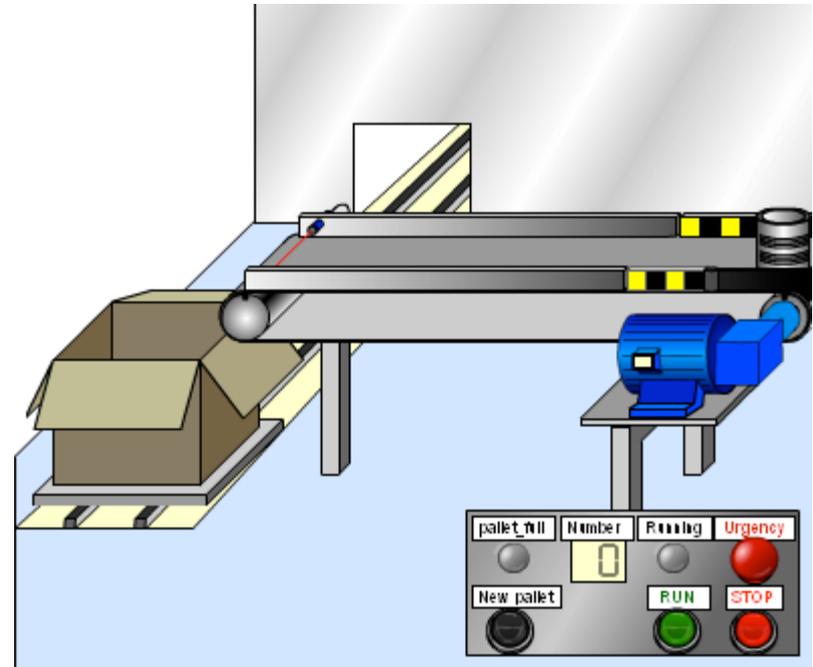
► Nous allons maintenant réaliser une extension du projet en rajoutant le palettiseur à l'installation précédente. Le convoyeur amène des pièces qui sont ensuite rangées par 10 sur une palette.

■ Les besoins en entrées :

- Une entrée pour le capteur optique permettant le comptage des pièces (**Optical_sensor**).
- Une entrée **New_palett** pour évacuer la palette, remettre le compteur (**Number**) à 0 et appeler une nouvelle palette.

■ Les besoins en sorties

- Une sortie pour le voyant **Palett_full**
- Une sortie **Blocker** pour piloter le bloqueur des pièces qui arrivent.



Utilisez les boutons de la maquette pour comprendre le cahier des charges : RUN, New Pallet



Programmation du palettiseur en langage ST

Cahier des charges du palettiseur (2/2)

Cahier des charges

Analyse

Déclaration des données

Programmation en ST

Mise au point

Les écrans d'exploitation

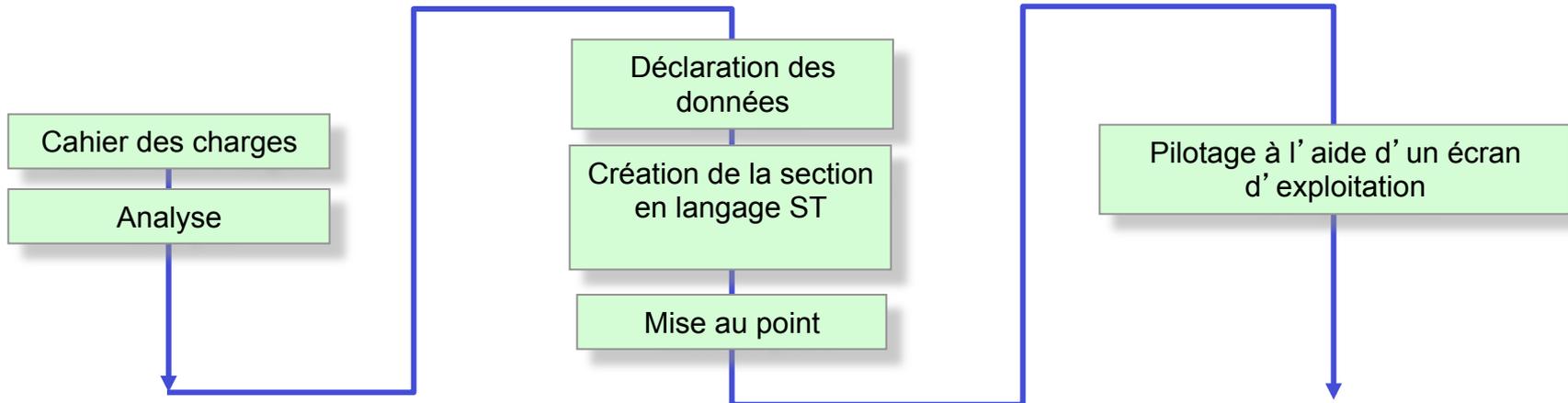
Personnalisation

La chronologie de développement que nous vous proposons est la suivante :

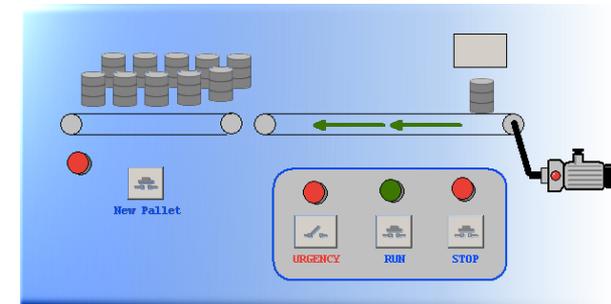
Prise en charge de l'application

Réalisation du projet

Ecrans d'exploitation



```
(* simulation du passage de pièce  
optical_sensor:= MOTOR and not  
  
If re (optical_sensor) then  
    inc (number);  
end_if;  
if re (new_pallet) then  
    number:=0;
```





Programmation du palettiseur en langage ST

Analyse du cahier des charges

Cahier des charges

Analyse

Déclaration des données

Programmation en ST

Mise au point

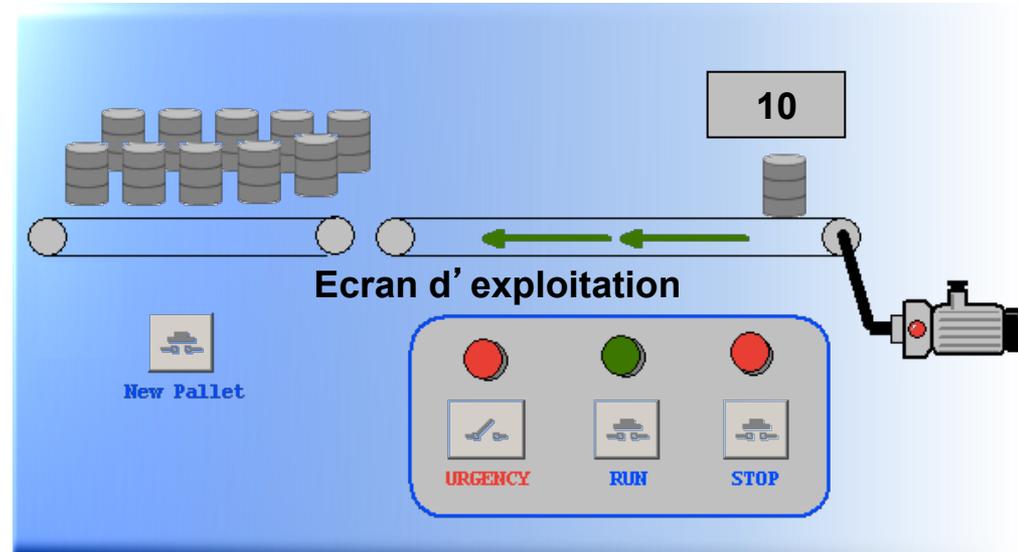
Les écrans d'exploitation

Personnalisation

Structure du programme et visualisation du procédé dans Unity Pro avec un écran d'exploitation .

■ Structure de la section Palettisation

- Si le convoyeur est en marche, les boîtes sont déposées sur le convoyeur.
- Les boîtes sont comptées. Lorsque l'on atteint 10 boîtes, la palette est pleine et le bloqueur est activé.
- Sur demande opérateur, la palette est évacuée pour en traiter une nouvelle.
- La section sera programmée en langage Littéral Structuré (ST).
- Un écran d'exploitation permet la commande et la visualisation de l'état du procédé.



Remarque :

Pour simuler l'approvisionnement des boîtes, chaque seconde par exemple, nous utiliserons le **bit système %S6**



Programmation du palettiseur en langage ST

Déclaration des données

Cahier des charges

Analyse

Déclaration des données

Programmation en ST

Mise au point

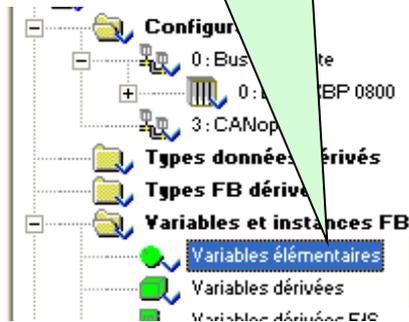
Les écrans d'exploitation

Personnalisation

Nous allons ajouter les nouvelles variables à la liste des variables déjà existantes.
Se déconnecter du simulateur (Menu **AUTOMATE** / **Déconnexion**) pour revenir en mode local.



Effectuer un double clic sur **Variables élémentaires** pour accéder à l'éditeur de données.



Symboliser toutes les données sans laisser d'espace dans le nom

Nom	Type	Adresse	Valeur	Commentaire	Utilisé
MOTOR	EBOOL			Commande du moteur	3
RUN	EBOOL			Départ convoyeur	2
RUNNING	EBOOL			Voyant marche mote...	2
STOP	EBOOL			Arrêt convoyeur	2
URGENCY	EBOOL			Arrêt d'urgence	2

L'écran apparaît avec les **variables déjà déclarées**.

Nom	Type	...	Valeur	Commentaire
BLOCKER	EBOOL			Bloqueur de pièces
NUMBER	INT			Nombre de pièces dans la palette
MOTOR	EBOOL			Commande du moteur
NEW_PALETT	EBOOL			Appel nouvelle palette
OPTICAL_SENSOR	EBOOL			Détecteur de passage des pièces
PALETT_FULL	EBOOL			Palette pleine
RUN	EBOOL			Départ convoyeur
RUNNING	EBOOL			Voyant marche moteur
STOP	EBOOL			Arrêt convoyeur
URGENCY	EBOOL			Arrêt d'urgence

Aide à la saisie du projet





Programmation du palettiseur en langage ST

Création de la section palettiseur (1/4)

Cahier des charges

Analyse

Déclaration des données

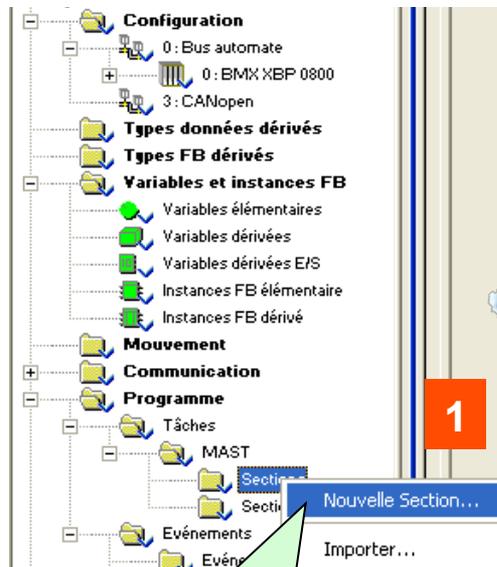
Programmation en ST

Mise au point

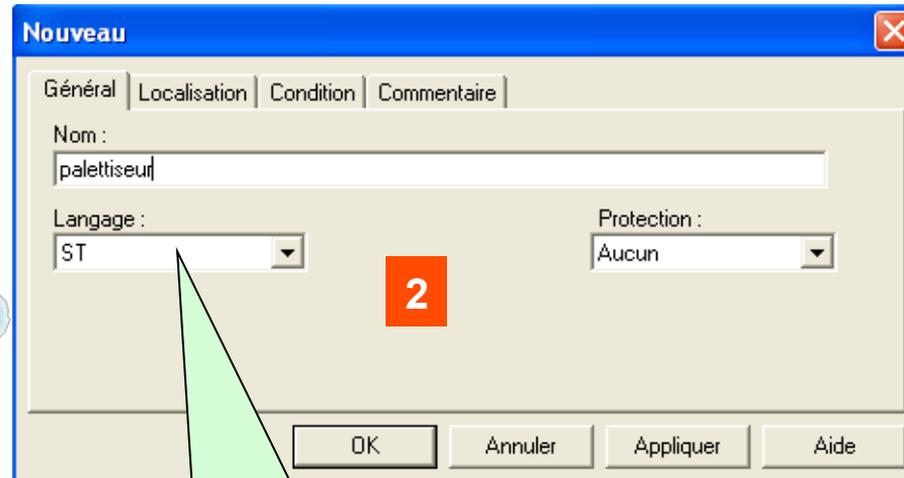
Les écrans d'exploitation

Personnalisation

Création de la section palettiseur en langage structuré (ST).



Effectuer un clic droit sur **Section** et sélectionner le menu **Nouvelle Section**.



Saisir le **Nom de la section** (Palettiseur) et sélectionner le **langage** à utiliser, ici **ST** puis valider par OK.





Programmation du palettiseur en langage ST

Création de la section palettiseur (2/4)

Cahier des charges

Analyse

Déclaration des données

Programmation en ST

Mise au point

Les écrans d'exploitation

Personnalisation

Création du programme en littéral structuré : logique de simulation du détecteur optique en utilisant des variables déjà déclarées

palettiseur : [MAST]
(*simulation du passage de pièces devant le détecteur *)

Saisie du commentaire entre (* et *)

1

Appeler par un clic droit l'assistant de saisie des données.

palettiseur : [MAST]
(*simulation du pass

2

Sélection de données

palettiseur : [MAST]
(*simulation du passag

3

Cliquer sur ... pour faire apparaître la liste des variables déjà déclarées.

palettiseur : [MAST]
(*simulation du passage de pièces devant le détecteur *)
OPTICAL_SENSOR

5

La variable est saisie.

Nom	Type	Valeur	Commentaire
BLOCKER	EBOOL		Bloqueur de pièces
NUMBER	INT		Nombre de pièces dans la palette
MOTOR	EBOOL		Commande du moteur
NEW_PALETT	EBOOL		Appel nouvelle palette
OPTICAL_SENSOR	EBOOL		Détecteur de passage des pièces
PALETT_FULL	EBOOL		Palette pleine

4

Faire un double-clic sur la variable OPTICAL_SENSOR.

Remarque : Les variables peuvent être également saisies directement.



Programmation du palettiseur en langage ST

Création de la section palettiseur (3/4)

Cahier des charges

Analyse

Déclaration des données

Programmation en ST

Mise au point

Les écrans d'exploitation

Personnalisation

Saisir le programme complet tel que ci-dessous.
Le contrôle du nombre de pièces se fait à l'aide de structure de contrôle : « If then else »
Un assistant de saisie est proposé.



2

Cliquer sur l'icône associée au IF
Un masque de saisie est affiché

1

Cliquer sur l'emplacement ou vous souhaitez insérer la structure de contrôle

```
(* Simulation présence pièce *)
Optical_sensor := Motor And not Palett_Full And ...
(* Comptage pièces *)
IF RE (Optical_sensor) THEN
INC (number);
END_IF;
if re (New_palett) then
    number:=0;
end_if;
if number=10 then
    Palett_Full:=true;
else
    Palett_full:=false;
end_if;

(*blocage de l'arrivée des pièces si palette pleine*)
blocker:=Palett_full;
```

Renseigner le masque

```
IF THEN
    ELSIF THEN
    ELSE
END_IF;
```

3

Aide à la saisie du projet



Programmation du palettiseur en langage ST

Création de la section palettiseur (4/4)

Cahier des charges

Analyse

Déclaration des données

Programmation en ST

Mise au point

Les écrans d'exploitation

Personnalisation

Analyse du projet, génération et transfert dans le simulateur pour la mise au point.



Sélectionner le menu **Analyse** pour contrôler les modifications, apporter les corrections si nécessaire, puis effectuer une génération partielle du code à l'aide du menu **Générer le projet**.

Passer en **Connexion** et **Transférer le projet** dans le simulateur par la même méthode que celle utilisée dans la phase 1 et passer en **RUN**



Remarque : la commande Générer le Projet/ Régénérer tout le projet enchaîne l'analyse, si celle-ci n'a pas été réalisée au préalable



Programmation du palettiseur en langage ST

Mise au point de la section palettiseur (1/2)

Cahier des charges

Analyse

Déclaration des données

Programmation en ST

Mise au point

Les écrans d'exploitation

Personnalisation

Des couleurs sont utilisées pour afficher l'état des variables en visualisation dynamique. Les fenêtres d'inspection permettent de compléter la visualisation dans l'éditeur.

Couleurs des variables en fonction de leur état

Type booléen :

Verte si la variable est TRUE,

Rouge si la variable est FALSE

Fond **Jaune** pour les autres types.

Pour visualiser la valeur d'une variable numérique dans une **info bulle** il suffit de pointer la variable avec la souris.

```
(* simulation du passage de pièces devant le détecteur*)
optical_sensor := MOTOR and not pallet_full and %S6;
(* Comptage pièces*)
If re (optical_sensor) then
    inc (number);
end_if;
if re (new_pallet) then
    number := 0;
end_if;
if number = 10 then
    pallet_full := true;
else
    pallet_full := false;
end_if;

(*blocage de l'arrivée des pièces si palette pleine*)
blocker := pallet_full;
```

The screenshot shows the ST editor interface. The code is color-coded: 'optical_sensor' is green (boolean TRUE), 'pallet_full' is red (boolean FALSE), and 'number' is yellow (numeric). An inspection window for the variable 'number' is open, showing its value as 6. A red circle highlights the 'lunette' icon in the toolbar.

Les **fenêtres inspection** permettent de visualiser l'état des variables. Pour cela cliquer sur la variable puis sur l'**icône lunette**. Les fenêtres inspection restent affichées malgré le scrolling



Programmation du palettiseur en langage ST

Mise au point de la section palettiseur (2/2)

Cahier des charges

Analyse

Déclaration des données

Programmation en ST

Mise au point

Les écrans d'exploitation

Personnalisation

Initialisation d'une table d'animation pour visualiser l'état des variables de la section Palettiseur. Modification des variables pour vérifier le fonctionnement de la machine.



```
(* simulation du passage de pièces devant le détecteur*)
optical_sensor:= MOTOR and not pallet_full;

If re (optical_sensor) then
  inc (number);
end_if;
if re (new_pallet) then
  number:=0;
end_if;
if number=10 then
  pallet_full:=true;
else
  pallet_full:=false;
end if;

(*blocage de l'arrivée des pièces*)
blocker:=pallet_full;
```

1

Sélectionner le programme de la Section palettiseur.

Effectuer un clic droit et sélectionner le menu Initialiser la table d'animation ou CTL + T.

2

Nom	Valeur	Type	Commentaire
Optical_sensor	0	EBOOL	Détecteur de pièces
Motor	0	EBOOL	Commande moteur
Palett_Full	0	EBOOL	Palette pleine
%S6	1	BOOL	
number	0	INT	Nombre de pièces
New_palett	0	EBOOL	Nouvelle palette
blocker	0	EBOOL	Bloqueur de pièces

