

# TIA Portal

HMI Panel KTP600 avec SIMATIC S7-1200

Présenté par : Ridha MAHJOUB

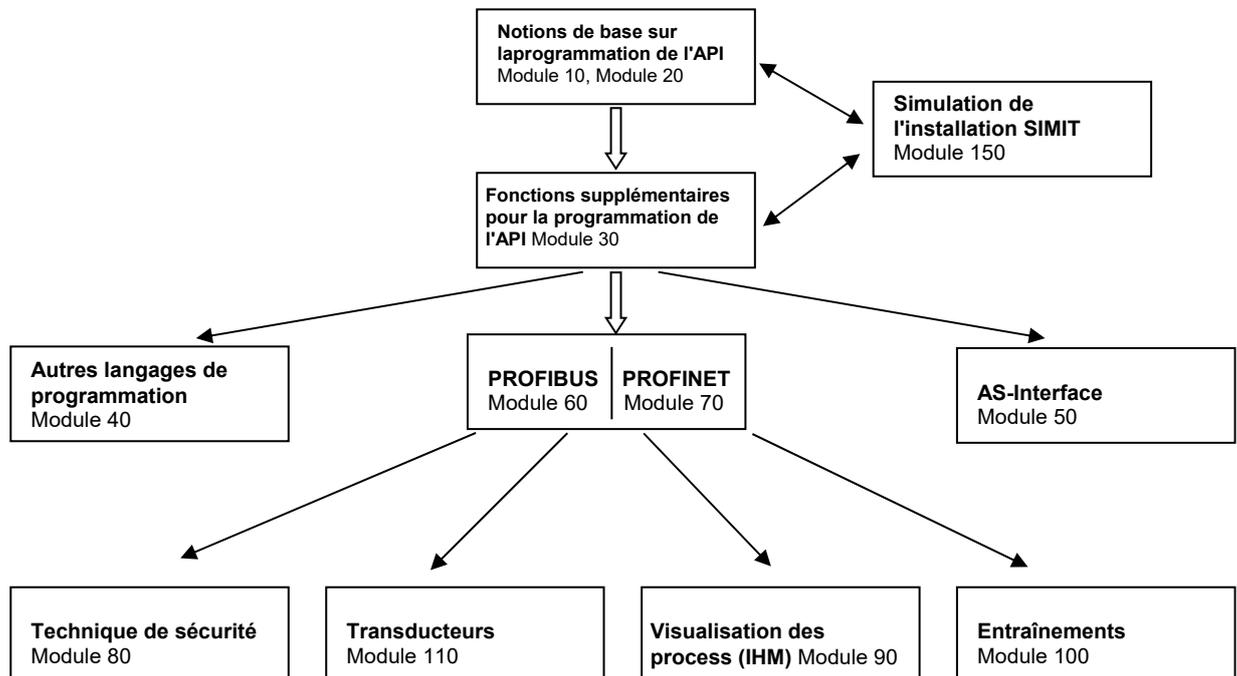
Agrégé de Génie Electrique, Maître Technologue à l'ISETSO



	<b>PAGE :</b>
1. Avant-propos .....	3
2. Notes concernant la programmation du SIMATIC S7-1200.....	5
2.1 Automate SIMATIC S7-1200.....	5
2.2 Logiciel de programmation STEP 7 Professional V11 (TIA Portal V11).....	5
2.3 Contrôle-commande avec WinCC.....	6
3. Commande d'un convoyeur avec compteur et multi-instance .....	7
3.1 Application à réaliser .....	7
3.2 Configuration .....	8
4 Insertion d'un Panel KTP600 PN dans le projet de la commande du convoyeur.....	8
4.1 Charger le modèle de projet et enregistrer à nouveau .....	9
5. Interface utilisateur de WinCC.....	21
5.1 Navigateur du projet .....	22
5.2 Barre de menus et boutons .....	22
5.3 Zone de travail.....	23
5.4 Outils.....	23
5.5 Fenêtre des propriétés .....	24
5.6 Vue détaillée.....	24
6. Vues de commande et connexions .....	25
6.1 Vue racine ou vue initiale .....	25
6.2 Connexions aux automates S7 .....	26
6.3 Affecter l'adresse IP .....	27
7 Configuration de la vue racine .....	28
7.1 Touche de fonction F6.....	29
7.2 Configuration des boutons Automatique et Manuel .....	30
7.3 Modifications dans le programme STEP 7 .....	34
7.4 Paramétrage de l'interface PG/PC pour la simulation Runtime .....	35
7.5 Démarrage de la configuration sous Runtime.....	36
7.6 Chargement et test de la configuration sur le Panel .....	37
7.7 Boutons Marche et Arrêt .....	38
7.8 Insertion de graphiques depuis le dossier graphique .....	41
7.9 Programme de simulation du mouvement de la bouteille .....	43
7.10 Activation du memento de cadence et affectation de MB100.....	45
7.11 Appel du bloc fonctionnel FB2 (Simulation) dans OB1 .....	46
7.12 Configuration du mouvement de la bouteille.....	48
7.13 Remise à zéro du compteur de bouteilles.....	52
7.14 Dessin de la caisse .....	54
7.15 Dessin des bouteilles dans la caisse .....	59

## 1. Avant-propos

Le contenu du module 010-080 est assigné à l'unité **"Notions de base sur la programmation d'API"**. Il décrit la configuration d'une visualisation WinCC entre le Panel KTP600 et une SIMATIC S7 1200.



### Objectif :

Dans ce module, le lecteur va être initié à la configuration d'un Panel pour visualiser des process en utilisant WinCC, en liaison avec un SIMATIC S7-1200.

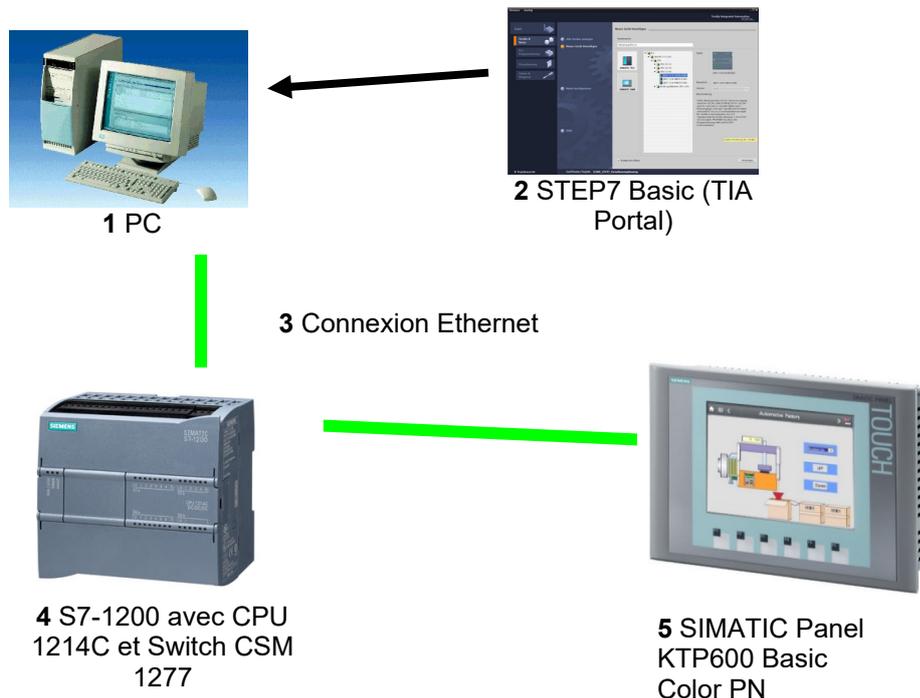
### Pré-requis :

Les connaissances suivantes sont requises pour l'étude de ce module :

- Connaissance pratique des systèmes d'exploitation Windows
- Notions de base sur la programmation d'API avec TIA Portal (par exemple le module 010-010 - Initiation à la programmation du SIMATIC S7-1200 avec TIA Portal V11)
- Blocs pour le SIMATIC S7-1200 (par exemple le module 010-020 – Types de blocs sur le SIMATIC S7-1200)
- Blocs de temporisations et de compteurs pour le SIMATIC S7-1200 (par exemple le module 010-030 – Multi-Instances sur le SIMATIC S7-1200)

**Configurations matérielles et logicielles requises**

- 1 PC Pentium 4, 1.7 GHz 1 (XP) – 2 Go RAM (Vista), env. 2 Go d'espace disponible  
Système d'exploitation Windows XP (Home SP3, Professional SP3)/Windows Vista (Home Premium SP1, Business SP1, Ultimate SP1
- 2 Logiciel STEP7 Basic V11 SP2 (Totally Integrated Automation (TIA) Portal V11)
- 3 Connexion Ethernet entre le PC, la CPU 1214C et le Panel
- 4 API SIMATIC S7-1200, p.ex. CPU 1214C avec Switch CSM 1277.  
Les entrées doivent être mises en évidence sur un pupitre.
- 5 SIMATIC Panel KTP600 Basic Color PN



---

---

## 2. Notes concernant la programmation du SIMATIC S7-1200

### 2.1 Automate SIMATIC S7-1200

L'automate SIMATIC S7-1200 est un mini-contrôleur modulaire utilisé pour les petites et moyennes performances.

Il existe un éventail complet de modules pour une adaptation optimisée à la tâche d'automatisation.

Le contrôleur S7 est composé d'une alimentation électrique, d'une CPU et de modules d'entrées/sorties pour les signaux numériques et analogiques.

Le cas échéant, des processeurs de communication et des modules fonctionnels sont ajoutés pour des tâches spéciales comme la commande de moteur pas à pas.

Le programme S7 permet à l'automate programmable industriel (API) de contrôler et commander une machine ou un processus. Les modules E/S sont interrogés dans le programme S7 au moyen d'adresses d'entrées (%E) et référencés au moyen d'adresses de sorties (%A).

Le système est programmé avec le logiciel STEP 7.

### 2.2 Logiciel de programmation STEP 7 Professional V11 (TIA Portal V11)

Le logiciel STEP 7 Professional V11 (TIA Portal V11) est l'outil de programmation des automates

- SIMATIC S7-1200
- SIMATIC S7-300
- SIMATIC S7-400
- SIMATIC WinAC

Avec STEP 7 Professional V11, les fonctions suivantes peuvent être utilisées pour automatiser une installation :

- Configuration et paramétrage du matériel
- Paramétrage de la communication
- Programmation
- Test, mise en service et dépannage avec les fonctions d'exploitation et de diagnostic
- Documentation
- Génération d'écrans de visualisation pour les Basic Panels SIMATIC avec WinCC Basic intégré.
- Il est également possible de générer des écrans de visualisation pour les PC et autres Panels à l'aide d'autres progiciels WinCC.

Toutes les fonctions sont détaillées dans l'aide en ligne.

## 2.3 Contrôle-commande avec WinCC

### Description du système

Sachant que les process deviennent de plus en plus complexes et que les machines et les systèmes doivent être toujours plus opérationnels, l'opérateur a besoin d'un outil efficace pour superviser et contrôler les équipements de production. Un système IHM (Interface Homme Machine) représente l'interface entre l'être humain (l'opérateur) et le process (machine/système). Le contrôle effectif du process s'effectue à travers le système de contrôle. Il y a donc une interface entre l'opérateur et WinCC (sur le pupitre opérateur) et une interface entre WinCC et le système de contrôle.

WinCC est le logiciel que vous utiliserez pour réaliser toutes les tâches de configuration requises.

WinCC Runtime est le logiciel qui permet de visualiser les process. Au Runtime, le projet est exécuté en mode process.

WinCC gère les tâches suivantes :

- **Représentation du process**

Le process est représenté sur le pupitre opérateur. Si, par exemple, un changement intervient dans le process, l'affichage est mis à jour sur le pupitre opérateur.

- **Commande du process**

L'opérateur peut commander le process via l'interface graphique. Par exemple, l'opérateur peut définir une consigne pour l'automate ou démarrer un moteur.

- **Affichage d'alarmes**

Si des états critiques surviennent dans le process, une alarme se déclenche automatiquement. Par exemple, quand une limite fixée est dépassée.

- **Archivage des valeurs de process et des alarmes**

Le système IHM peut archiver des alarmes et des valeurs de process. Cela vous permet de documenter les caractéristiques du process ou d'accéder ultérieurement à des données de production plus anciennes.

- **Documentation des valeurs et des alarmes**

Le système IHM peut afficher les alarmes et les valeurs de process sous forme de protocole. Vous pouvez ainsi afficher les données de production à chaque changement d'équipe.

- **Gestion des paramètres du process et des machines**

Le système IHM peut enregistrer les paramètres de process et des machines dans des recettes. Cela vous permet de transférer ces paramètres en une seule fois du pupitre opérateur à l'automate, par exemple pour modifier une variante de production.

### 3. Commande d'un convoyeur avec compteur et multi-instance

Pour visualiser notre process avec WinCC, un compteur et une multi-instance doivent être ajoutés à l'exemple de la commande d'un convoyeur.

Avec ce convoyeur, 20 bouteilles doivent toujours être acheminées vers une caisse. Quand la caisse est pleine, le convoyeur doit s'arrêter et la caisse doit être remplacée.

Avec le bouton S1, on peut sélectionner le mode de fonctionnement Manuel et avec le bouton S2, on peut sélectionner le mode Automatique.

En mode Manuel, le moteur est alimenté tant qu'on appuie sur le bouton S3 et que le bouton S4 n'est pas activé.

En mode Automatique, le moteur du convoyeur est allumé avec le bouton S3 et éteint avec le bouton S4 (NF).

De plus, un capteur 'B0' compte le nombre de bouteilles dans les caisses. Quand 20 bouteilles sont comptées, le convoyeur s'arrête.

Quand une nouvelle caisse est amenée, il faut le confirmer au moyen du bouton 'S5'.

#### Tableau d'affectations :

Adresse	Symbole	Commentaire
%E 0.0	S1	Bouton mode manuel, S1 NO
%E 0.1	S2	Bouton mode automatique S2 NO
%E 0.2	S3	Bouton Marche S3 NO
%E 0.3	S4	Bouton Arrêt, S4 NF
%E 0.6	S5	Bouton S5 NO Réinitialiser le compteur / nouvelle caisse
%E 0.7	B0	Capteur B0 NO Compteur bouteilles
%A 0.2	M01	Moteur du convoyeur M01

#### 3.1 Application à réaliser

Le contrôle du transport doit être effectué et contrôlé en utilisant le Panel.

Avec l'aide du Panel, les conditions suivantes doivent être remplies :

Le mode opérationnel est modifié à partir du Panel, et le nouveau mode choisi doit y apparaître.

Le moteur du convoyeur est démarré et arrêté à partir du Panel.

Le changement de caisse est confirmé sur le Panel.

Le transport des bouteilles et le remplissage de la caisse doivent être représentés graphiquement.

### 3.2 Configuration

La visualisation du process pour la commande du convoyeur est créée sous le logiciel de configuration STEP7 V11 à l'aide de la version intégrée de WinCC. Les valeurs du process sont représentées via des écrans et des objets sur l'écran. Les valeurs par défaut peuvent être transférées à l'automate en utilisant les éléments de commande. La communication entre le pupitre opérateur et la machine ou le process s'effectue via l'automate au moyen de **variables**. La valeur d'une variable est écrite dans une zone mémoire (adresse) de l'automate où elle est lue par le pupitre opérateur.

**La visualisation du process est enregistrée et chargée dans le Panel KTP600 Basic color PN. Après l'allumage du Panel, la commande du convoyeur peut être contrôlée et commandée.**

## 4 Insertion d'un Panel KTP600 PN dans le projet de la commande du convoyeur

La gestion du projet et sa programmation se font grâce au logiciel **Totally Integrated Automation Portal V11**.

Là, sous une même interface, les éléments tels que le contrôleur, la visualisation et la mise en réseau de la solution d'automatisation sont créés, paramétrés et programmés. Les outils en ligne sont disponibles pour les diagnostics d'erreur.

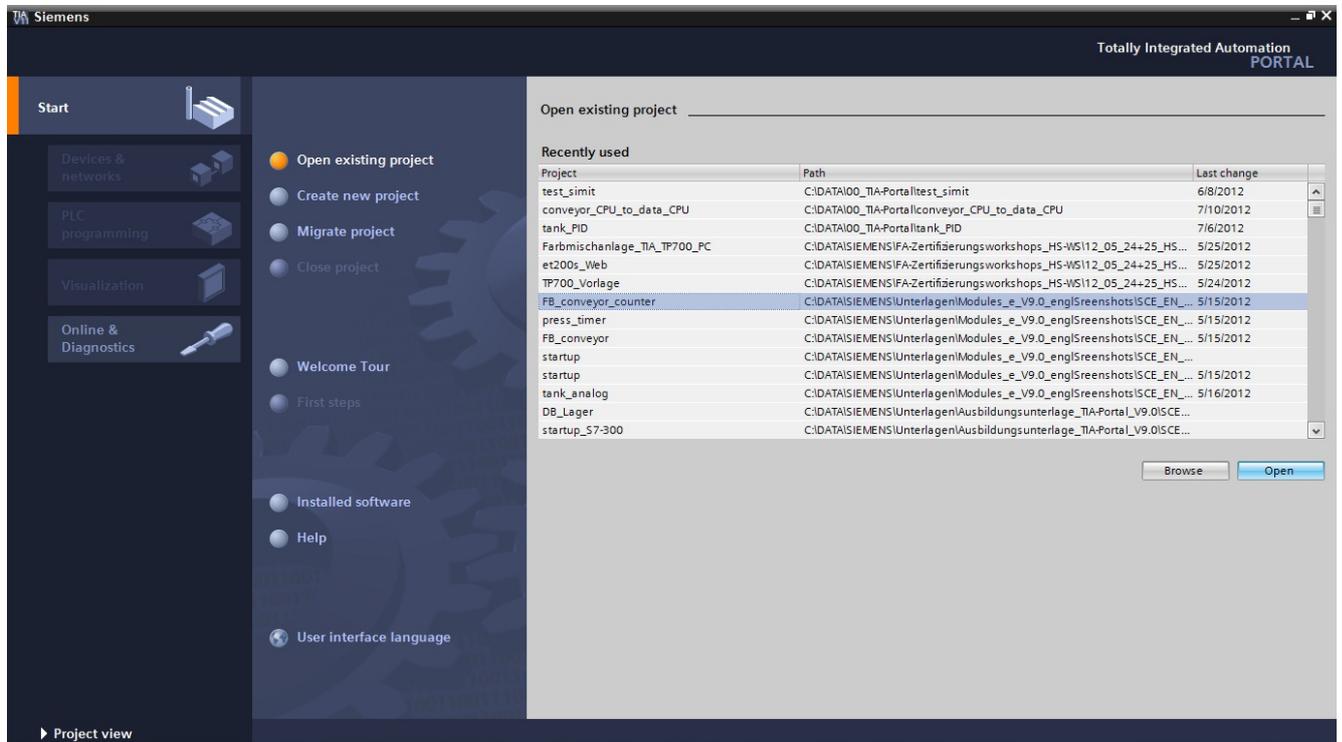
Dans les étapes suivantes, nous allons ouvrir un projet pour le SIMATIC S7-1200, l'enregistrer sous un nouveau nom et le modifier pour qu'il réponde aux nouvelles exigences.

L'outil que nous allons utiliser est **TIA Portal V11**, que l'on appelle ici d'un double-clic.

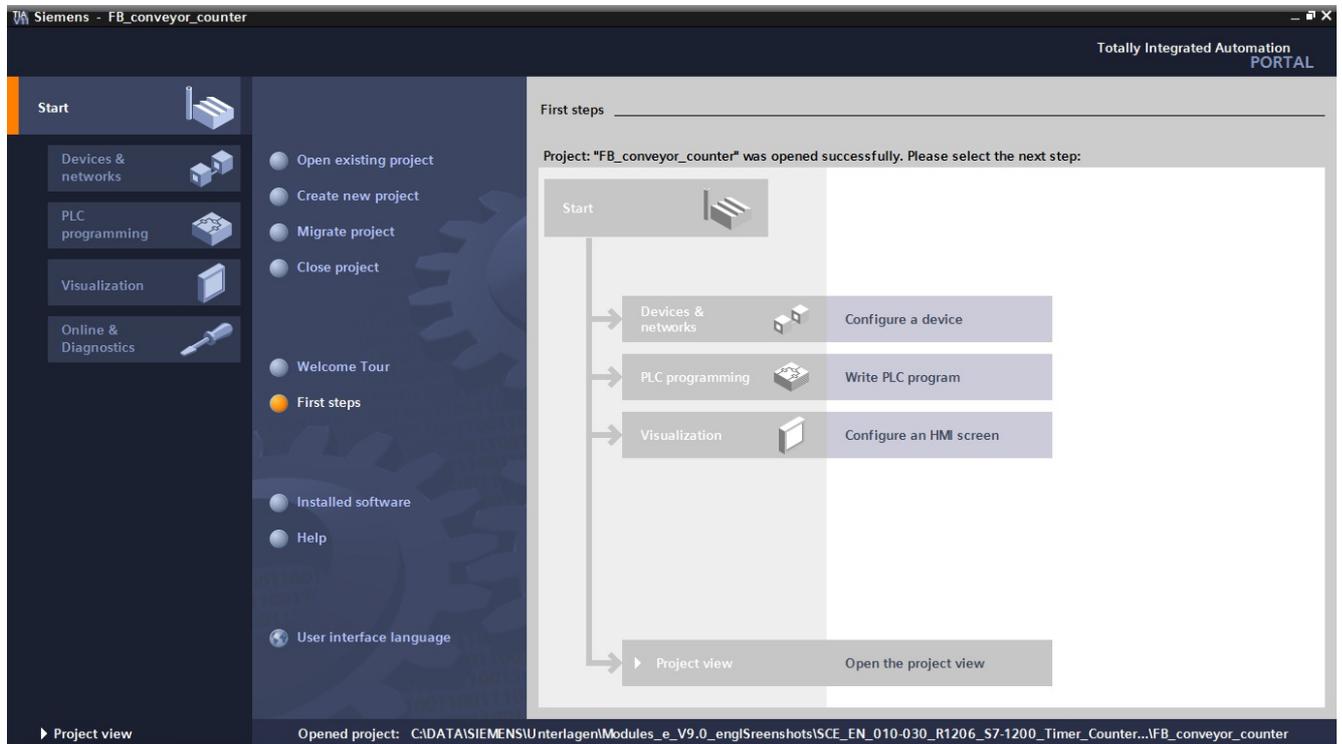


## 4.1 Charger le modèle de projet et enregistrer à nouveau

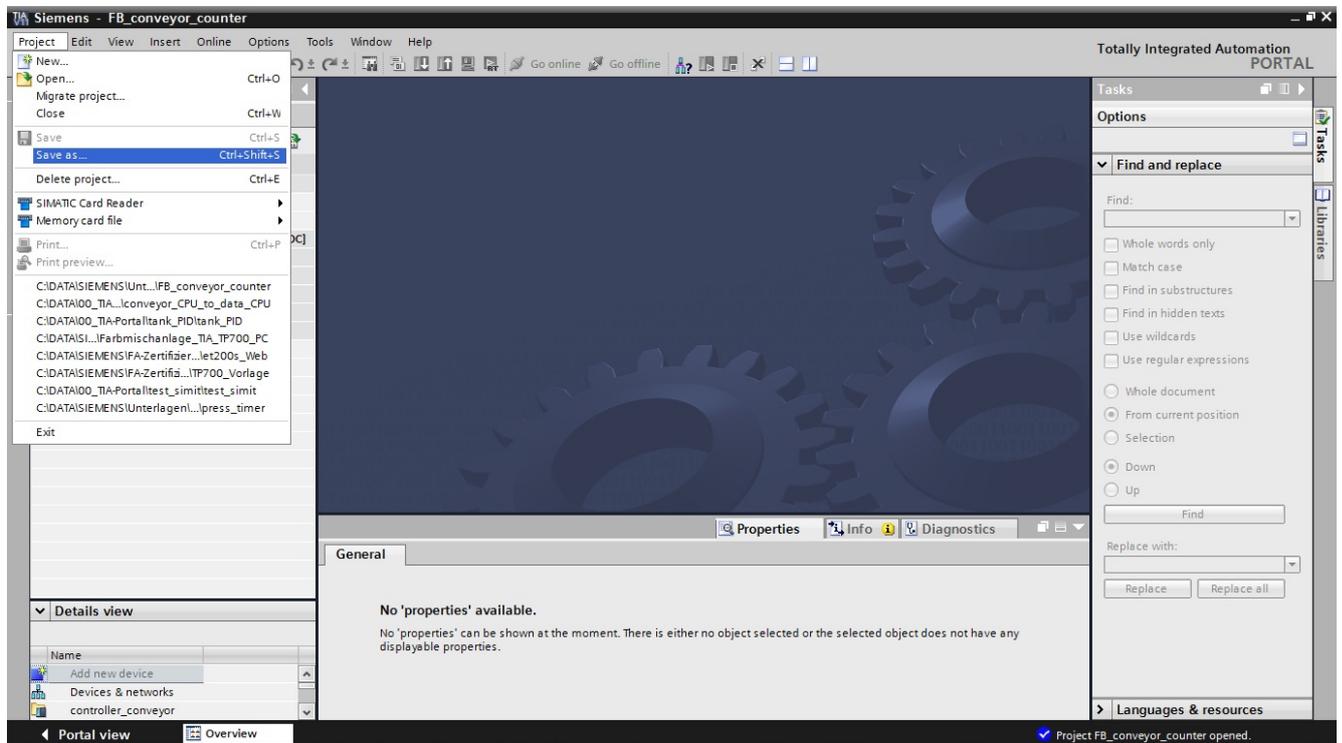
Nous allons maintenant ouvrir le projet **FB\_conveyor\_counter (FB\_convoyeur\_compteur)** du module 010-030 dans la vue du portail. Ce projet servira de base pour le programme.



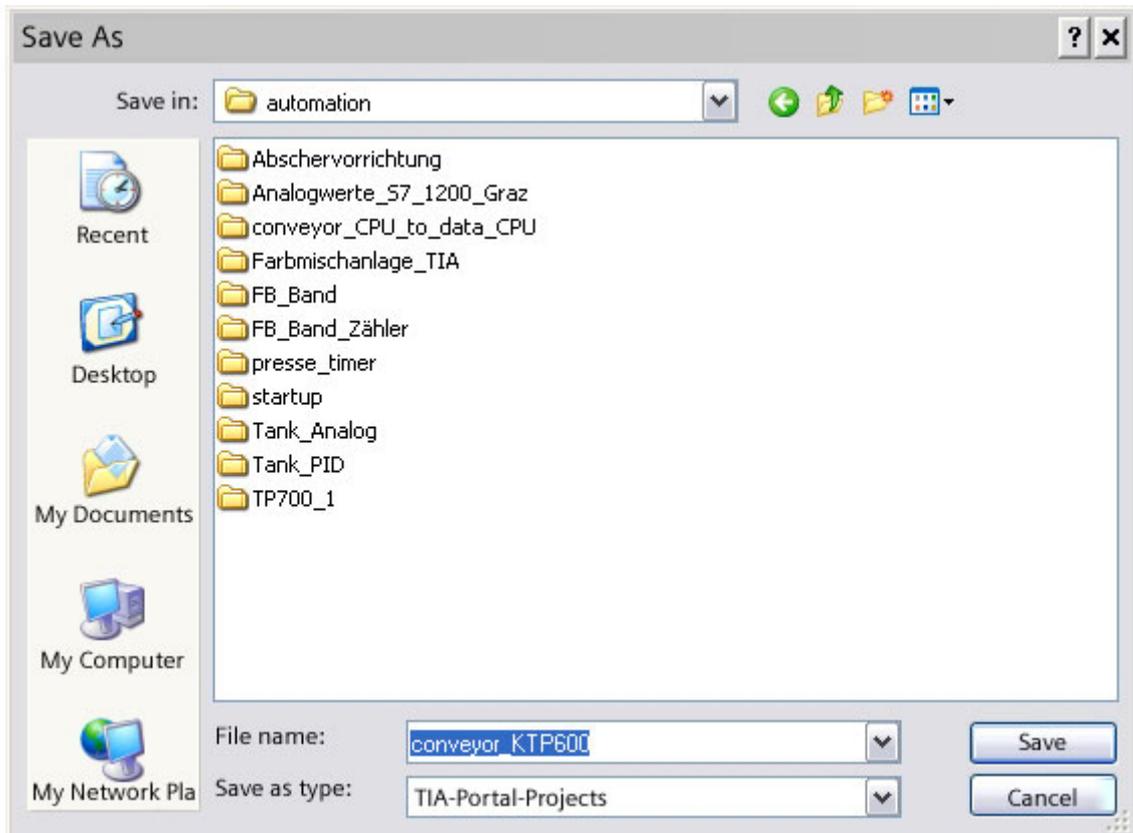
L'option **First steps (Mise en route)** est proposée pour la configuration.  
Cliquer sur **Open the project view (Ouvrir la vue du projet)**.



Le projet doit d'abord être enregistré sous un autre nom.  
Dans le menu **Project (Projet)**, cliquer sur **Save as (Enregistrer sous)**.



**Enregistrer** le projet sous le nom **Conveyor\_KTP600 (Convoyeur\_KTP600)**.

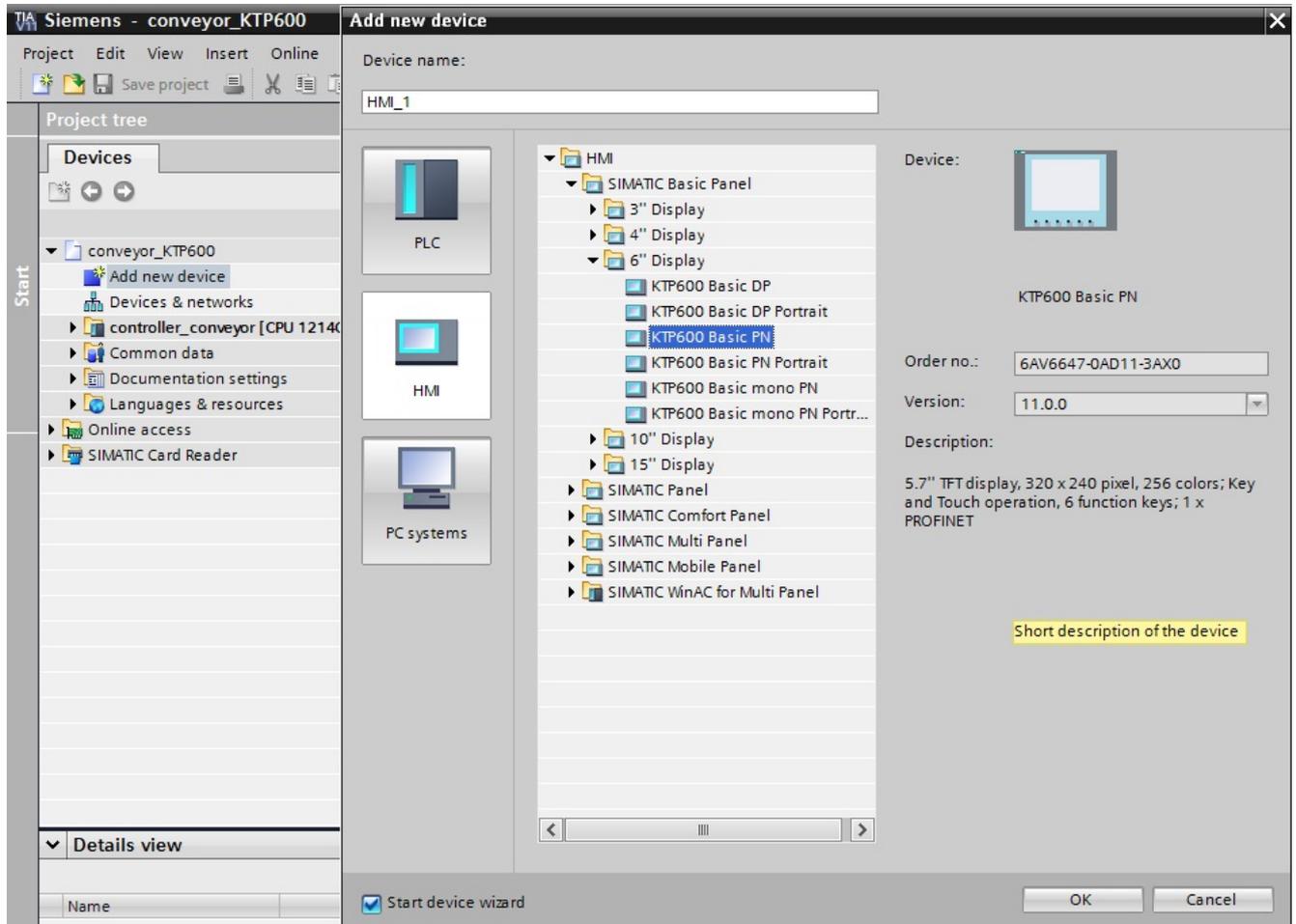


Pour créer un nouveau Panel dans le projet, ouvrir la fenêtre de sélection en double-cliquant sur **Add new device (Ajouter un appareil)**.

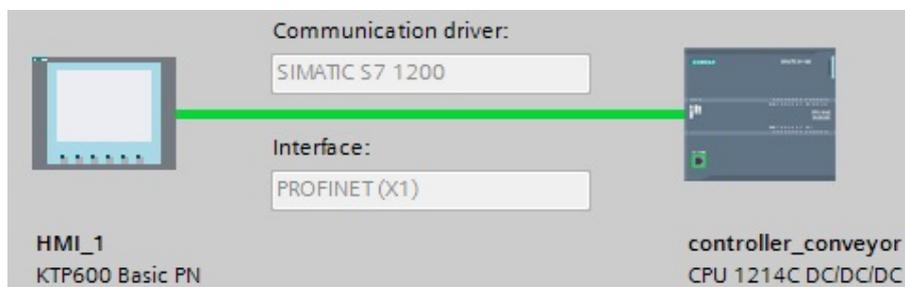
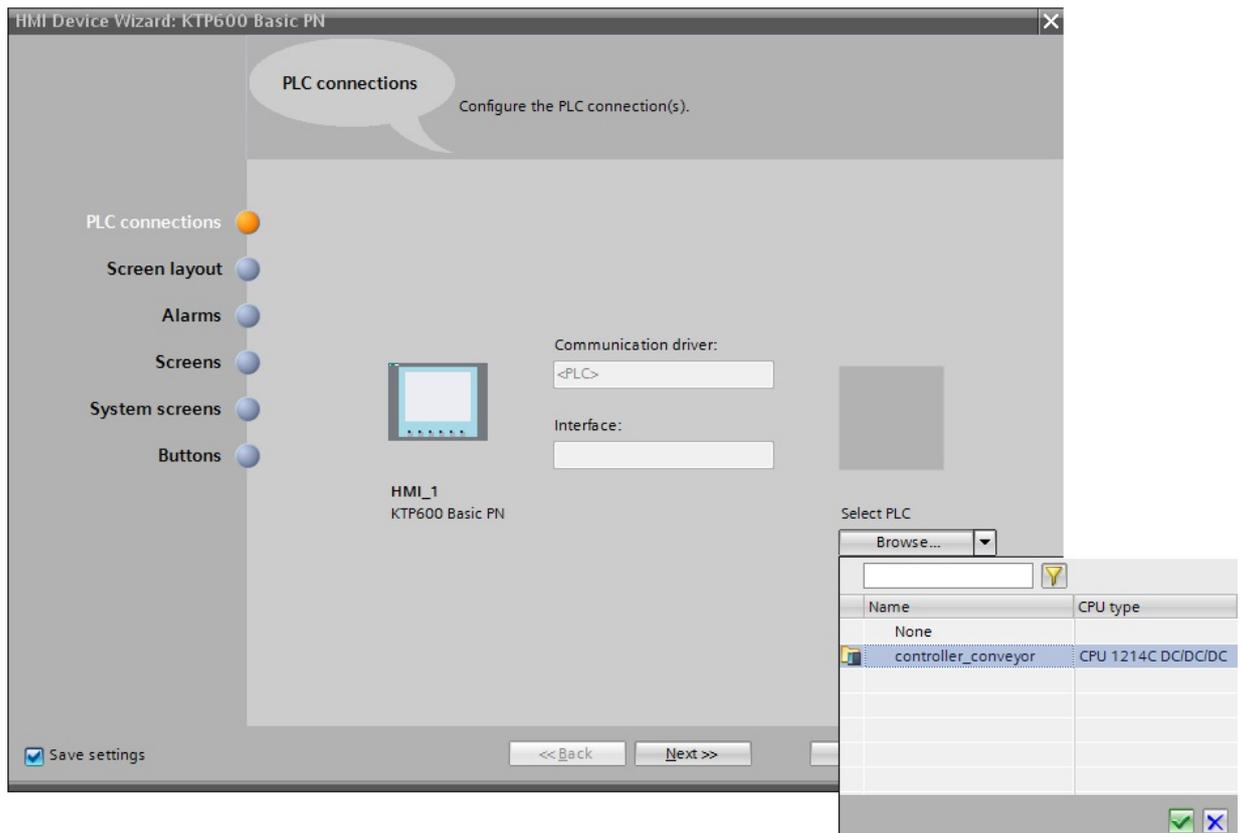
Sous SIMATIC HMI, 6" Display, choisir **KTP600 Basic PN**.

Cocher la case **Open device view (Ouvrir la vue des appareils)**.

Cliquez sur le bouton **OK**.

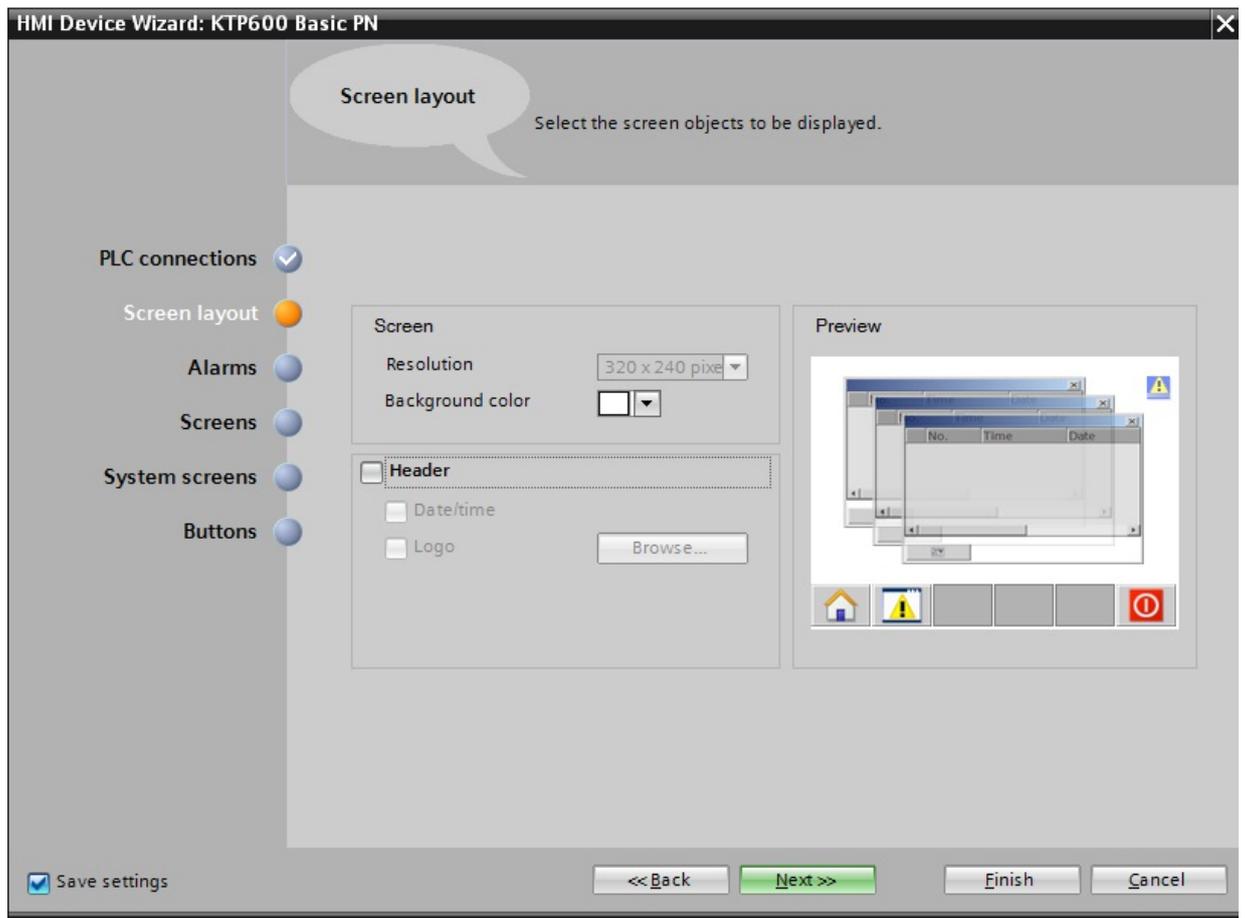


Sous Select PLC (sélectionner API), choisir **controller\_conveyor (commande\_convoyeur)**.



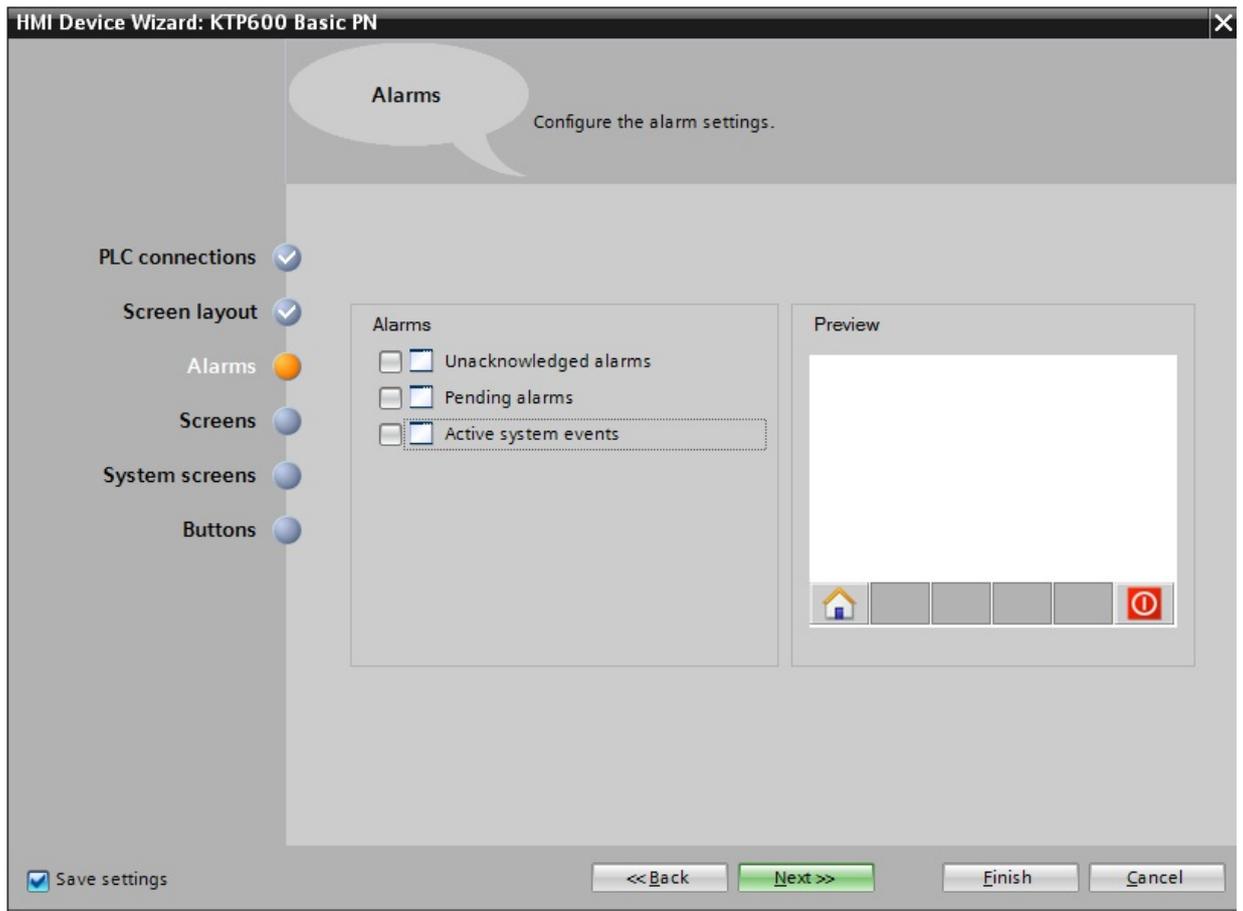
Cliquer ensuite sur le bouton **Next (Suivant)**.

Sous Screen Layout (Présentation écran), modifier la couleur d'arrière-plan et choisir **blanc**, puis désactiver la case à cocher **Header (En-tête)**.



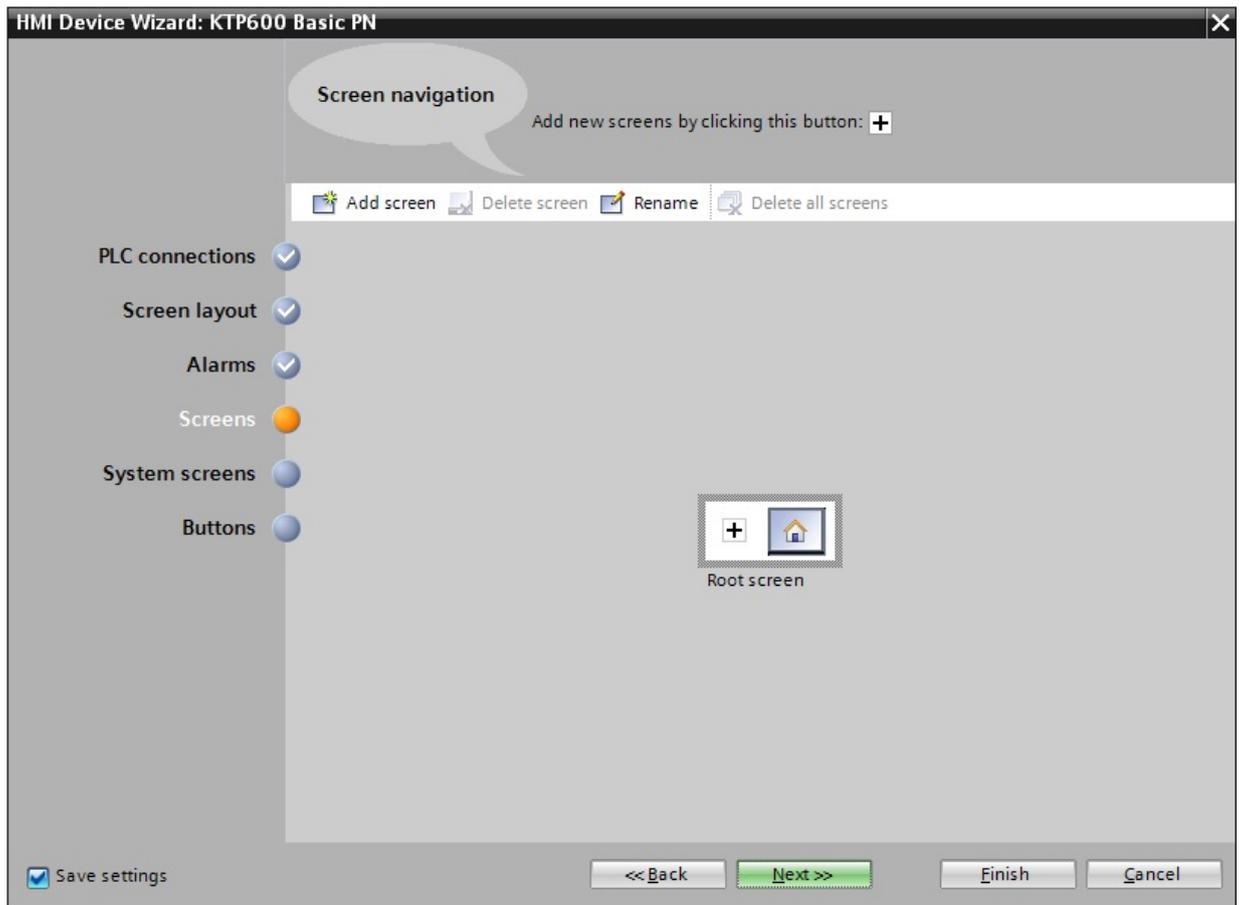
Cliquer ensuite sur le bouton **Next (Suivant)**.

Désactiver toutes les cases à cocher sous Alarms (Alarmes).



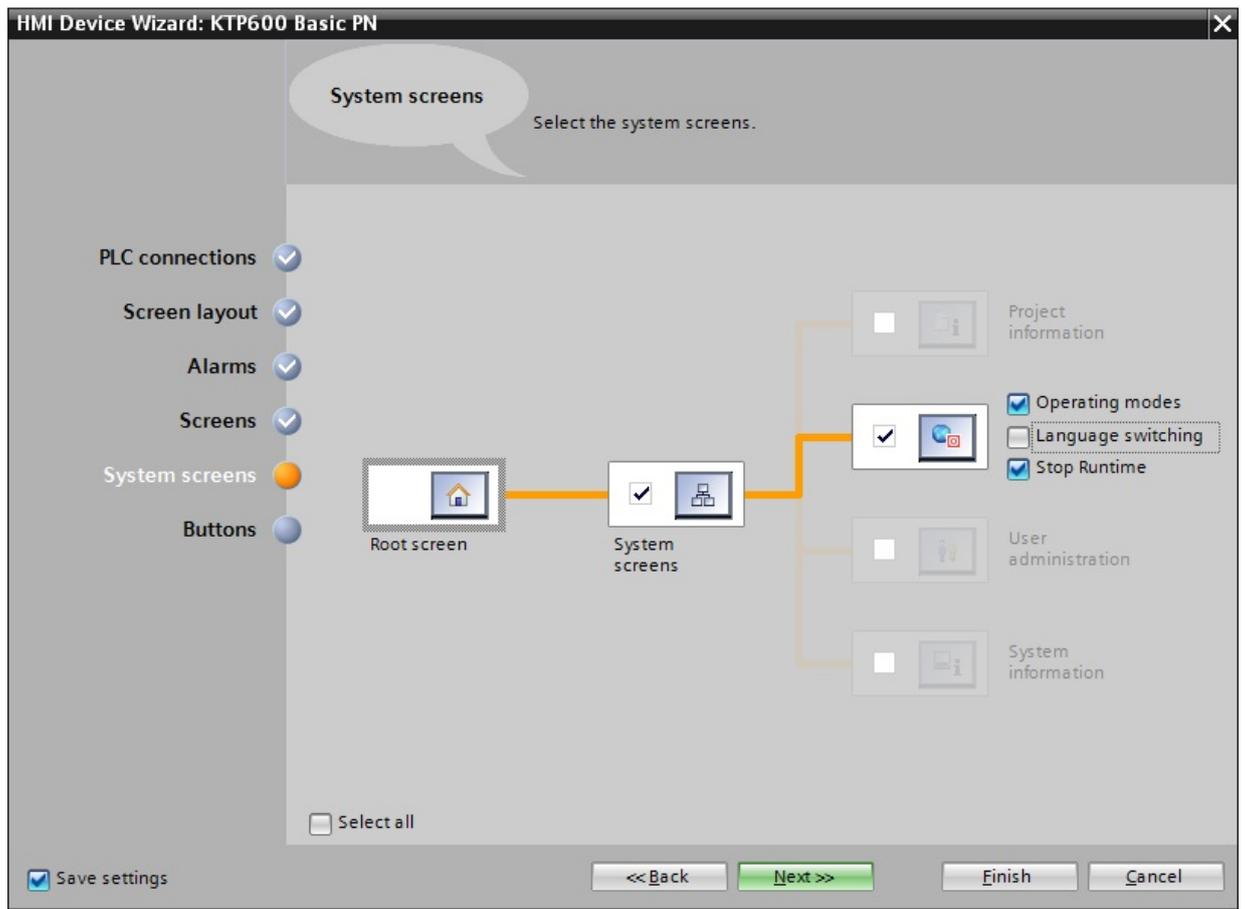
Cliquer ensuite sur le bouton **Next (Suivant)**.

Sous Screen navigation (Navigation écran), il est possible de créer une arborescence de menus. Pour notre exemple, la vue "Root Screen" (Racine) est suffisante.



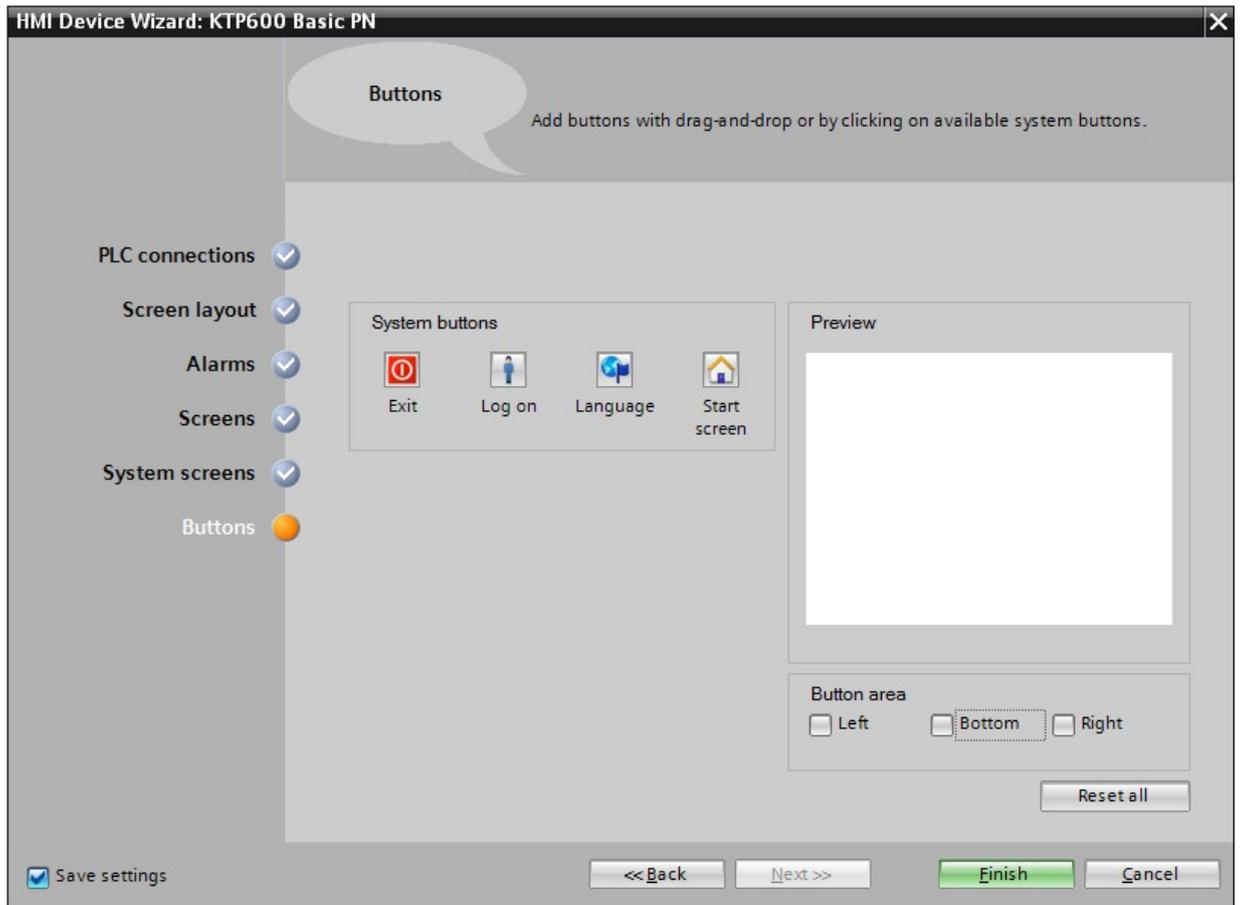
Cliquer ensuite sur le bouton **Next (Suivant)**.

Sous System screens (écrans système) activer **Operating modes (Modes opératoires)** et **Stop Runtime (Arrêter Runtime)**.



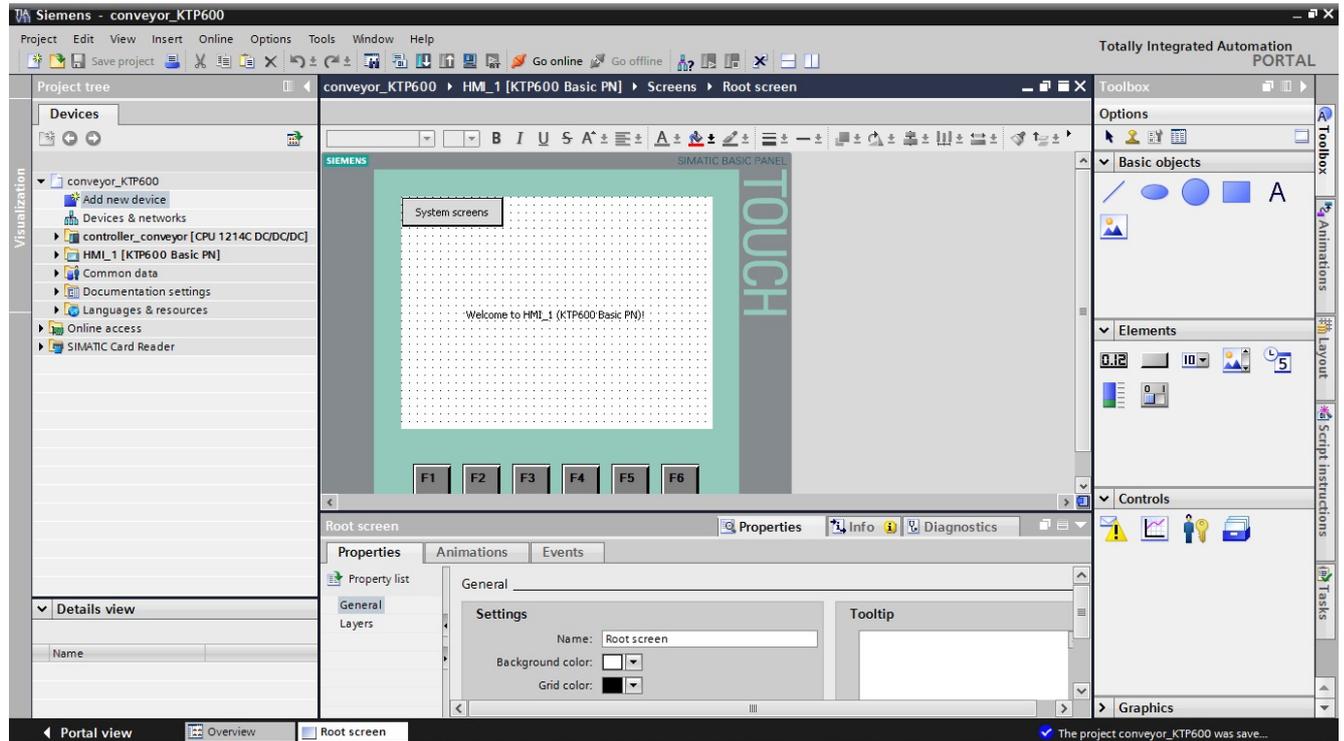
Cliquer ensuite sur le bouton **Next (Suivant)**.

Enfin, il est possible de placer des touches systèmes préconfigurées.  
Désactiver toutes les cases.

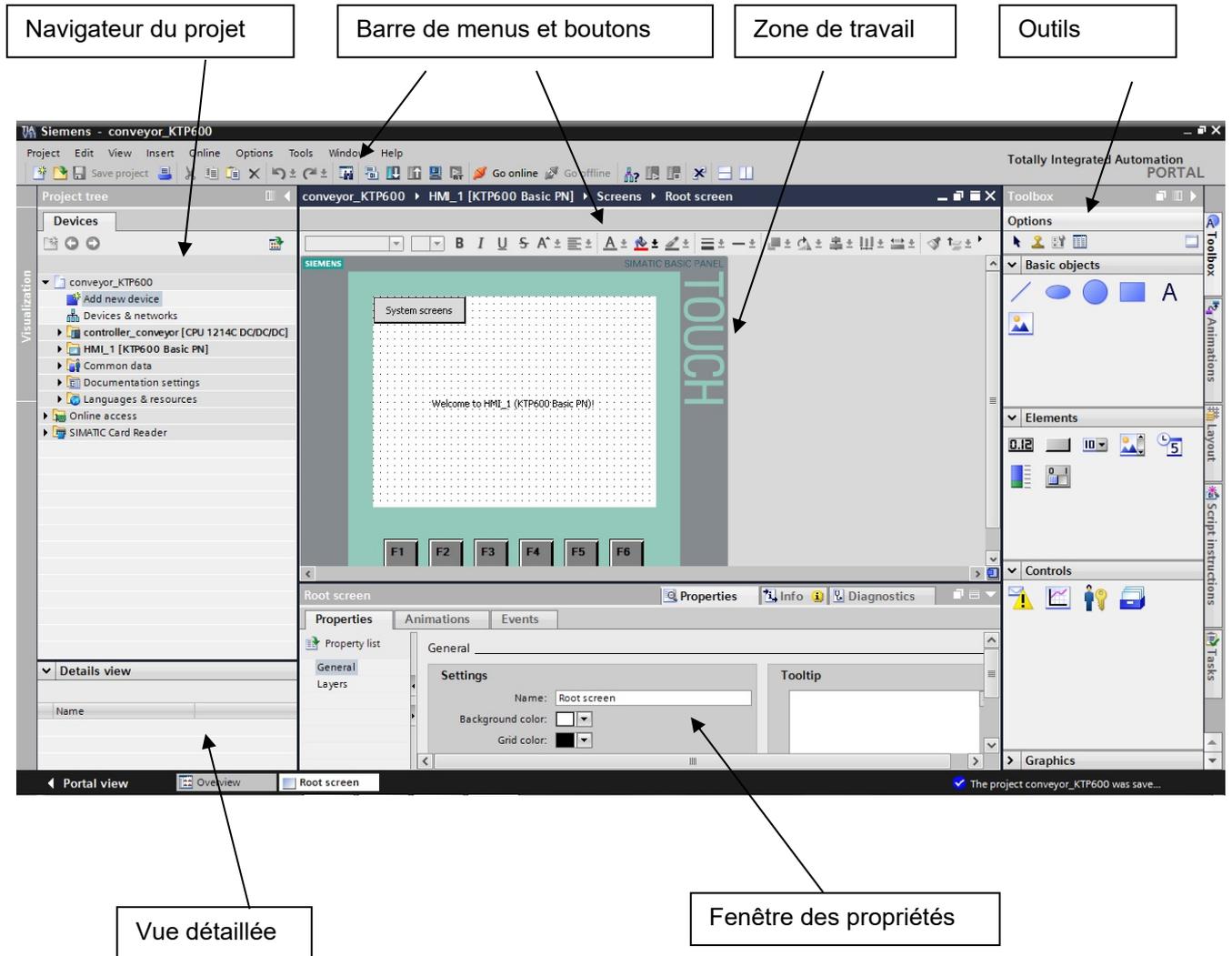


Cliquer ensuite sur le bouton **Finish (Terminer)**.

L'interface de WinCC est maintenant ouverte avec la vue racine.



## 5. Interface utilisateur de WinCC



## 5.1 Navigateur du projet

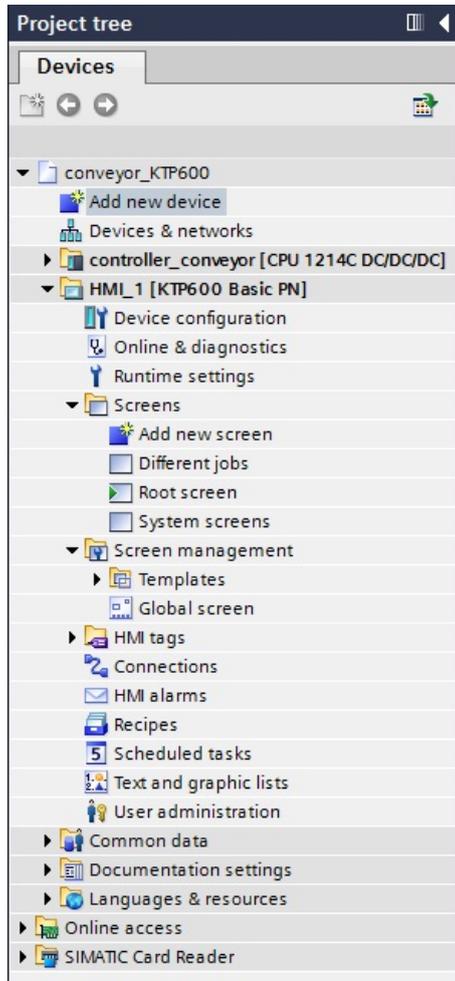
L'arborescence du projet (Project tree) est l'élément central pour le traitement du projet.

Tous les éléments constitutifs et tous les éditeurs disponibles du projet sont affichés dans une arborescence et ils peuvent être ouverts à partir de cette fenêtre.

A chaque éditeur est lié un symbole avec lequel vous pouvez identifier les objets qui lui sont associés.

Seuls les éléments que le pupitre opérateur prend en charge sont affichés dans la fenêtre du projet.

Dans cette fenêtre, les paramètres des différents appareils du pupitre opérateur sont consultables.



## 5.2 Barre de menus et boutons

Toutes les fonctions dont vous avez besoin pour configurer le pupitre opérateur se trouvent dans les menus et les barres d'outils. Si un éditeur est activé, un menu de commandes pour l'éditeur correspondant et une barre d'outils apparaissent.

Passer la souris sur une commande pour obtenir une info-bulle sur chaque fonction.



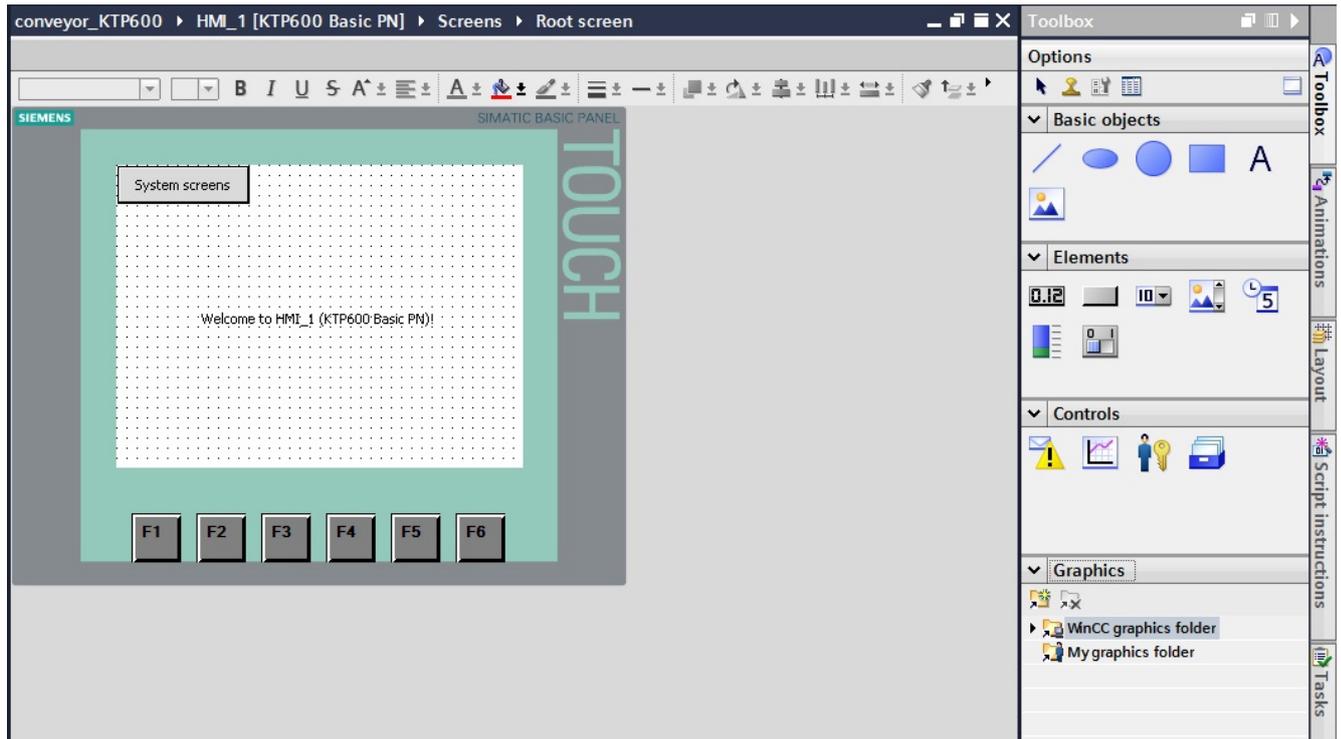
### 5.3 Zone de travail

La zone de travail sert à éditer les objets du projet. Tous les éléments de WinCC sont arrangés autour de cette zone.

On y édite les données du projet soit sous la forme de tables (par exemple, variables), soit de manière graphique (par exemple, vue de process).

La partie supérieure de la zone de travail contient une barre d'outils.

Ici, on trouve par exemple la police, la couleur de la police ou des fonctions comme Rotation, Aligner.etc...



### 5.4 Outils

La fenêtre des outils fournit une liste d'objets que vous pouvez insérer dans vos écrans, par exemple des objets graphiques ou des éléments de commande. Elle comprend également des bibliothèques contenant des objets et un ensemble de blocs d'affichage.

Les objets sont glissés et déposés dans la zone de travail.

### 5.5 Fenêtre des propriétés

Les propriétés des objets sont éditées dans la fenêtre des propriétés, par exemple la couleur des objets de vue.

Cette fenêtre n'est disponible que pour certains éditeurs.

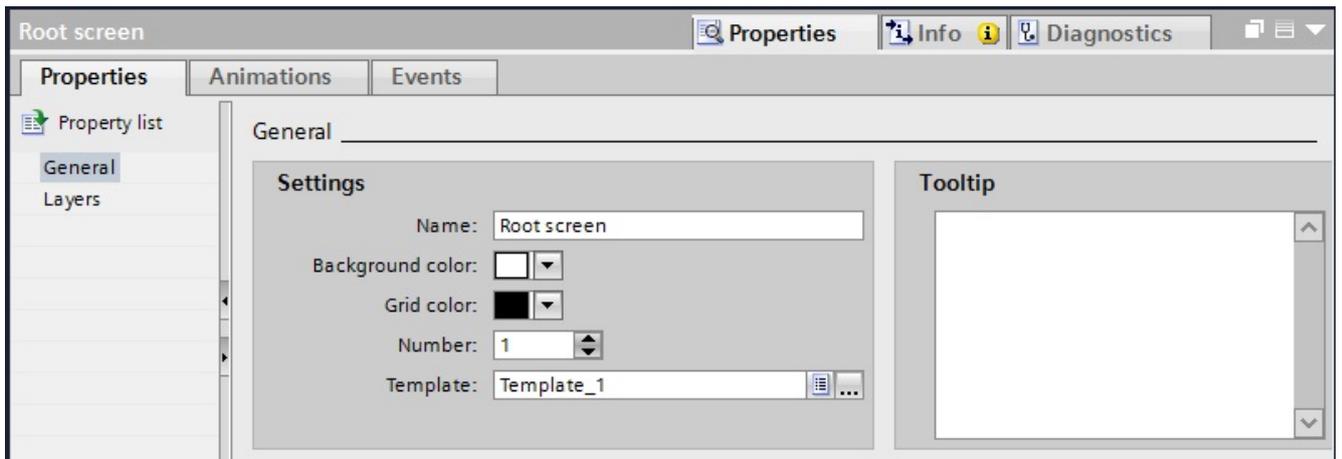
La fenêtre des propriétés affiche les propriétés de l'objet sélectionné classées par catégories.

Aussitôt que vous quittez une zone de saisie, les modifications de valeurs effectuées sont actives.

Si vous entrez une valeur incorrecte, le champ devient coloré.

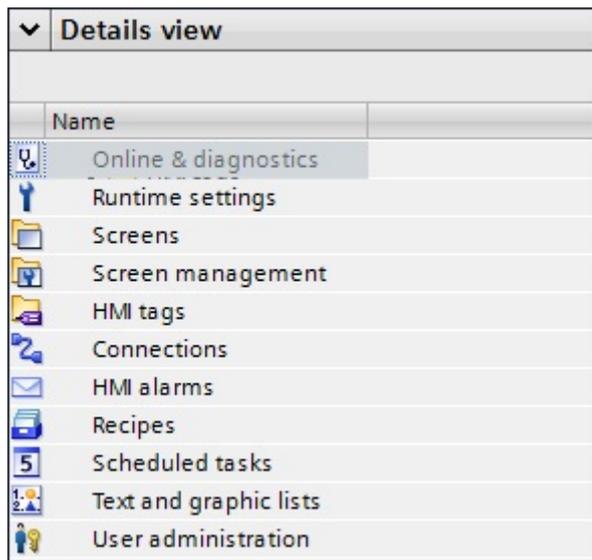
Une info-bulle indique p.ex. l'intervalle de valeurs admissibles.

Dans cette fenêtre, des animations et des événements sont également configurables pour l'objet sélectionné. par exemple, passer à un autre écran en relâchant le bouton.



### 5.6 Vue détaillée

La vue détaillée affiche des renseignements supplémentaires sur l'objet sélectionné dans le navigateur.



## 6. Vues de commande et connexions

Une vue peut se composer d'éléments statiques et dynamiques.

Les éléments statiques, tels que les textes et les graphiques, ne sont pas mis à jour par l'automate. Les éléments dynamiques sont connectés à l'automate et visualisent les valeurs en cours qui se trouvent en mémoire. La visualisation peut être sous la forme d'affichages alphanumériques, de courbes et de barres. Les données saisies sur le pupitre opérateur et écrites dans la mémoire de l'automate sont aussi des éléments dynamiques. Elles communiquent avec l'automate grâce aux **variables (tags)**.

Tout d'abord, nous allons créer une seule pour la commande du convoyeur.

### 6.1 Vue racine ou vue initiale

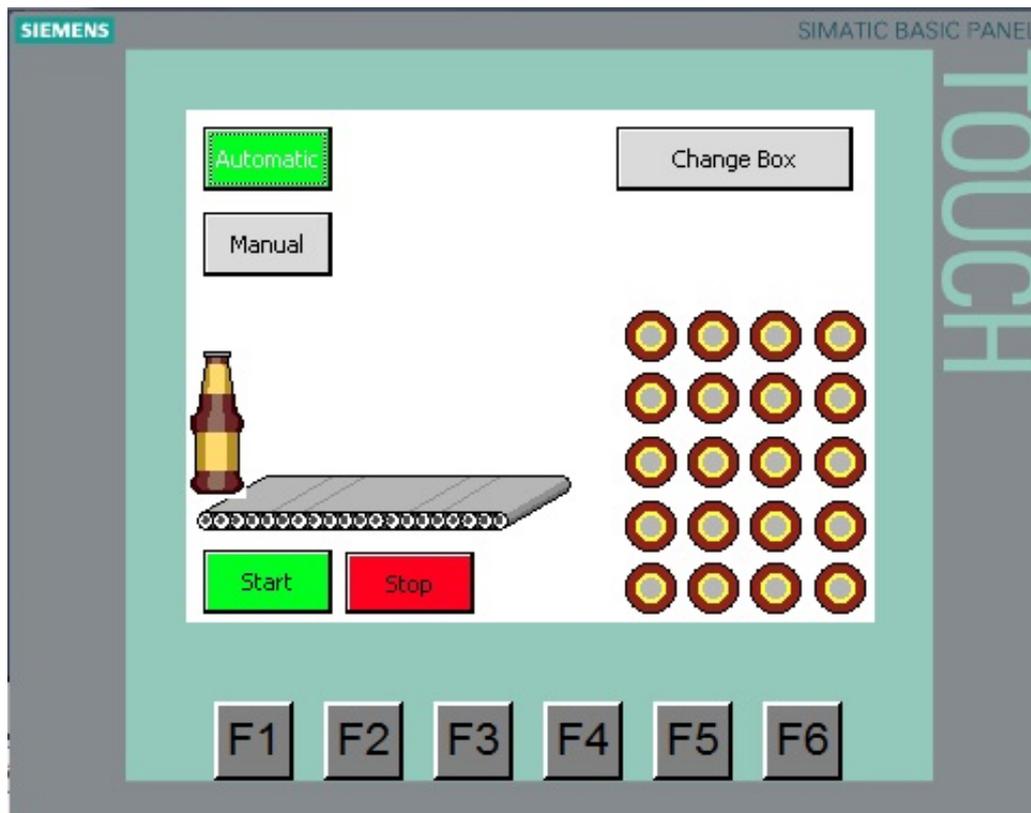
Cette vue a été créée automatiquement et a été définie comme vue initiale.

Ici, l'installation est représentée entièrement.

Les fonctions de changement de mode entre automatique et manuel, marche-arrêt du moteur du convoyeur et remplacement de la caisse peuvent être effectuées via les boutons.

Le mouvement de la bouteille sur le tapis roulant et le niveau de remplissage de la caisse sont représentés graphiquement.

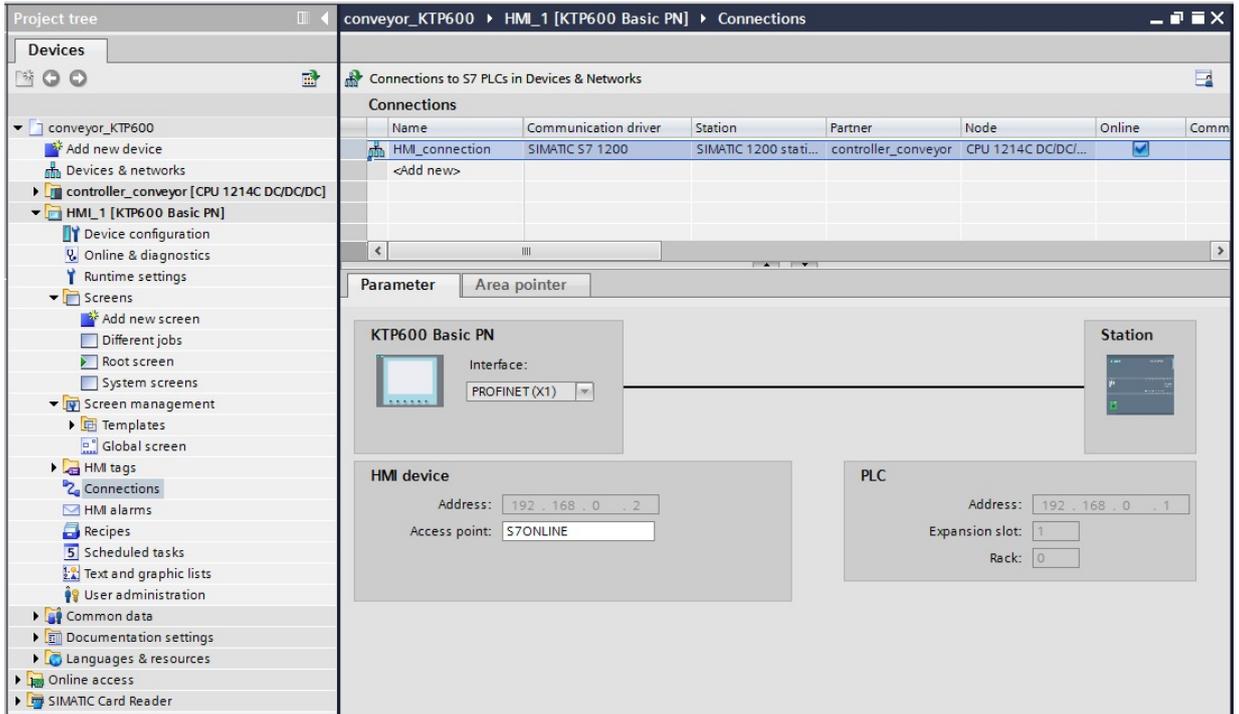
En utilisant la touche F6, on passe à la vue système.



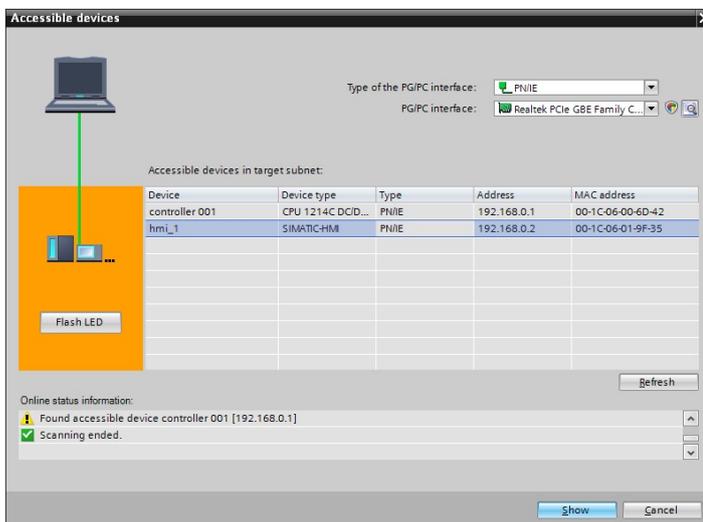
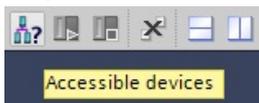
## 6.2 Connexions aux automates S7

Pour pouvoir assurer l'affichage et la commande des valeurs de process d'un automate, il faut en premier lieu configurer une liaison vers celui-ci.  
 Dans cette partie, nous allons expliquer comment le Panel communique avec l'automate et avec quelle interface.

Dans le navigateur du projet, double-cliquer sur **Connections (Connexions)**.  
 Grâce aux paramètres de configuration du matériel, tous les paramètres sont déjà réglés.

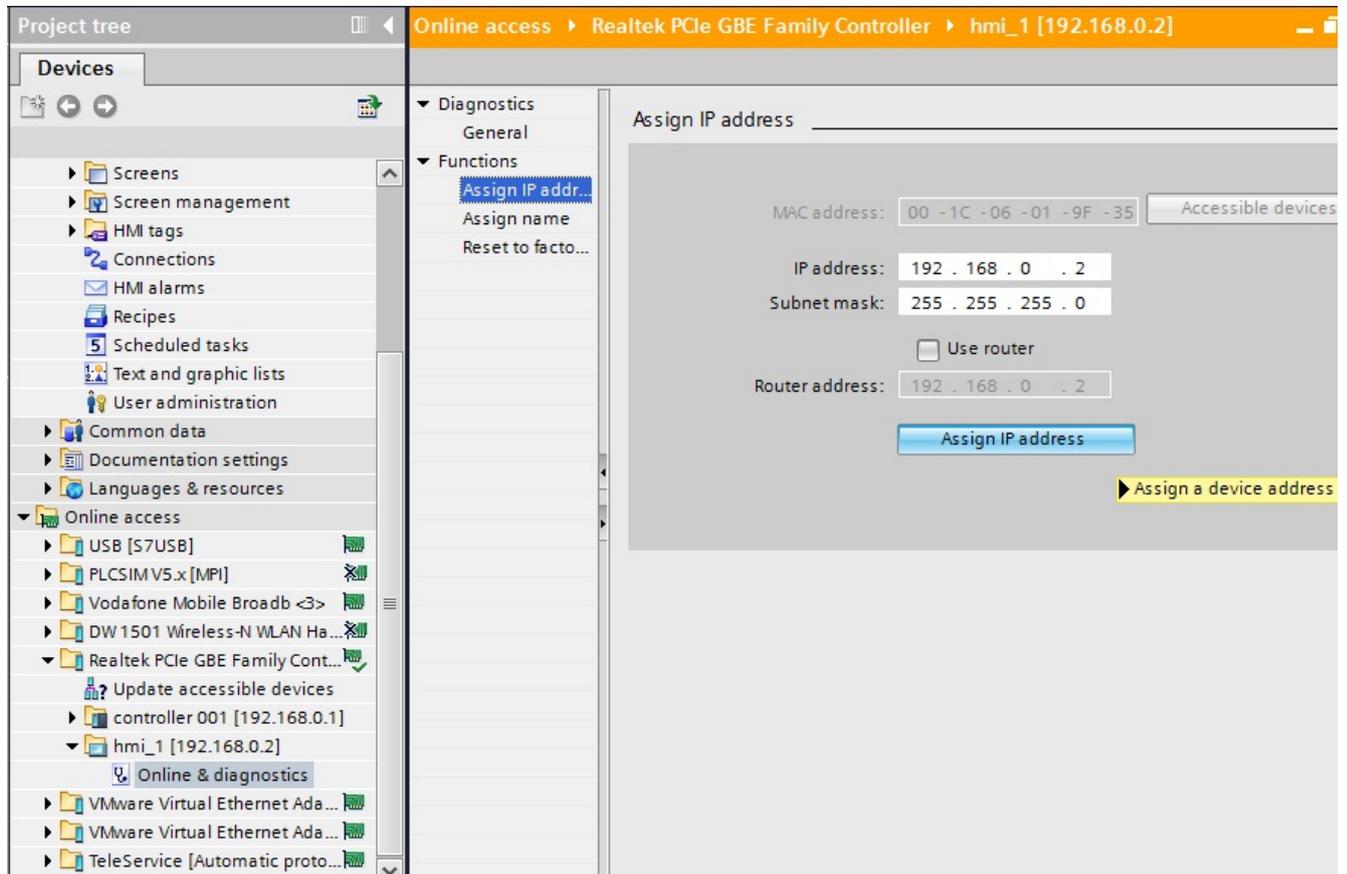


Il ne reste plus qu'à attribuer au Panel une adresse IP.  
 Lire l'adresse MAC du Panel via **Accessible devices (Abonnés accessibles)**  
 Cliquer sur le bouton **Display (Afficher)**.



### 6.3 Affecter l'adresse IP

Après avoir entré l'adresse MAC, l'adresse IP peut être affectée sous **Online and diagnostics (En ligne et Diagnostics)**. Pour ce faire, le Panel doit obligatoirement être en mode Transfert.



#### Remarque

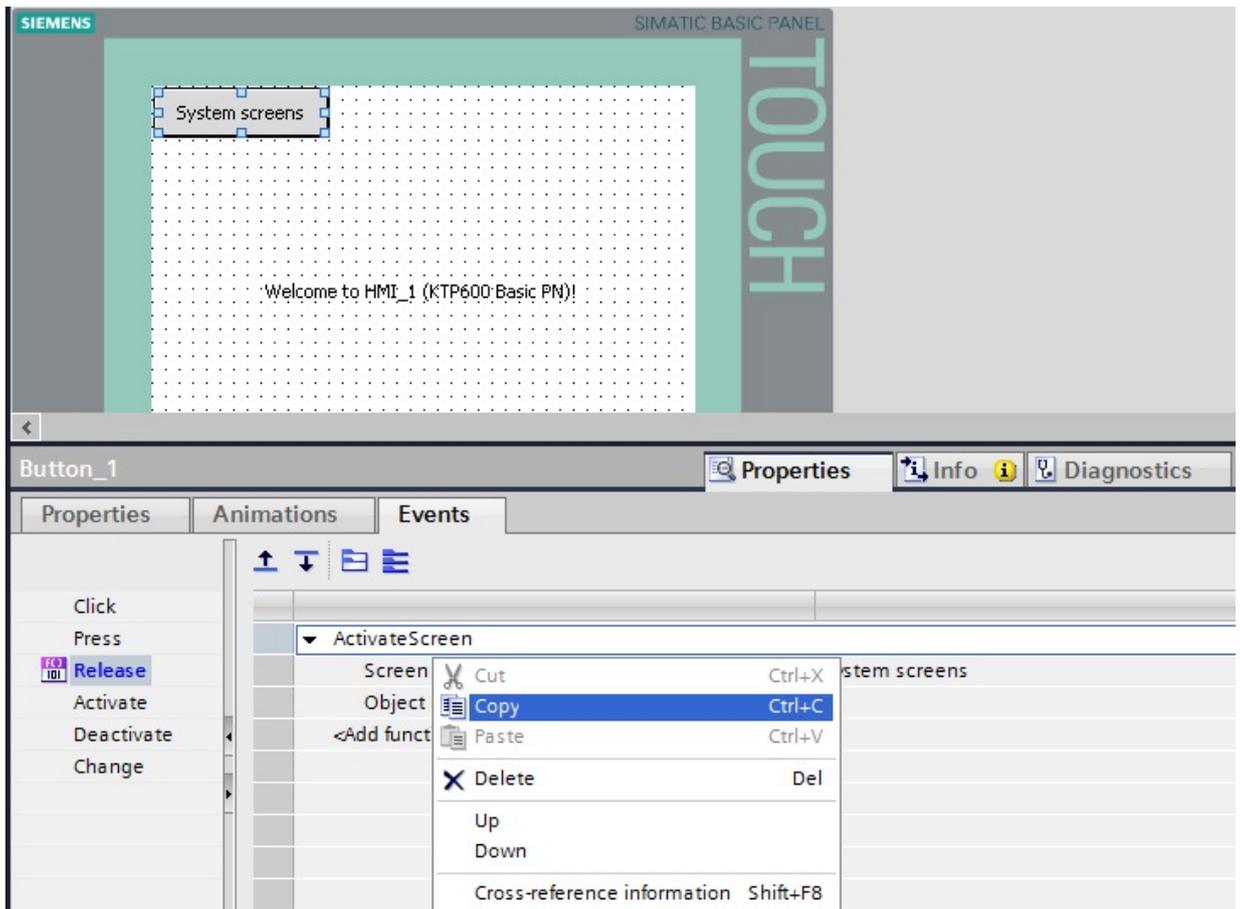
L'adresse IP peut également être vérifiée ou entrée directement sur le Panel, sous **Control Panel, Profinet**.

## 7 Configuration de la vue racine

En cliquant sur le bouton "System screens (Vues système)", la vue système apparaît.

On veut transférer la fonction du bouton **System screens (Vues système)** vers la touche de fonction **F6**.

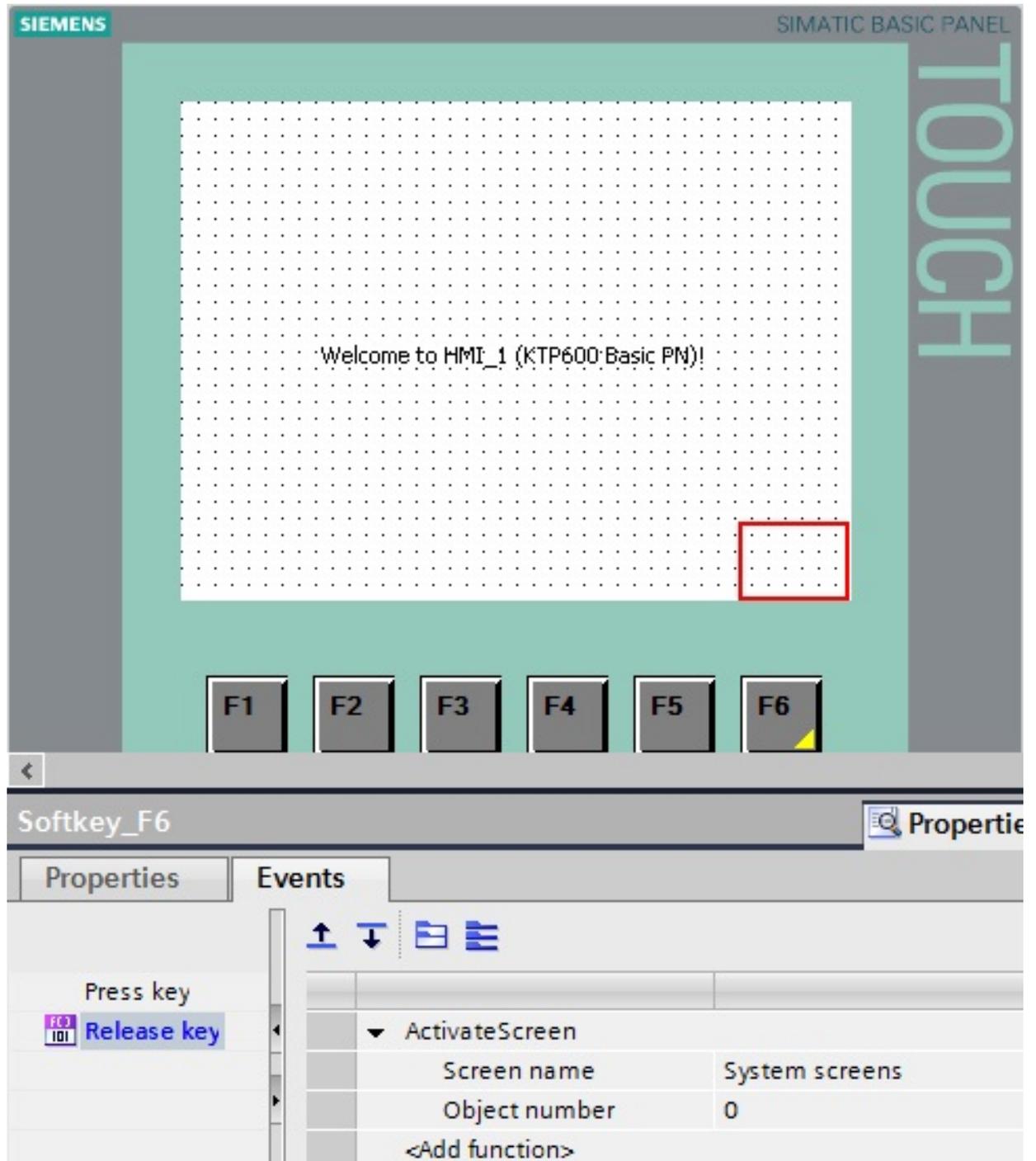
Sélectionner le bouton **System screens (Vues système)** et en-dessous, dans la fenêtre des propriétés, copier la fonction **ActivateScreen (Activer vue)** sous **Events (Événements), Release (Relâcher)**.



## 7.1 Touche de fonction F6

Sélectionner la touche de fonction „F6 et en-dessous, dans la fenêtre des propriétés, coller la fonction **ActivateScreen (Activer vue)** sous **Events (Evénements)**, **Release key (Relâcher bouton)**.

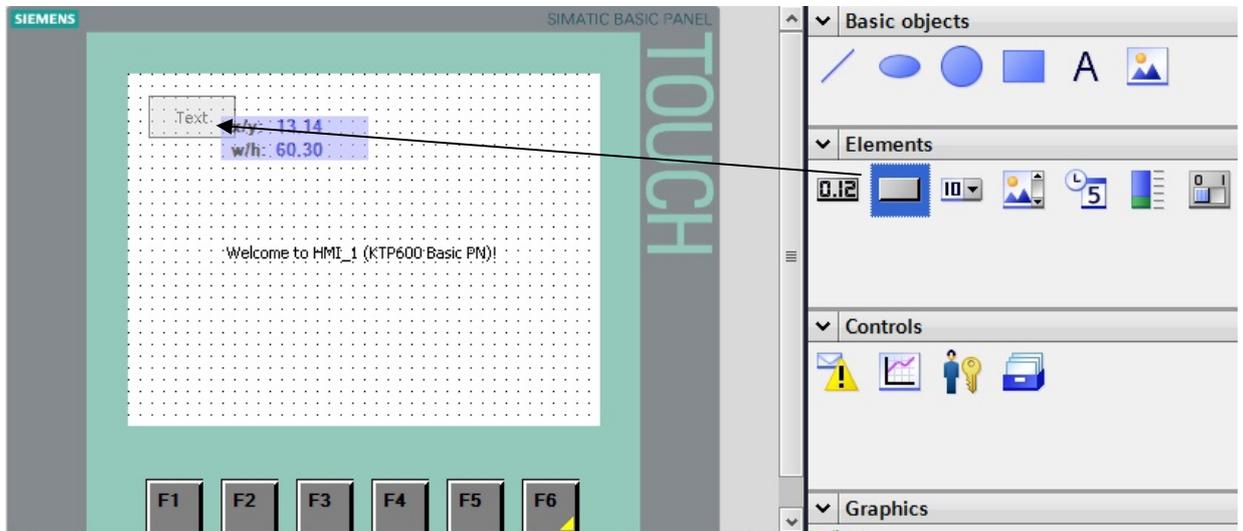
Effacer ensuite la zone de texte au centre et supprimer le bouton "System screens (Vues système)".



Le coin jaune sur la touche F6 indique qu'elle est configurée.

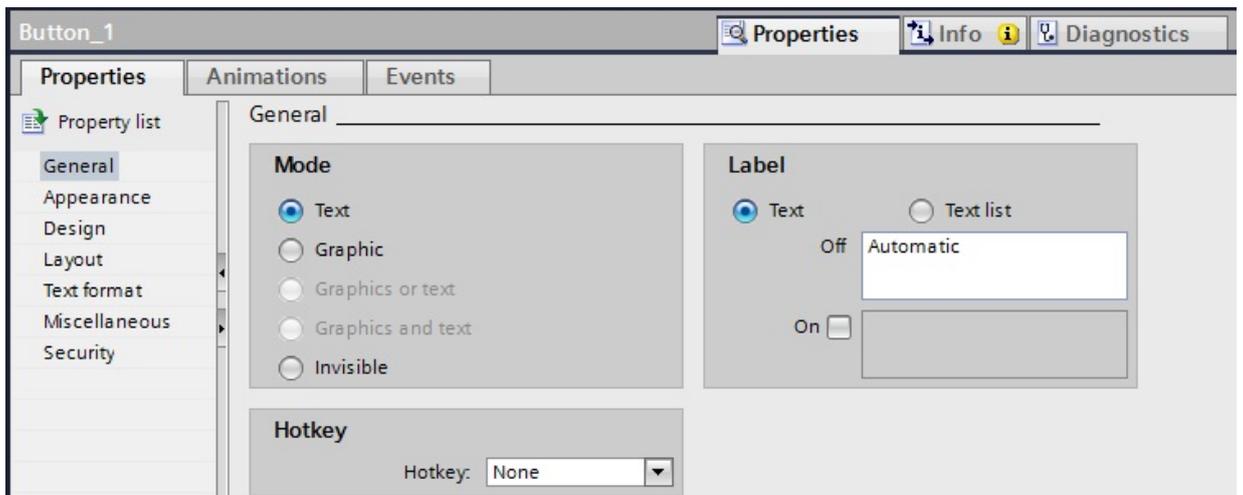
## 7.2 Configuration des boutons Automatique et Manuel

Glisser un bouton dans la zone de travail de la vue racine.

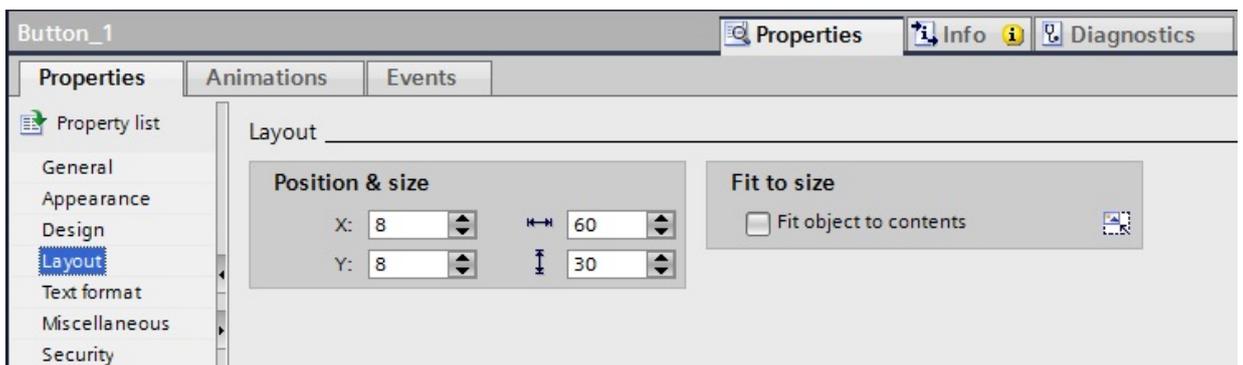


Comme légende, entrer **Automatic**.

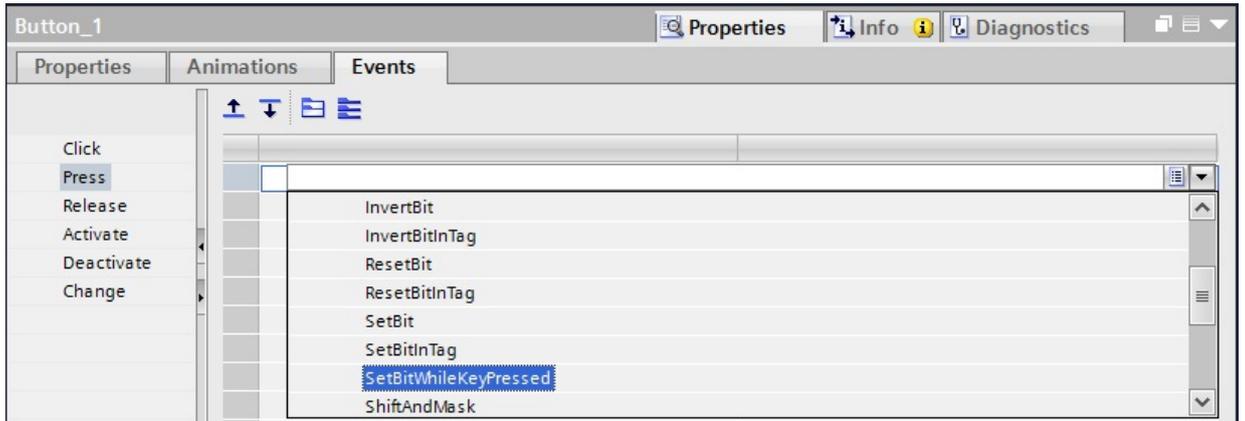
**Attention !** Ne pas appuyer sur la touche Entrée, cela créerait une autre ligne de texte.



Sous Layout (Mise en page), saisir la position et la taille souhaitées.



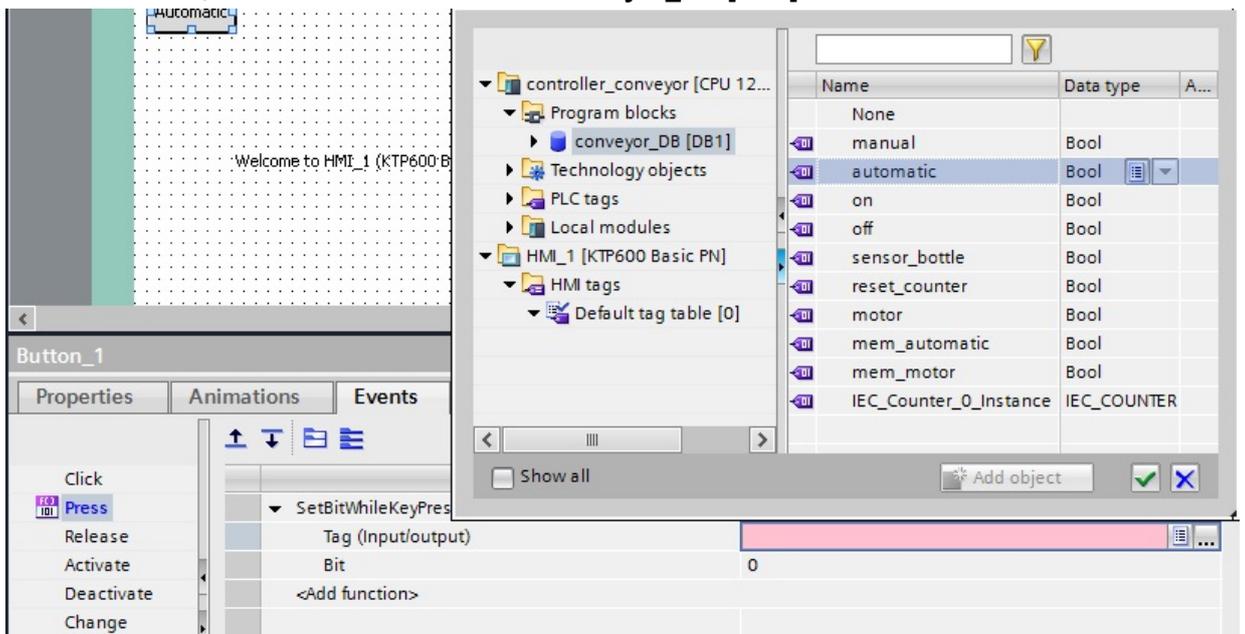
Sous Event (Événements), Press (Appuyer), ajouter la fonction **SetBitWhileKeyPressed** (**PositionnerBitTantQueToucheEnfoncée**).



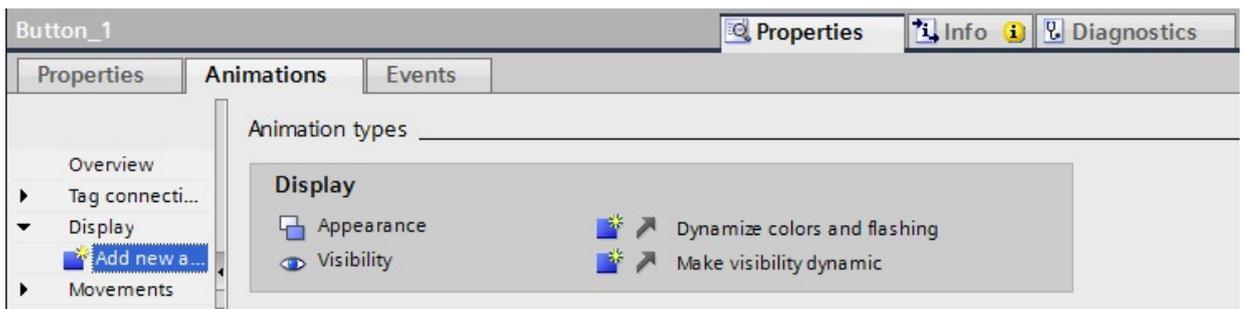
Ensuite, cliquer sur le champ Variable (Entrée/sortie) et ouvrir la fenêtre des variables à l'aide du bouton ...

Vous pouvez également accéder à la déclaration d'interface des blocs de données.

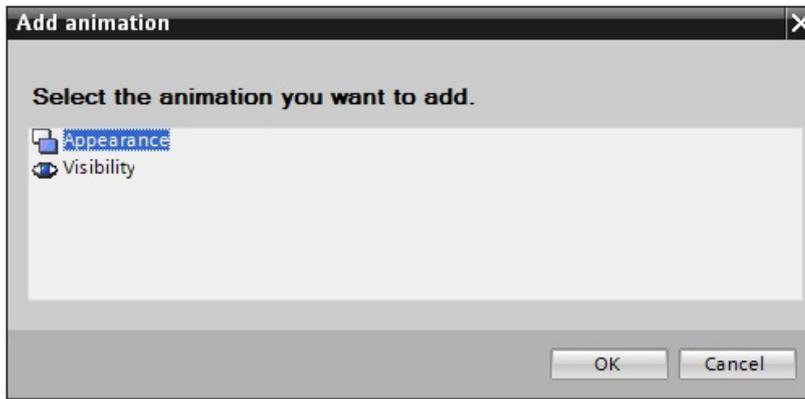
Comme variable, choisissez **automatic** sous **conveyor\_DB [DB1]**.



Maintenant, le bouton doit clignoter et changer de couleur quand le mode automatique est activé. Sous Animations, sélectionner \ Add new... (Nouvelle animation).

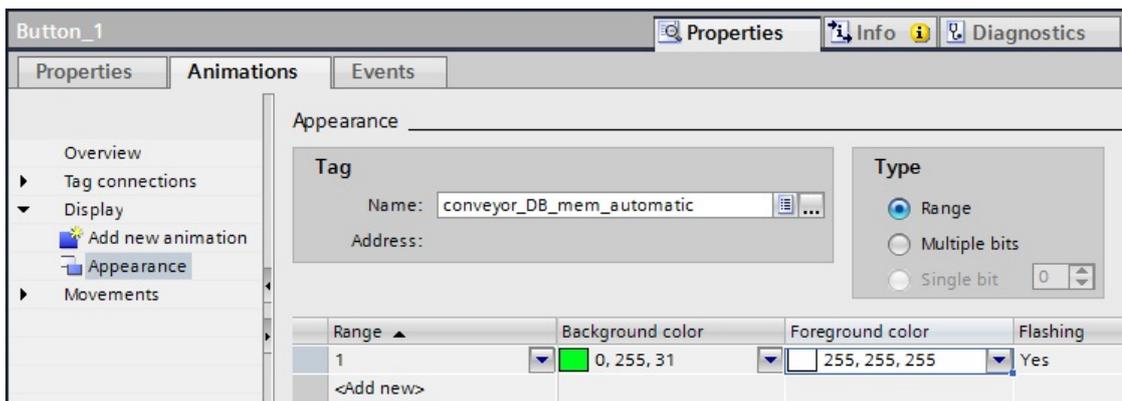


Sélectionner **Appearance (Aspect)** et confirmer par **OK**.



Comme variable, choisissez **mem\_automatic** sous **conveyor\_DB [DB1]**.

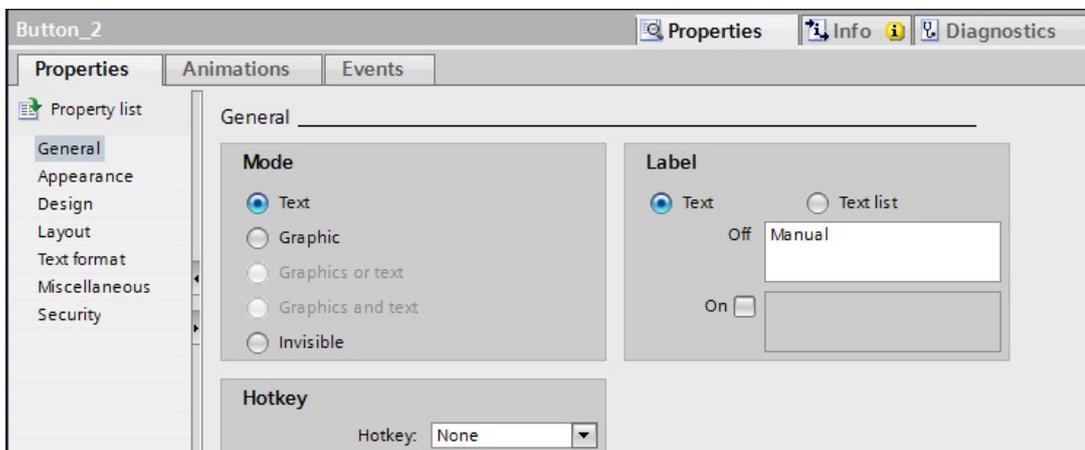
Le bouton doit changer de couleur en mode automatique : c'est-à-dire lorsque la variable **mem\_automatic** est à **1**. Pour que le changement de couleur soit visible, sous **Appearance (Aspect)** changez la couleur d'avant-plan en **blanc** et la couleur d'arrière-plan en **vert**. Mettez également **Yes (Oui)** pour le clignotement.



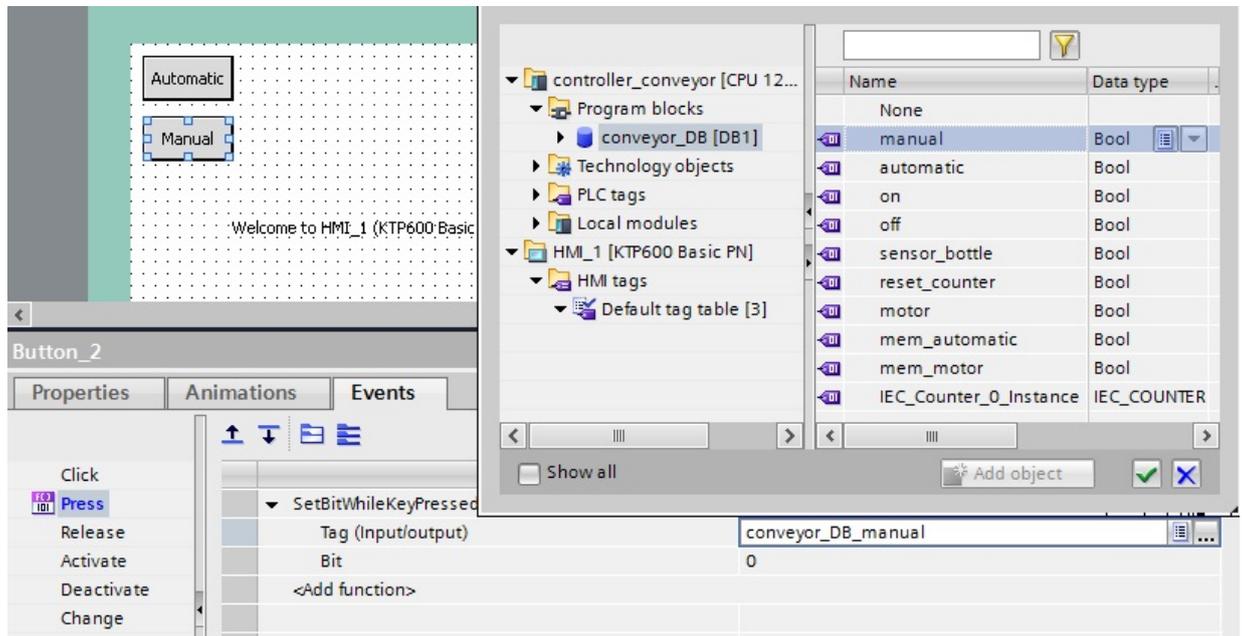
Copier et coller le bouton **Automatic**.  
Placez le bouton ajouté sous le bouton Automatic.

Comme légende, entrer **Manual**.

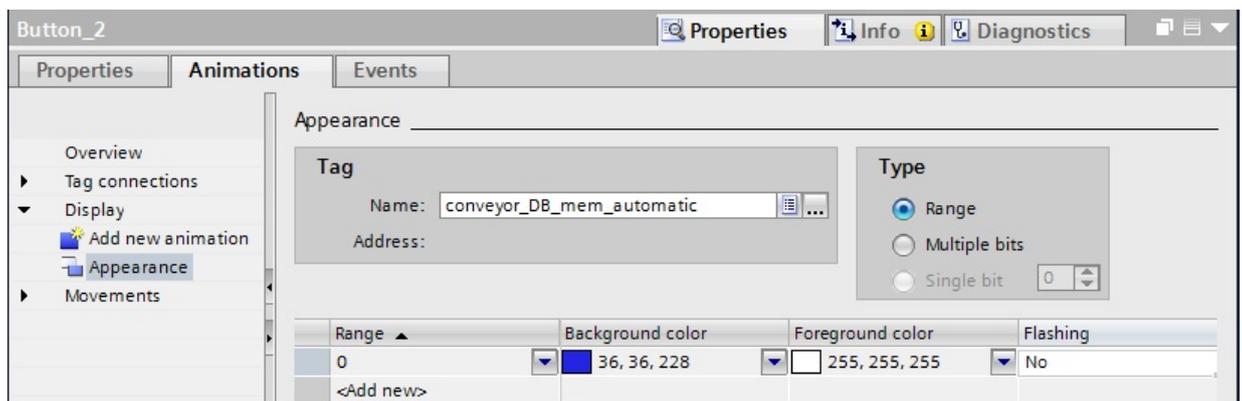
**Attention !** Ne pas appuyer sur la touche Entrée, cela créerait une autre ligne de texte.



Sous Events (Événement), Press (Enfoncer) choisir la variable **manual** sous **conveyor\_DB [DB1]**. La variable doit être sélectionnée car c'est seulement de cette manière que la nouvelle variable IHM sera générée.



Le bouton doit changer de couleur en mode manuel : c'est-à-dire lorsque la variable **mem\_automatic** est à **0**. Pour que le changement de couleur soit visible, sous **Appearance (Aspect)** changez la couleur d'avant-plan en **blanc** et la couleur d'arrière-plan en **bleu**. Mettez également **No (Non)** pour le clignotement.



Enregistrer le projet.

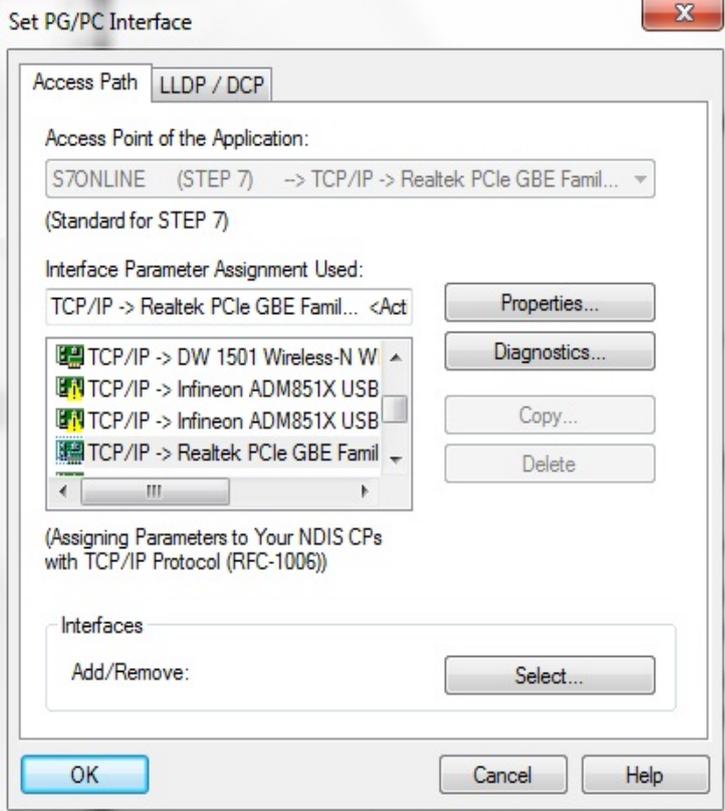
### 7.3 Modifications dans le programme STEP 7

Avant de tester la visualisation, nous devons faire un changement dans le programme STEP 7. Dans OB1, supprimer l'affectation S1\_Conveyor1 et S2\_Conveyor1 lors de l'appel du FB1. C'est indispensable, sinon les signaux du Panel sont écrasés par la mémoire image des entrées. Enregistrer et charger le programme modifié.

"S3_CONVEYOR1"	%Q.2	conveyor1 pushbutton conveyor ON (no conta...
"M1_CONVEYOR1"	%Q.2	conveyor1 motor conveyor belt M1
"S4_CONVEYOR1"	%Q.3	conveyor1 pushbutton conveyor OFF (nc cont...
"B0_CONVEYOR1"	%Q.7	conveyor1 sensor bottle-counter
"S5_CONVEYOR1"	%Q.6	conveyor1 reset counter / new box

### 7.4 Paramétrage de l'interface PG/PC pour la simulation Runtime

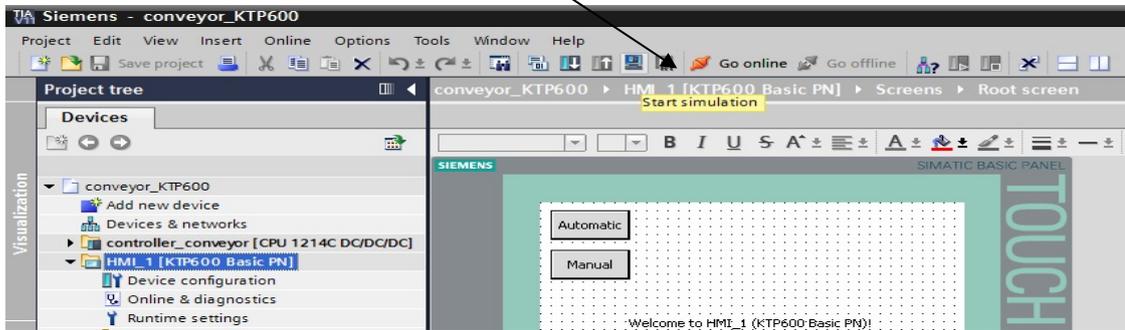
Pour établir la connexion entre la simulation Runtime sur le PG/PC et la CPU S7-1200, nous devons d'abord paramétrer l'interface PG/PC pour TCP/IP.

N°	Marche à suivre :
1	Ouvrir le panneau de configuration <ul style="list-style-type: none"> <li>• via "Démarrer &gt; Panneau de configuration" (menu Démarrer pour un accès simplifié aux programmes sous Windows XP),</li> <li>• ou via "Démarrer &gt; Paramètres &gt; Panneau de configuration" (dans le menu Démarrer classique comme pour les anciennes versions de Windows).</li> </ul>
2	Dans le panneau de configuration, double-cliquer sur l'icône "Paramétrage de l'interface PG/PC". <div style="text-align: center;">  <p>PG/PC-Schnittstelle einstellen</p> </div>
3	Définir les paramètres suivants dans l'onglet "Chemin d'accès" : <ol style="list-style-type: none"> <li>1. Sélectionner "S7ONLINE [STEP 7]" pour le point d'accès de l'application dans la liste déroulante.</li> <li>2. Sélectionner dans la liste de paramétrage d'interface utilisée l'interface "TCP/IP(Auto) -&gt; avec votre carte réseau qui est directement connectée avec le Panel et l'automate, par exemple 3Com EtherLink XL.</li> <li>3. Cliquer ensuite sur OK puis confirmer également le message suivant en cliquant sur OK.</li> </ol> <div style="text-align: center;">  </div>

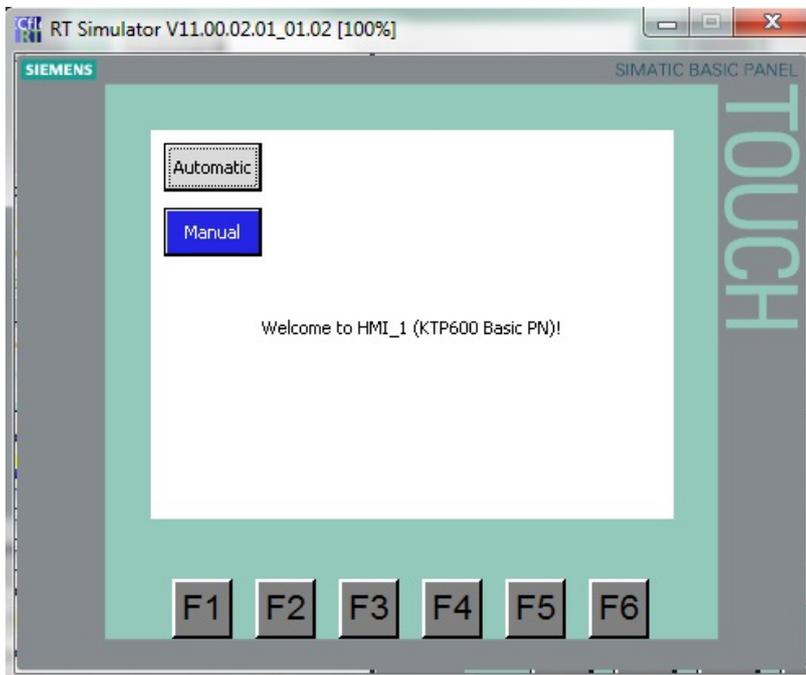
### 7.5 Démarrage de la configuration sous Runtime

Sélectionner dans la fenêtre de projet le Panel HMI\_1 [KTP600 Basic PN]

Cliquer sur le bouton **Start simulation (Démarrer simulation)**.

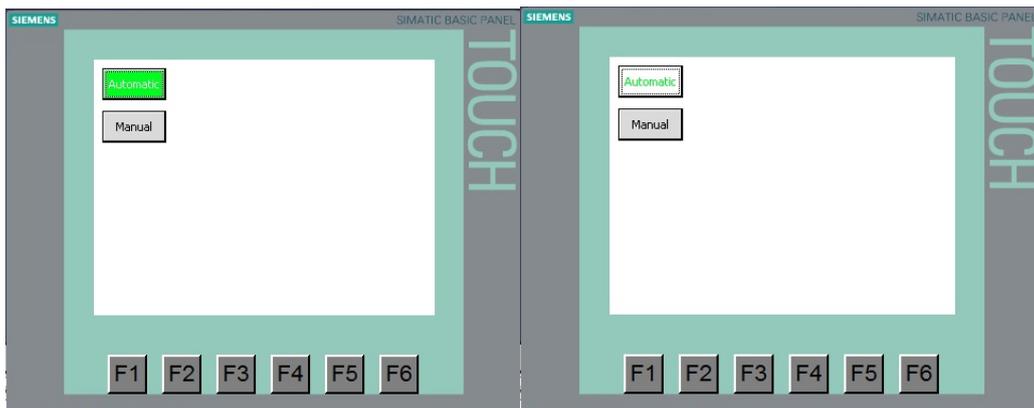


La visualisation apparaît dans RT Simulator.



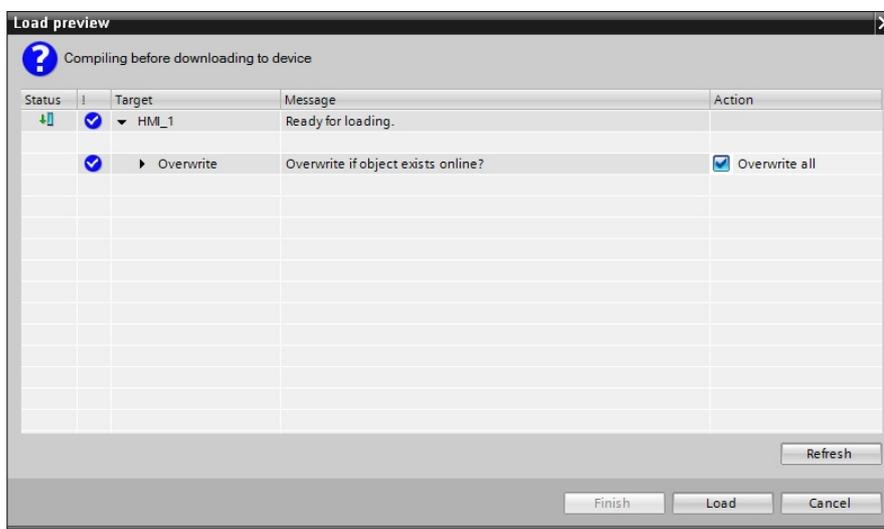
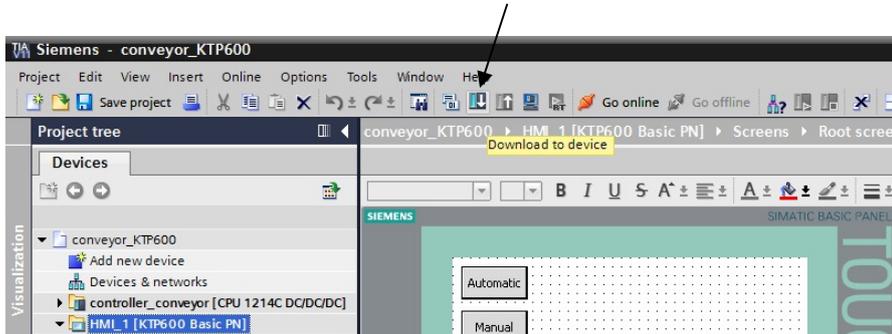
Tester votre projet.

Maintenant, le mode automatique ou manuel est sélectionné par défaut sur le Panel.

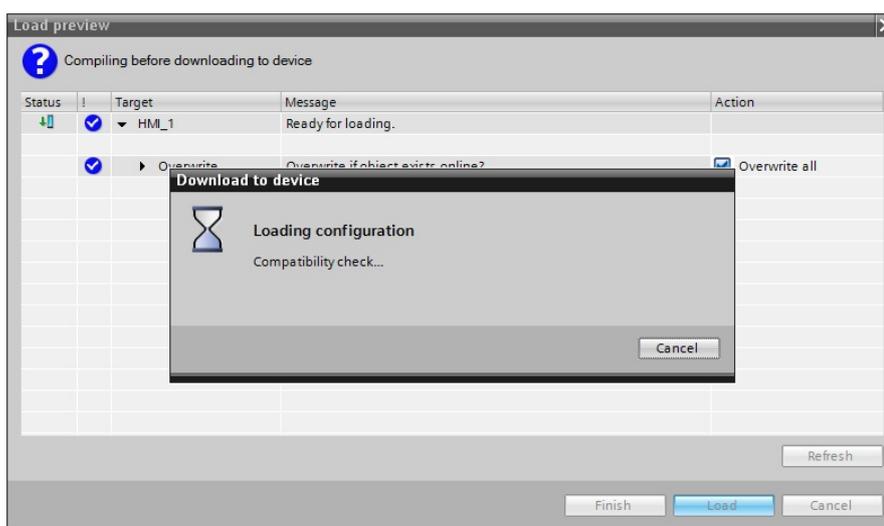


## 7.6 Chargement et test de la configuration sur le Panel

Sélectionner dans la fenêtre de projet le Panel **HMI\_1 [KTP600 Basic PN]**  
 Cliquer ensuite sur le bouton **Download to device (Charger dans l'appareil)**.



Cliquer sur le bouton **Load (Charger)**.



Si le système d'exploitation sur le Panel n'est pas à jour, une fenêtre supplémentaire s'affiche pour le mettre à jour.

Tester aussi la touche de fonction **F6**.

## 7.7 Boutons Marche et Arrêt

Nous allons maintenant configurer les boutons Marche et Arrêt.

Le bouton **Start (Marche)** est créé exactement comme les boutons Automatique et Manuel.

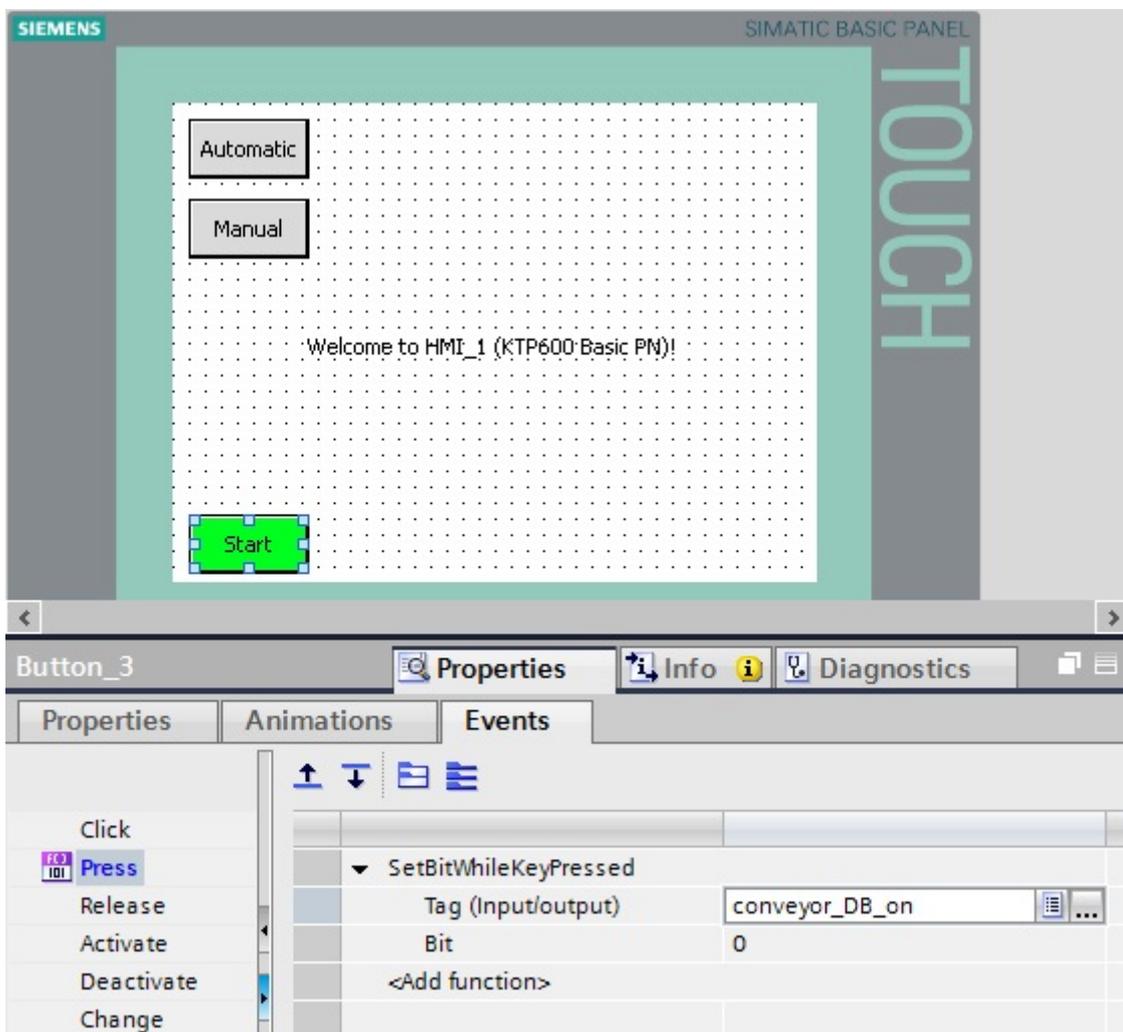
Le bouton **Stop (Arrêt)** a une fonction d'ouverture et doit supprimer le signal lorsqu'il est activé.

Créer le bouton **Start (Marche)**.

Mettre l'arrière-plan en **vert**.

Sous Event (Événements), Press (Appuyer), ajouter la fonction **SetBitWhileKeyPressed** (**PositionnerBitTantQueToucheEnfoncée**).

Choisir la variable **on** sous **conveyor\_DB [DB1]**.

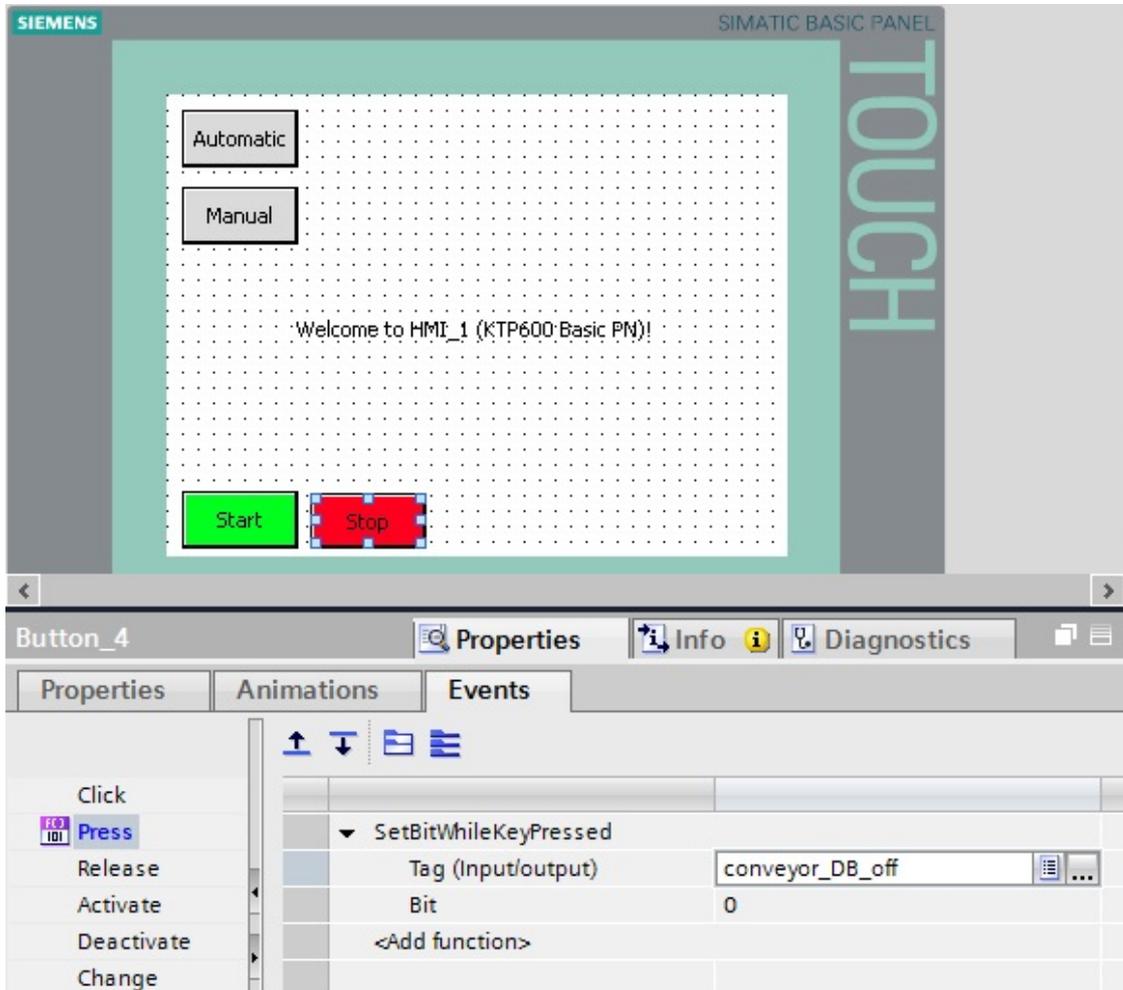


Créer le bouton **Stop (Arrêt)**.

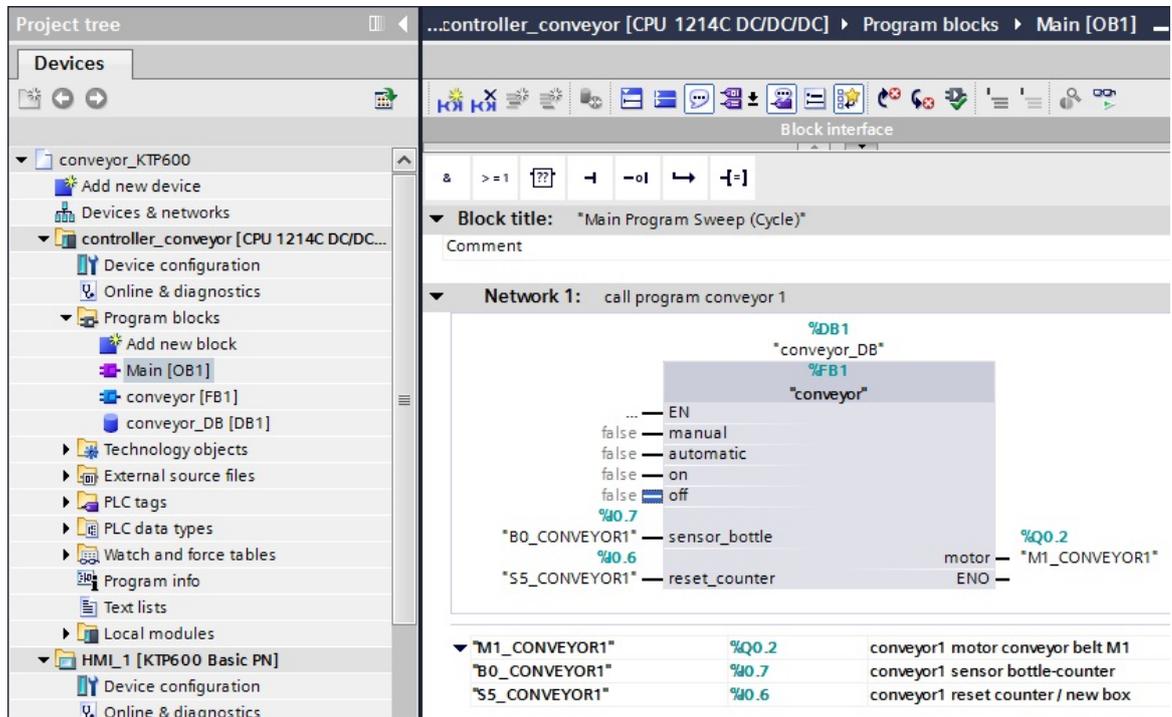
Mettre l'arrière-plan en **rouge**.

Sous Event (Événements), Press (Appuyer), ajouter la fonction **SetBitWhileKeyPressed (PositionnerBitTantQueToucheEnfoncée)**.

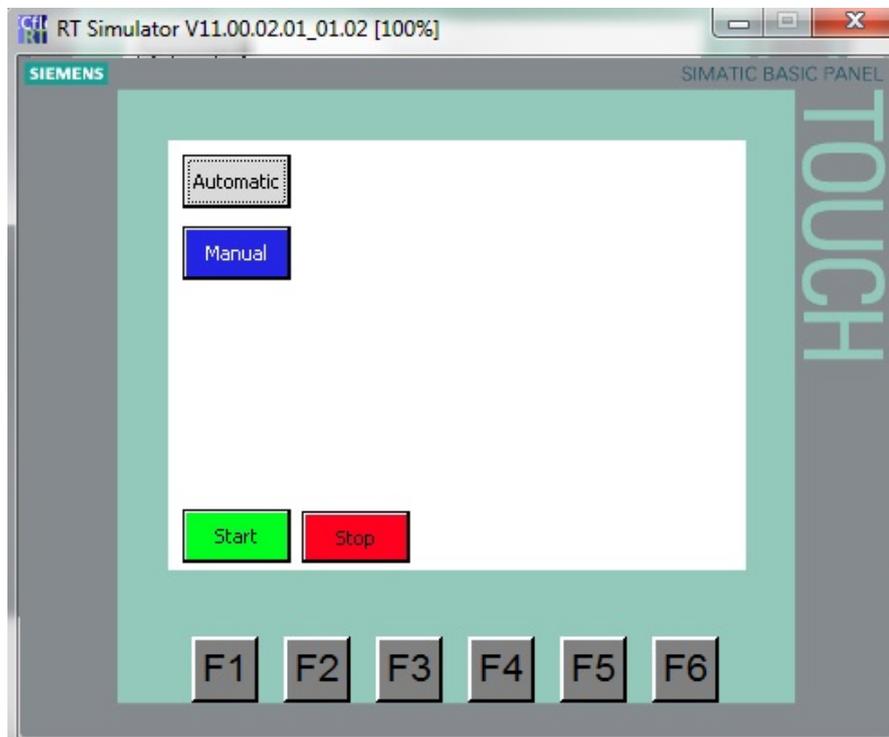
Choisir la variable **off** sous **conveyor\_DB [DB1]**.



Avant de tester la visualisation, nous devons faire un autre changement dans le programme STEP 7.  
 Dans OB1, supprimer l'affectation S3\_Conveyor1 et S4\_Conveyor1 lors de l'appel du FB1.  
 Supprimer la négation sur l'entrée **off** du bloc.  
 Enregistrer et charger le programme modifié.



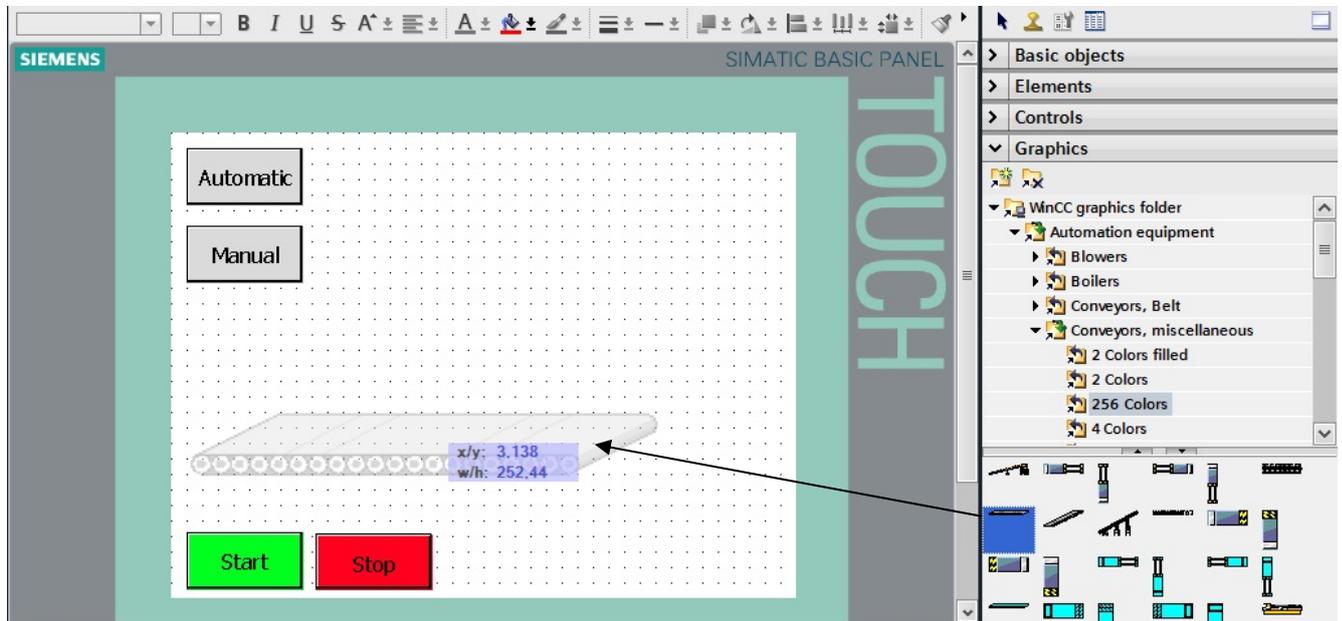
Charger la configuration sur le Panel et tester les boutons **Start (Marche)** et **Stop (Arrêt)**.



## 7.8 Insertion de graphiques depuis le dossier graphique

Dans la boîte à outils, sous Graphiques, ouvrir l'arborescence **WinCC graphics folder (Dossier graphique WinCC)**

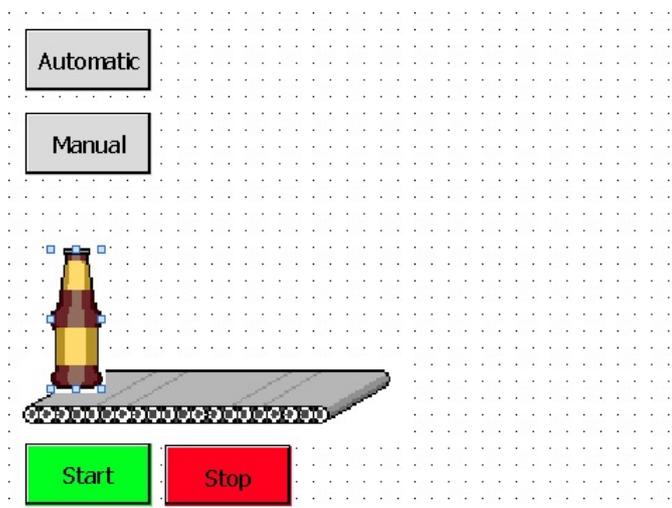
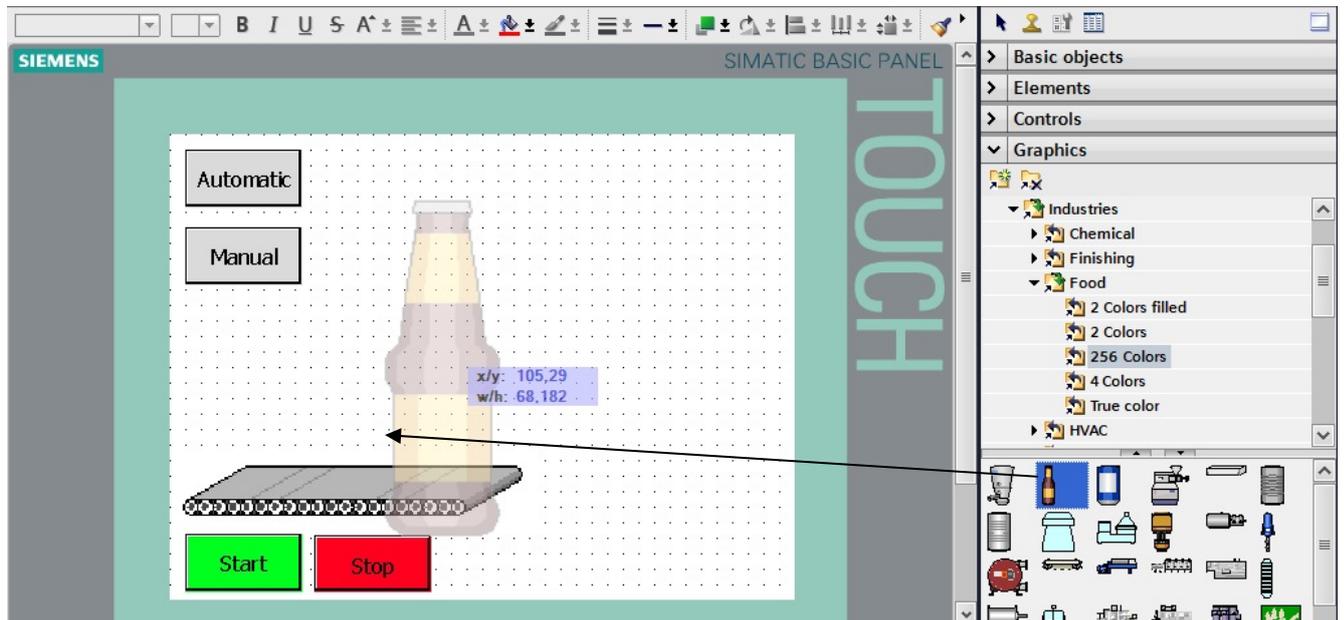
Faites glisser le graphique du convoyeur sur la vue racine.



Dans la boîte à outils, sous Graphiques, ouvrir l'arborescence **WinCC graphics folder (Dossier graphique WinCC)**

Faites glisser le graphique de la bouteille de bière sur la vue racine.

Modifier la taille et la position de la bouteille.



### Remarque

Tous les objets de vue doivent se trouver **dans** la zone de travail (320x240 pixels).

### 7.9 Programme de simulation du mouvement de la bouteille

Pour simuler le mouvement de la bouteille et le capteur de bouteille, il faut créer un nouveau bloc. Le bloc Simulation [FB2] ci-dessous, avec déclarations de variables et réseaux, est constitué d'un compteur qui compte toujours de 0 à 51 à partir d'un signal de départ.

Dans le réseau 1, le CTU (comptage) est inséré en multi-instance.

Dans le réseau 2, un signal d'impulsion du capteur de bouteille est reçu quand le nombre 50 est atteint.

Tout ceci va simuler le fait qu'une bouteille est sortie du convoyeur.

**Project tree**

- Devices
- conveyor\_KTP600
  - Add new device
  - Devices & network
  - controller\_conve
    - Device config
    - Online & diagnostics
    - Program blocks
      - Add new block
      - Main [OB1]
      - simulation [FB2]
      - conveyor [FB1]
      - conveyor\_DB [DB1]
    - System blocks
    - Technology objects
    - External source files
    - PLC tags
    - PLC data types
    - Watch and force tables
    - Program info
    - Text lists
    - Local modules
    - HML\_1 [KTP600 Basic PN]
      - Device configuration
      - Online & diagnostics

Interface						
	Name	Data type	Default value	Retain	Visible in ...	Comment
1	Input					
2	start	Bool	false	Non-retentive	<input checked="" type="checkbox"/>	
3	pulse	Bool	false	Non-retentive	<input checked="" type="checkbox"/>	
4	<Add new>				<input type="checkbox"/>	
5	Output					
6	bottle_sensor	Bool	false	Non-retentive	<input checked="" type="checkbox"/>	
7	<Add new>				<input type="checkbox"/>	
8	InOut					
9	<Add new>				<input type="checkbox"/>	
10	Static					
11	IEC_Counter_1	IEC_COUNTER		Non-ret...	<input checked="" type="checkbox"/>	
12	CU	Bool	false	Non-retentive	<input checked="" type="checkbox"/>	
13	CD	Bool	false	Non-retentive	<input checked="" type="checkbox"/>	
14	R	Bool	false	Non-retentive	<input checked="" type="checkbox"/>	
15	LD	Bool	false	Non-retentive	<input checked="" type="checkbox"/>	
16	QU	Bool	false	Non-retentive	<input checked="" type="checkbox"/>	
17	QD	Bool	false	Non-retentive	<input checked="" type="checkbox"/>	
18	PV	Int	0	Non-retentive	<input checked="" type="checkbox"/>	
19	CV	Int	0	Non-retentive	<input checked="" type="checkbox"/>	
20	<Add new>				<input type="checkbox"/>	
21	Temp					
22	count_value	Int			<input type="checkbox"/>	

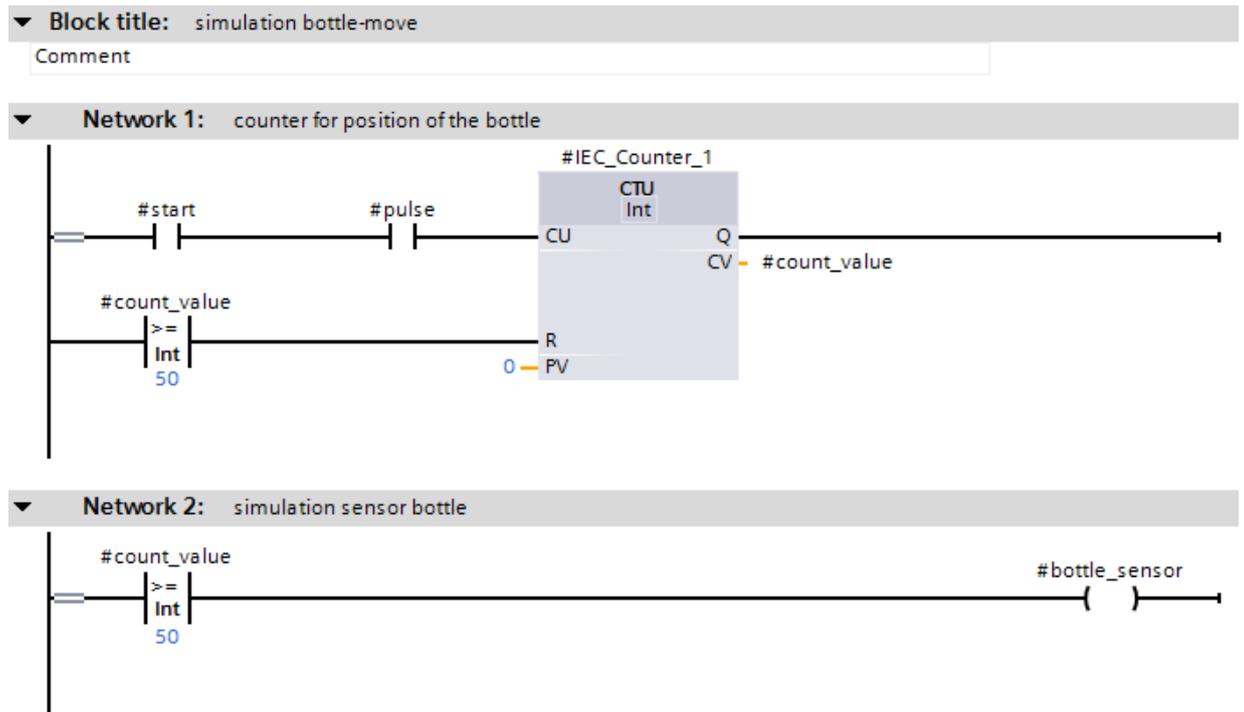
  

**Network 1: counter for position of the bottle**

**Network 2: simulation sensor bottle**

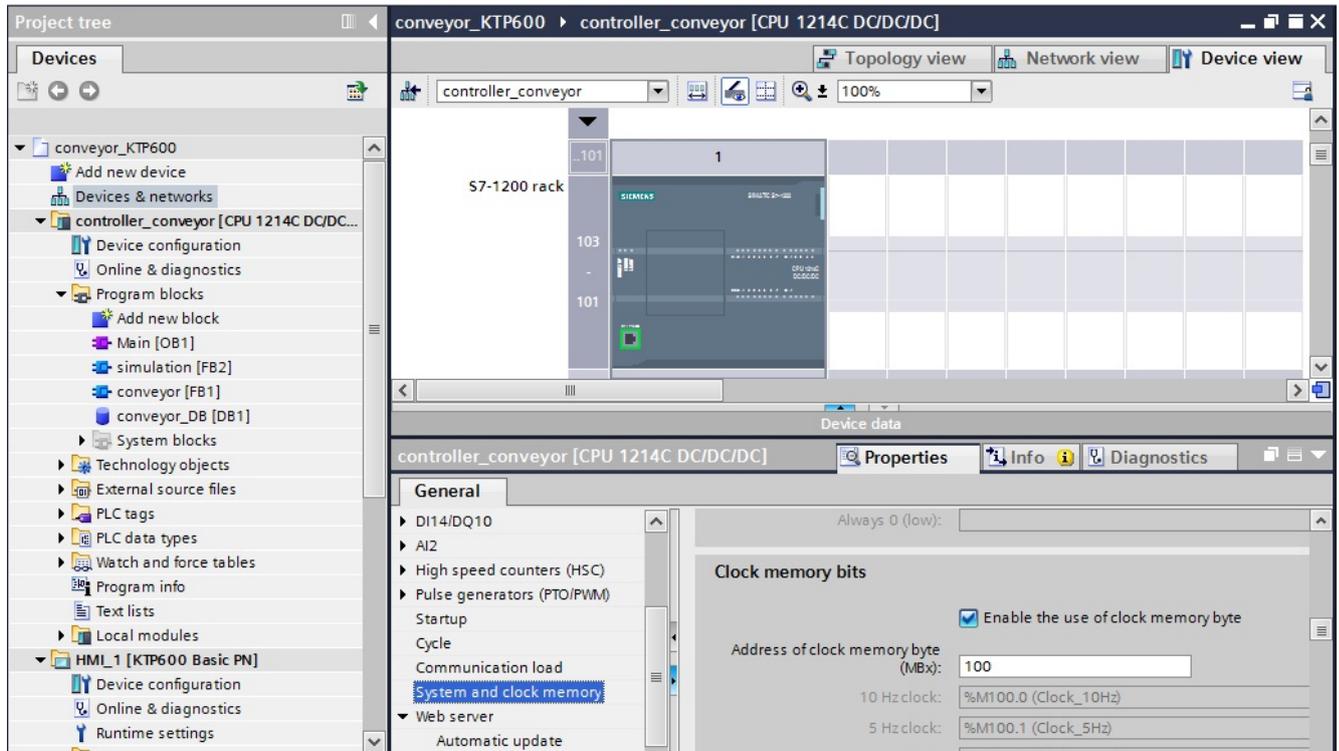
**Programme dans le schéma à contacts (CONT) :**



## 7.10 Activation du memento de cadence et affectation de MB100

Un bit de memento de cadence interne à l'API est utilisé comme horloge.

Activer les bits de mementos de cadence et attribuer comme adresse MB100.



## 7.11 Appel du bloc fonctionnel FB2 (Simulation) dans OB1

Avant d'appeler le bloc fonctionnel FB1 (conveyor), ajouter un nouveau réseau.

Appeler le bloc Simulation (FB2) **avant** le bloc Conveyor (FB1).

Dans OB1, créer la variable Temp **bottle (bouteille)** et interconnecter les blocs.

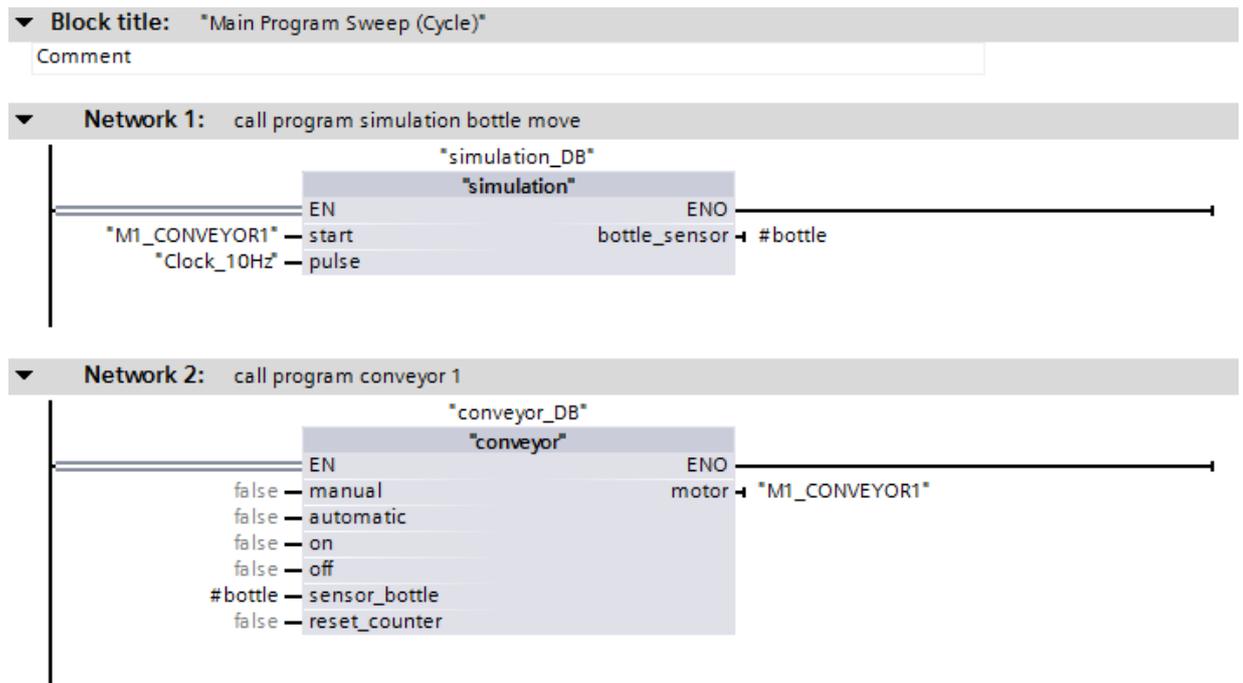
Ensuite, enregistrer le projet et le charger dans l'API.

The screenshot displays the SIMATIC Manager interface with the following details:

- Project tree (left):** Shows the project structure for 'conveyor\_KTP600', including 'controller\_conveyor [CPU 1214C DC/DC...]' and 'Program blocks' such as 'simulation [FB2]', 'simulation\_DB [DB3]', and 'conveyor [FB1]'.
- Interface table (top right):**

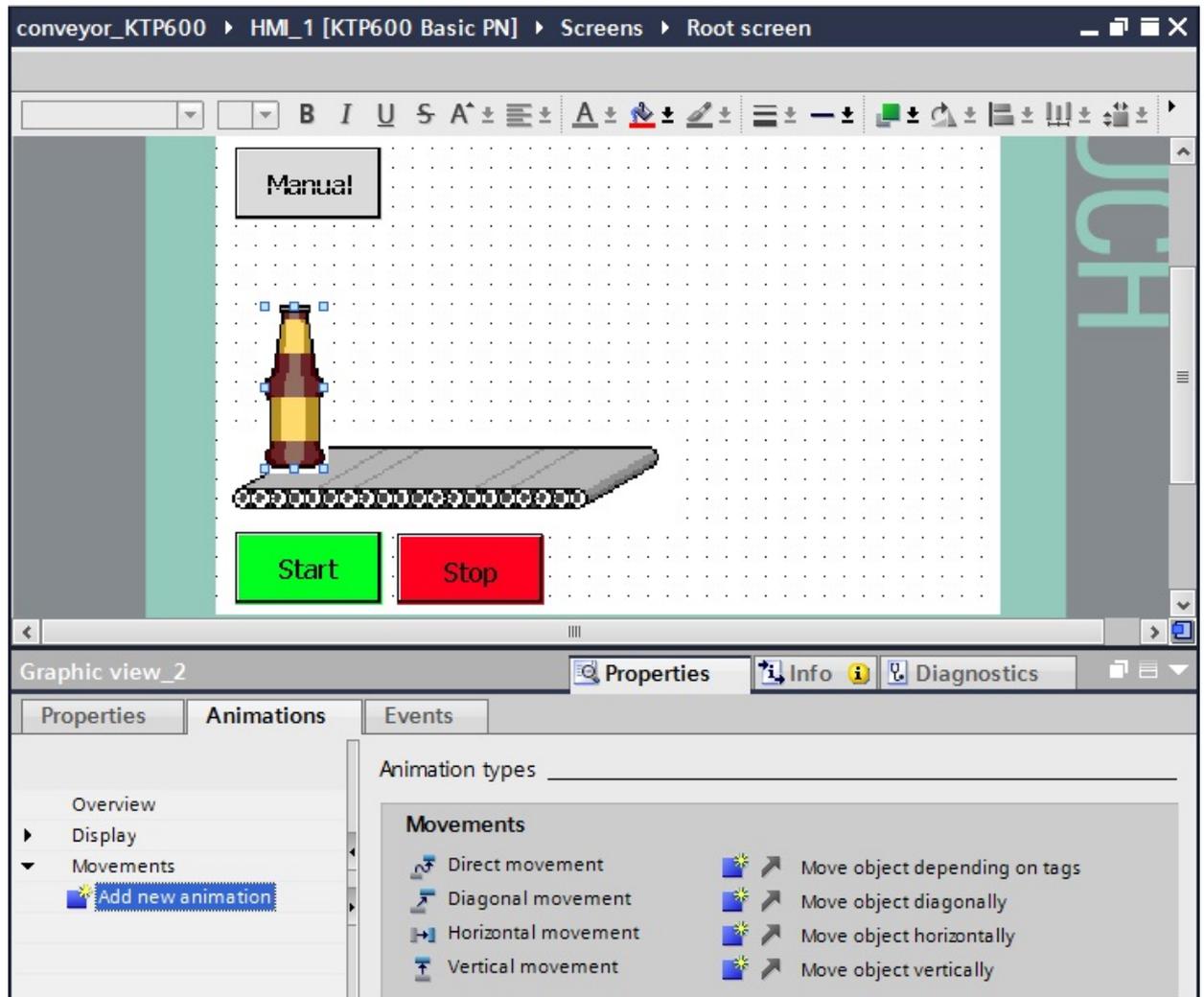
Name	Data type	Comment
1	Temp	
2	bottle	Bool
- Network 1: call program simulation bottle move**
  - Block: **"simulation"** (FB2) with data type **%FB2** and DB **%DB3 "simulation\_DB"**.
  - Inputs: **start** (M1\_CONVEYOR1), **pulse** (Clock\_10Hz).
  - Outputs: **bottle\_sensor** (labeled #bottle), **ENO**.
- Network 2: call program conveyor 1**
  - Block: **"conveyor"** (FB1) with data type **%FB1** and DB **%DB1 "conveyor\_DB"**.
  - Inputs: **manual** (false), **automatic** (false), **on** (false), **off** (false), **sensor\_bottle** (#bottle), **reset\_counter** (S5\_CONVEYOR1).
  - Outputs: **motor** (M1\_CONVEYOR1), **ENO**.

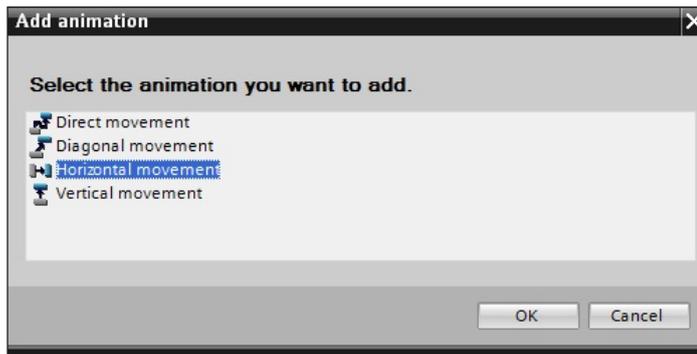
**Programme dans le schéma à contacts (CONT) :**



## 7.12 Configuration du mouvement de la bouteille

Sélectionner la bouteille et sous l'onglet **Propriétés (Propriétés) / Animations**, Movements choisir **Add new animation (Ajouter une animation)**.

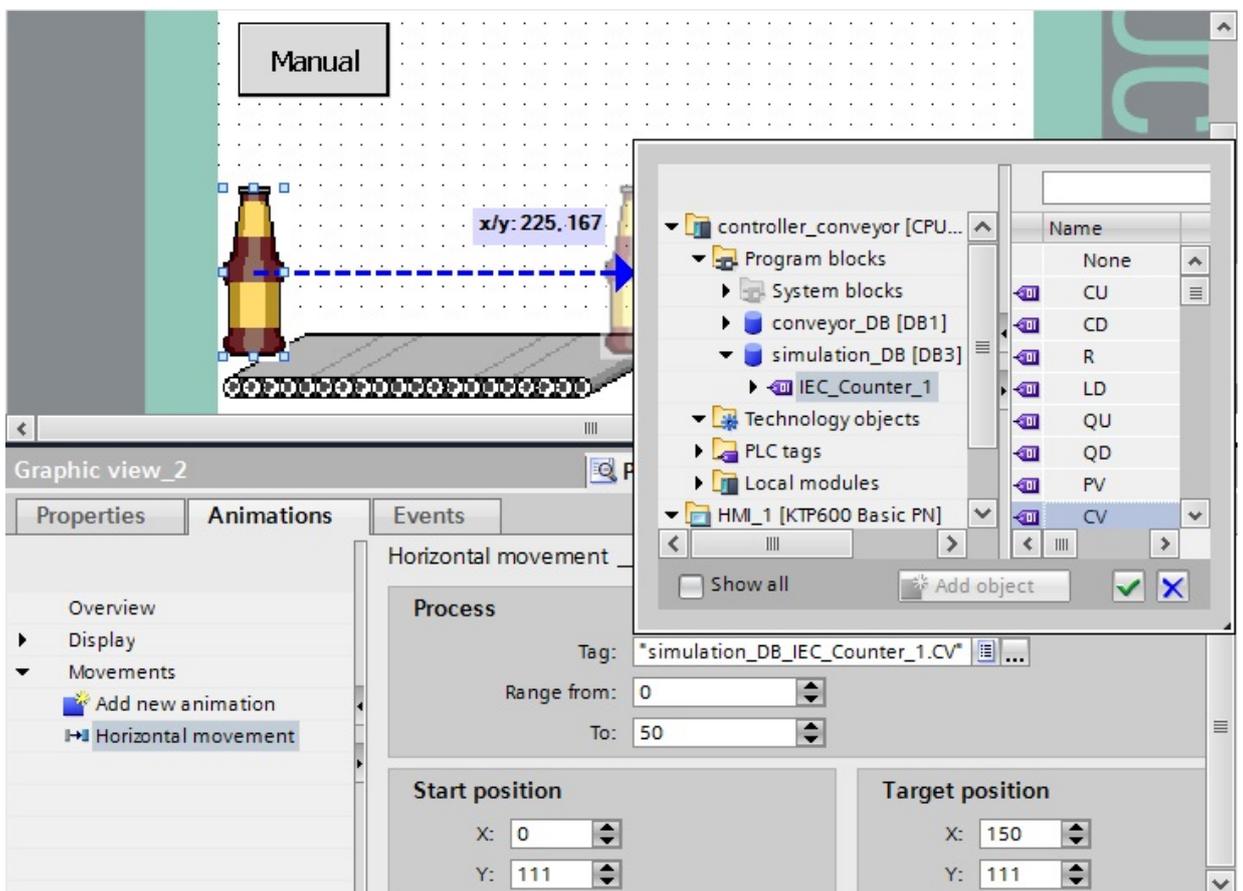




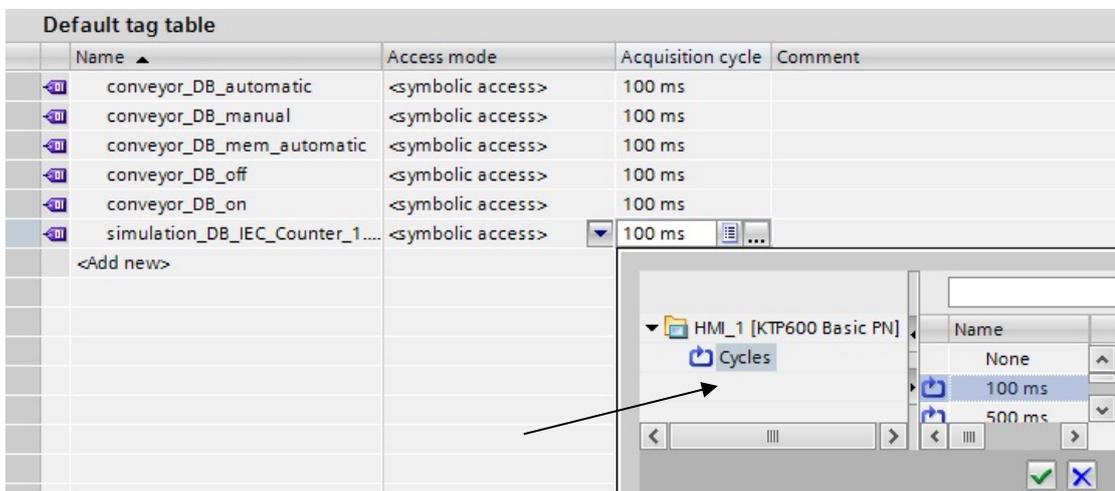
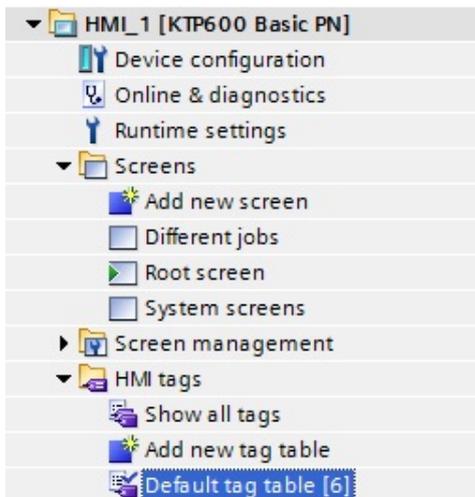
Choisir la variable **CV** du IEC\_Counter\_1 dans Simulations\_DB [DB2].

Sous Range entrer la plage de **0 à 50**.

Modifier la position cible de la bouteille jusqu'à la fin **X150**.



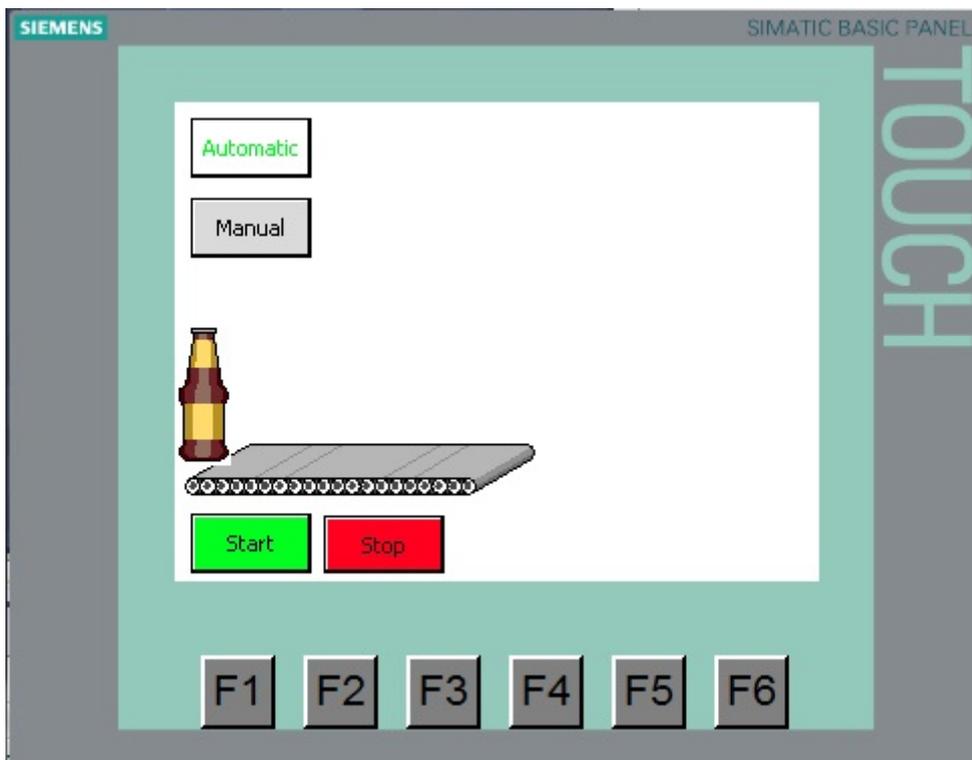
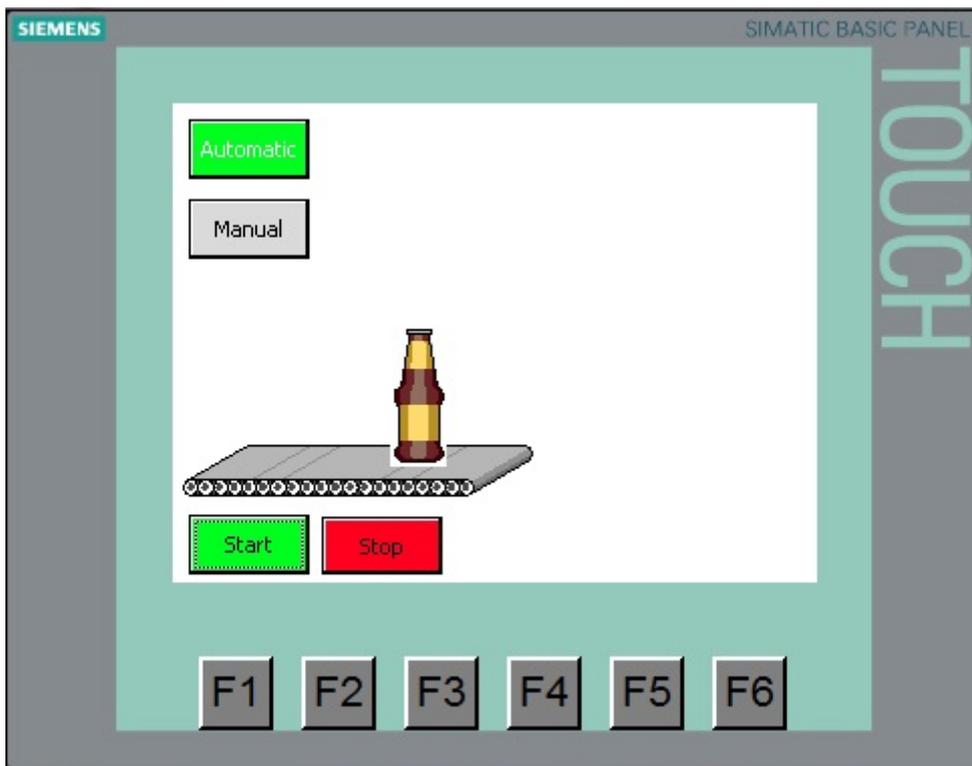
Dans le navigateur du projet, choisir **HMI tags (Variables IHM)**.



Déplacer le curseur de la fenêtre vers la droite pour faire apparaître la colonne Acquisition cycle (Cycle d'acquisition).

Régler le cycle d'acquisition des **variables IHM** sur **100 ms**.

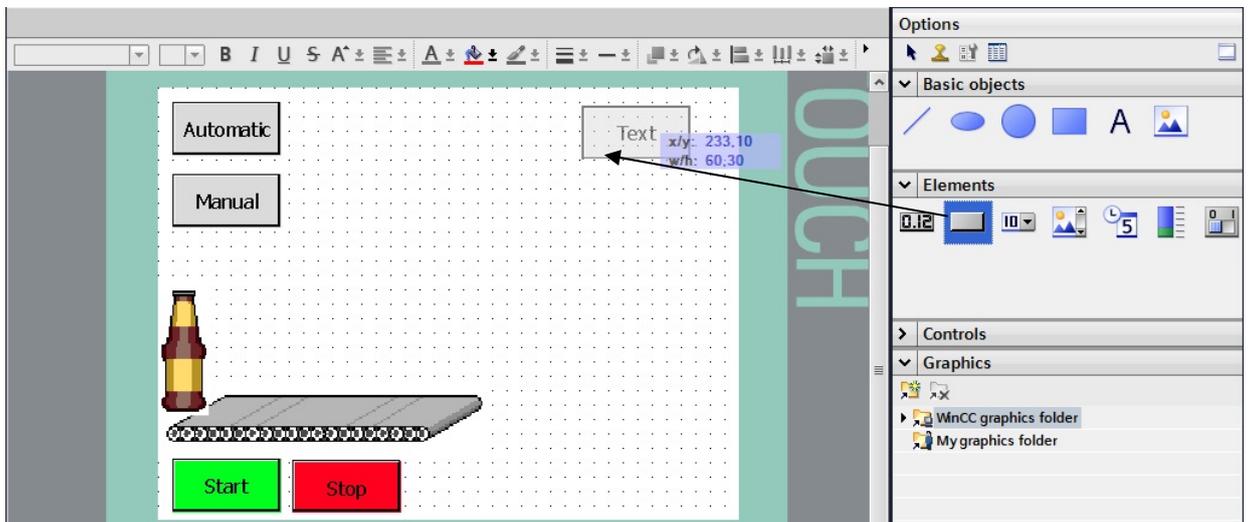
Ensuite, enregistrer le projet, le charger et le tester dans le Panel.



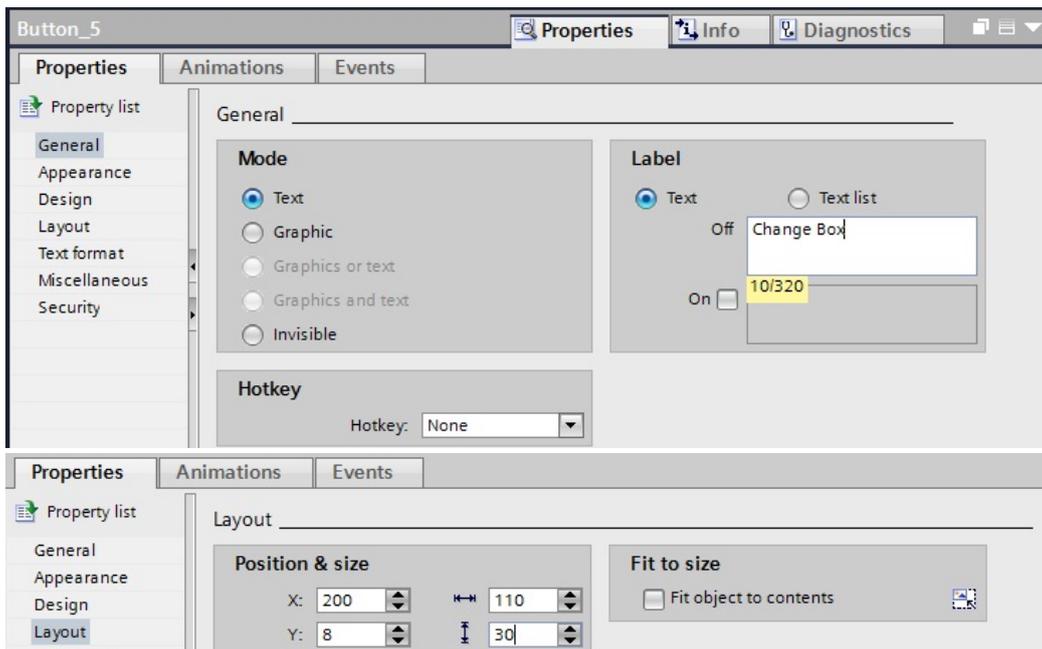
Après 20 bouteilles, le moteur du convoyeur s'arrête.  
Le compteur de bouteilles doit être remis à zéro pour qu'il reparte.

### 7.13 Remise à zéro du compteur de bouteilles

Placer un nouveau bouton dans la vue racine.

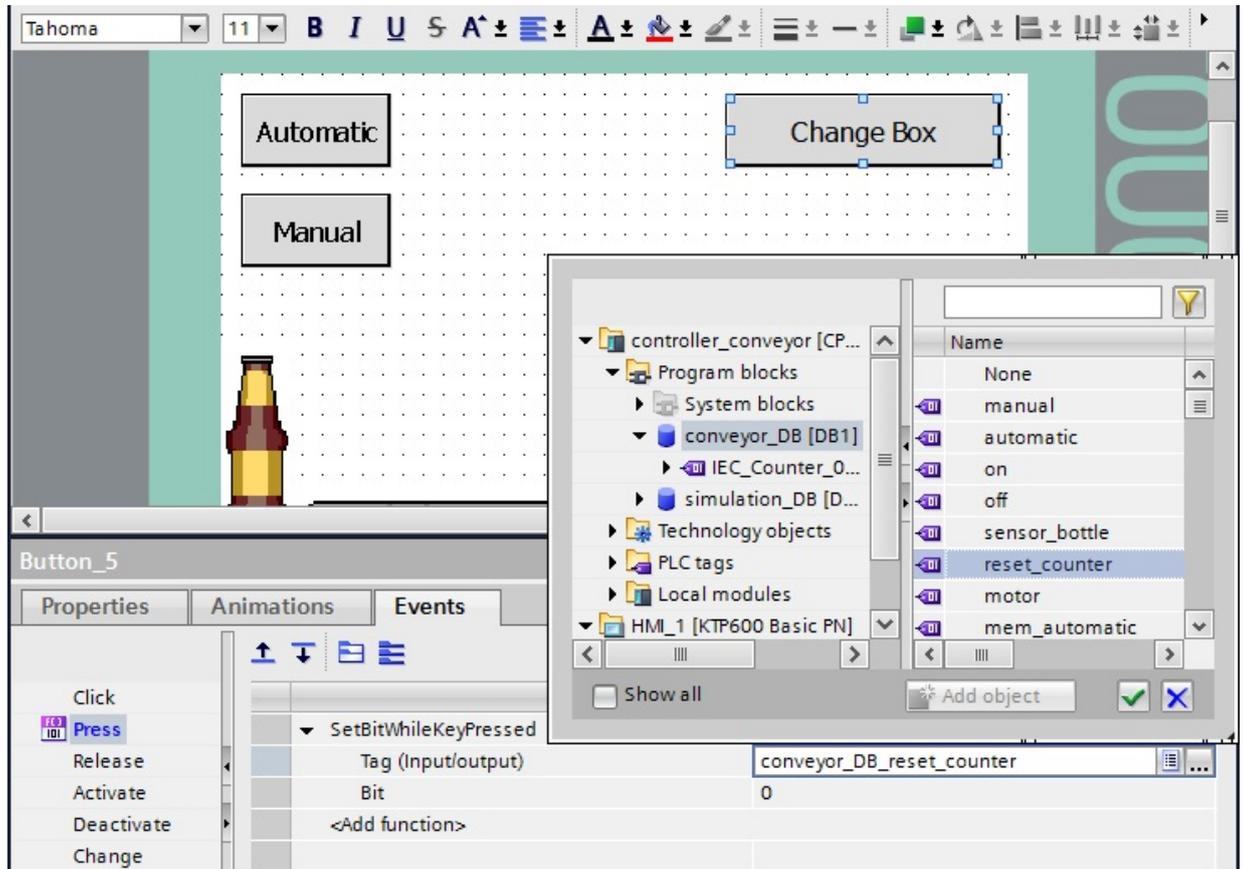


Saisir le texte **Change bottle (Changer la caisse)** et modifier la couleur (jaune), la position et la taille du bouton.



Sous Event (Événements), Press (Appuyer), ajouter la fonction **SetBitWhileKeyPressed** (**PositionnerBitTantQueToucheEnfoncée**).

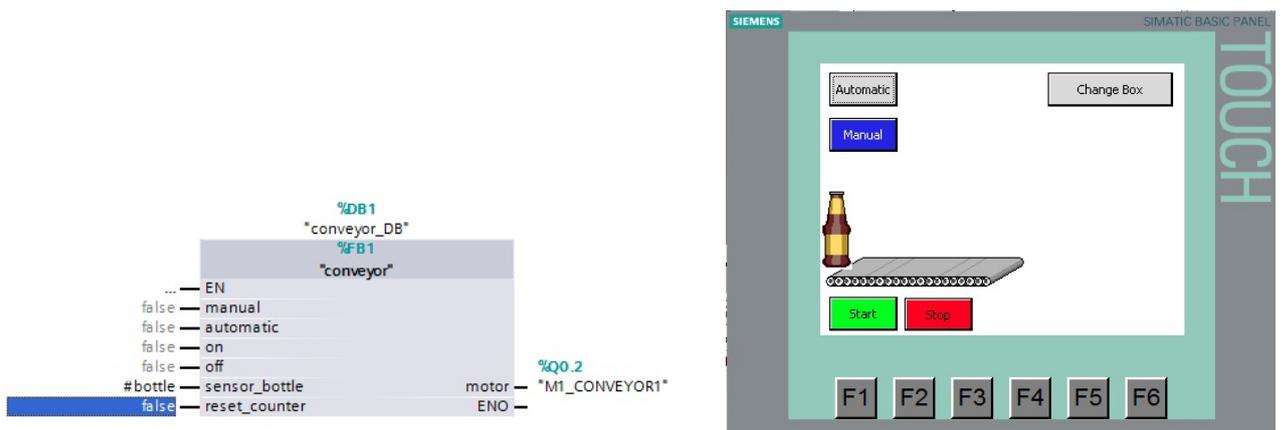
Choisir la variable **reset\_counter** sous **conveyor\_DB [DB1]**.



Régler le cycle d'acquisition des nouvelles **variables IHM** sur **100 ms**.

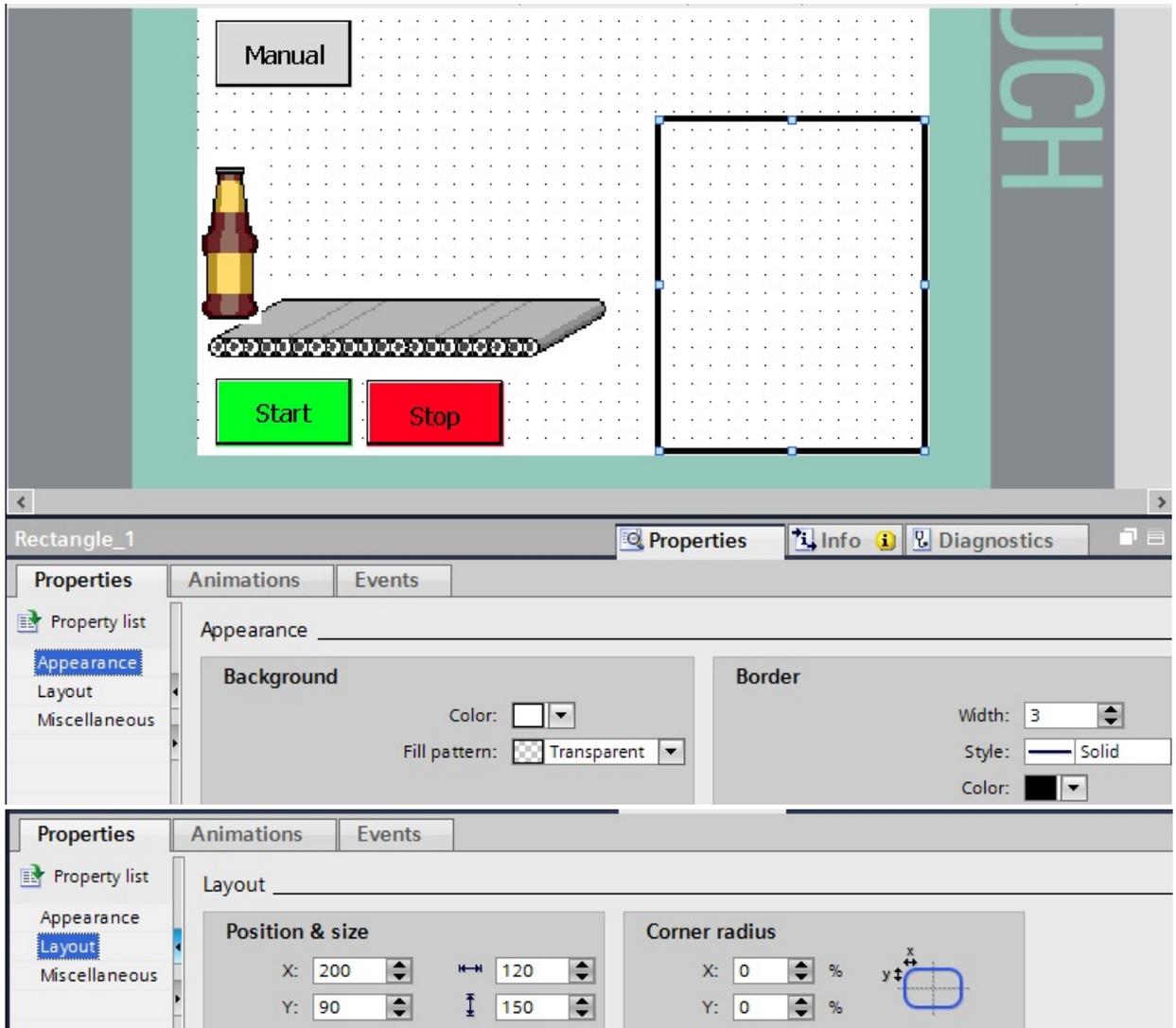
Dans OB1, supprimer lors de l'appel du Conveyor\_FB le câblage sous **reset\_counter**.

Ensuite, enregistrer le projet, le charger et le tester dans la CPU et le Panel.



## 7.14 Dessin de la caisse

Tracer un rectangle avec un arrière-plan transparent.  
Saisir la largeur de la bordure, la position et la taille.



Tracer une ligne verticale à une distance de 30 pixels.

The image shows a software interface for creating a user interface. At the top, there is a design canvas with a grid. On the left, there is a 'Manual' button, a bottle of beer on a conveyor belt, and 'Start' and 'Stop' buttons. On the right, there is a large rectangular area with a vertical line drawn at a distance of 30 pixels from the left edge. Below the canvas, there are two property panels for the selected vertical line.

**Line\_1 Properties Panel (Appearance):**

- Line Width: 2
- Line Style: Solid
- Line Color: Black
- Background color: White
- Fill pattern: Transparent
- Line ends Start: Default
- Line ends End: Default
- Line end shape: Round

**Line\_1 Properties Panel (Layout):**

- Position & size X: 230, Y: 90, Width: 0, Height: 150
- Start point X: 230, Y: 90
- End point X: 230, Y: 240

Tracer une ligne horizontale à une distance de 30 pixels.

The image shows a software interface for creating a user interface. At the top, there is a workspace with a grid background. On the left, there is a 'Manual' button, a beer bottle icon, a conveyor belt, and two buttons labeled 'Start' (green) and 'Stop' (red). On the right, there is a large rectangular frame with a grid inside. Below the workspace, there are two property panels for a selected line object named 'Line\_2'.

The first property panel is titled 'Line' and is under the 'Appearance' tab. It shows the following settings:

- Width: 2
- Style: Solid
- Color: Black
- Background color: White
- Fill pattern: Transparent

The 'Line ends' section shows:

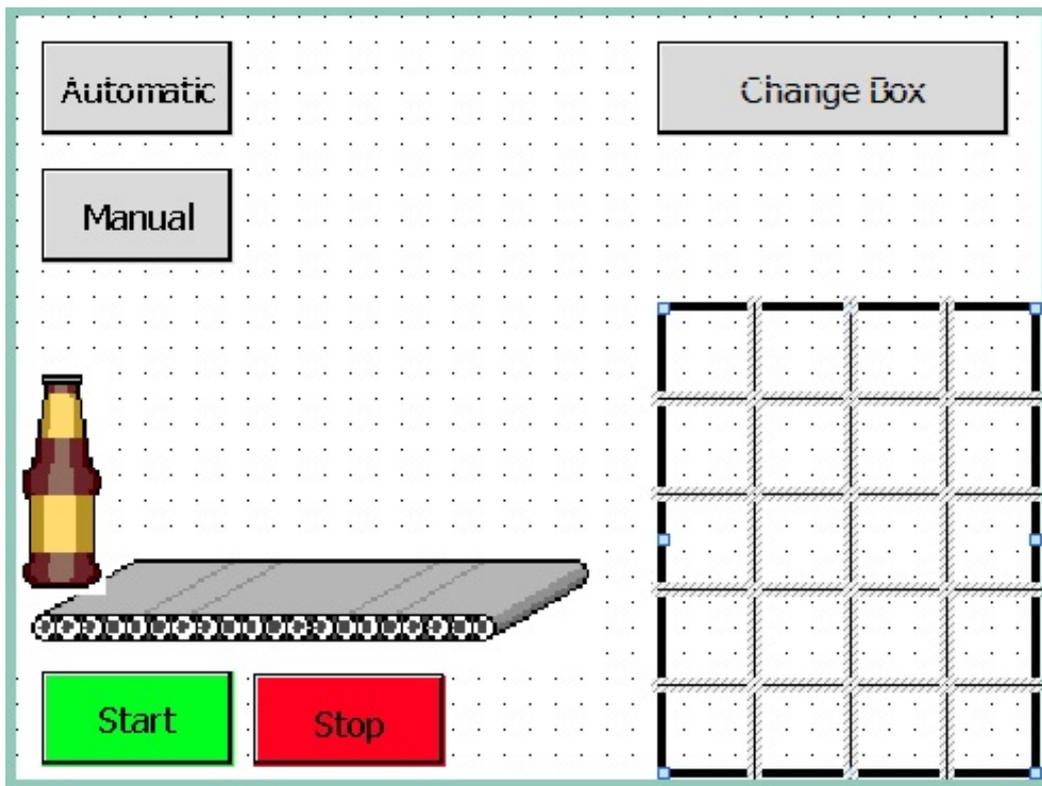
- Start: Default
- End: Default
- Line end shape: Round

The second property panel is titled 'Layout' and is under the 'Layout' tab. It shows the following settings:

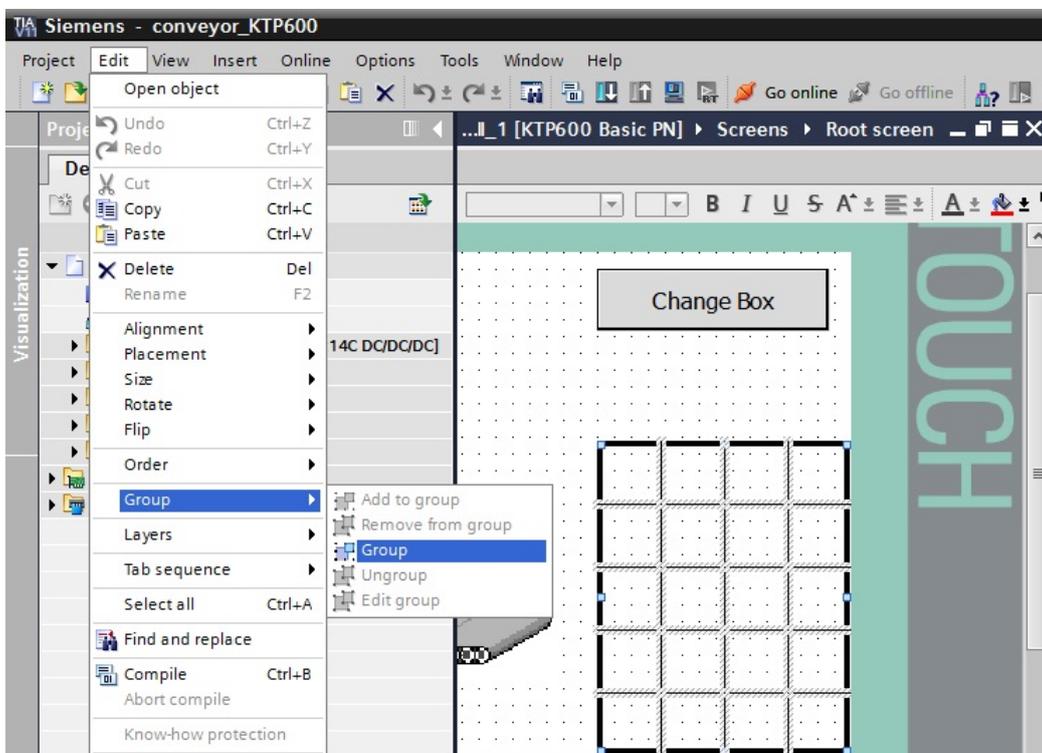
- Position & size: X: 200, Y: 120, Width: 120, Height: 0
- Start point: X: 200, Y: 120
- End point: X: 320, Y: 120

Avec l'aide de copier-coller, créez les lignes restantes, espacées chacune de 30 pixels.

Sélectionner la caisse en faisant un rectangle autour d'elle avec la souris.



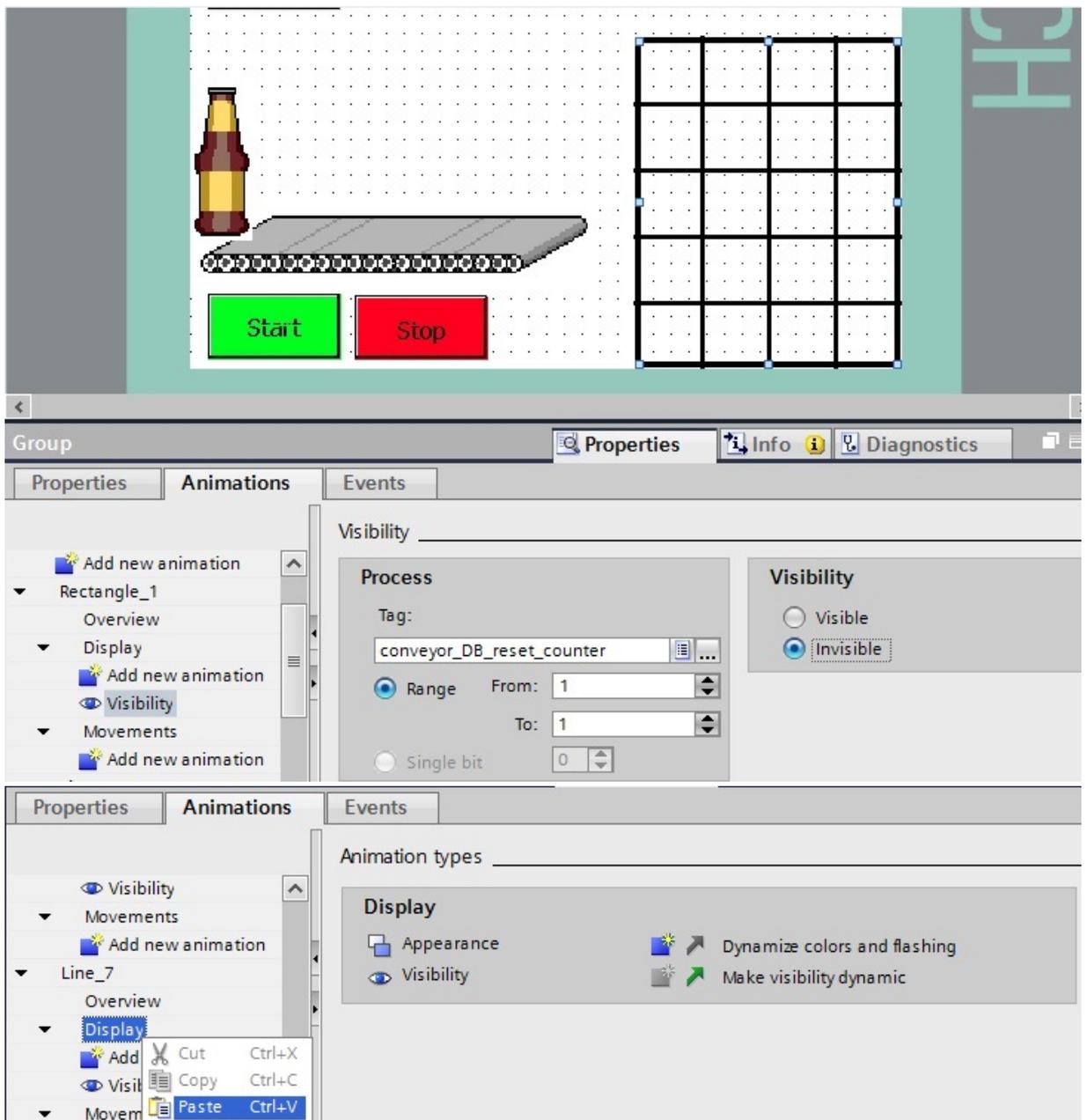
Sous le menu **Edit (Edition)**, choisir la fonction **Group (Grouper)**.



On ne souhaite pas afficher le rectangle et les lignes au moment où la caisse est changée.

Sous Rectangle\_1 et toutes les lignes, créer l'animation **Visibility (Visibilité)** avec comme variable **conveyor\_DB\_reset\_counter** et invisible quand 1.

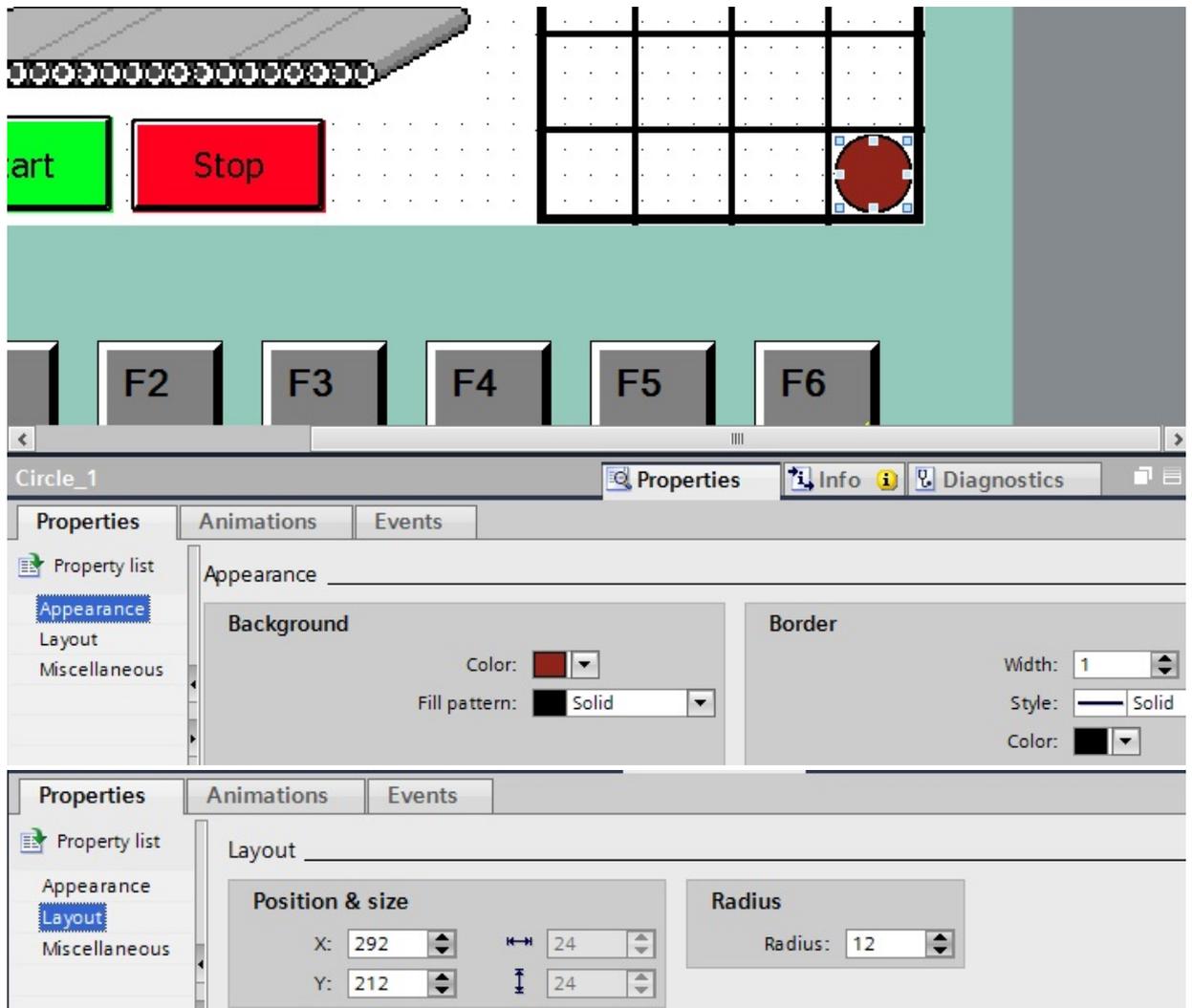
Pour les lignes, vous pouvez effectuer un copier-coller de l'animation pour chaque.



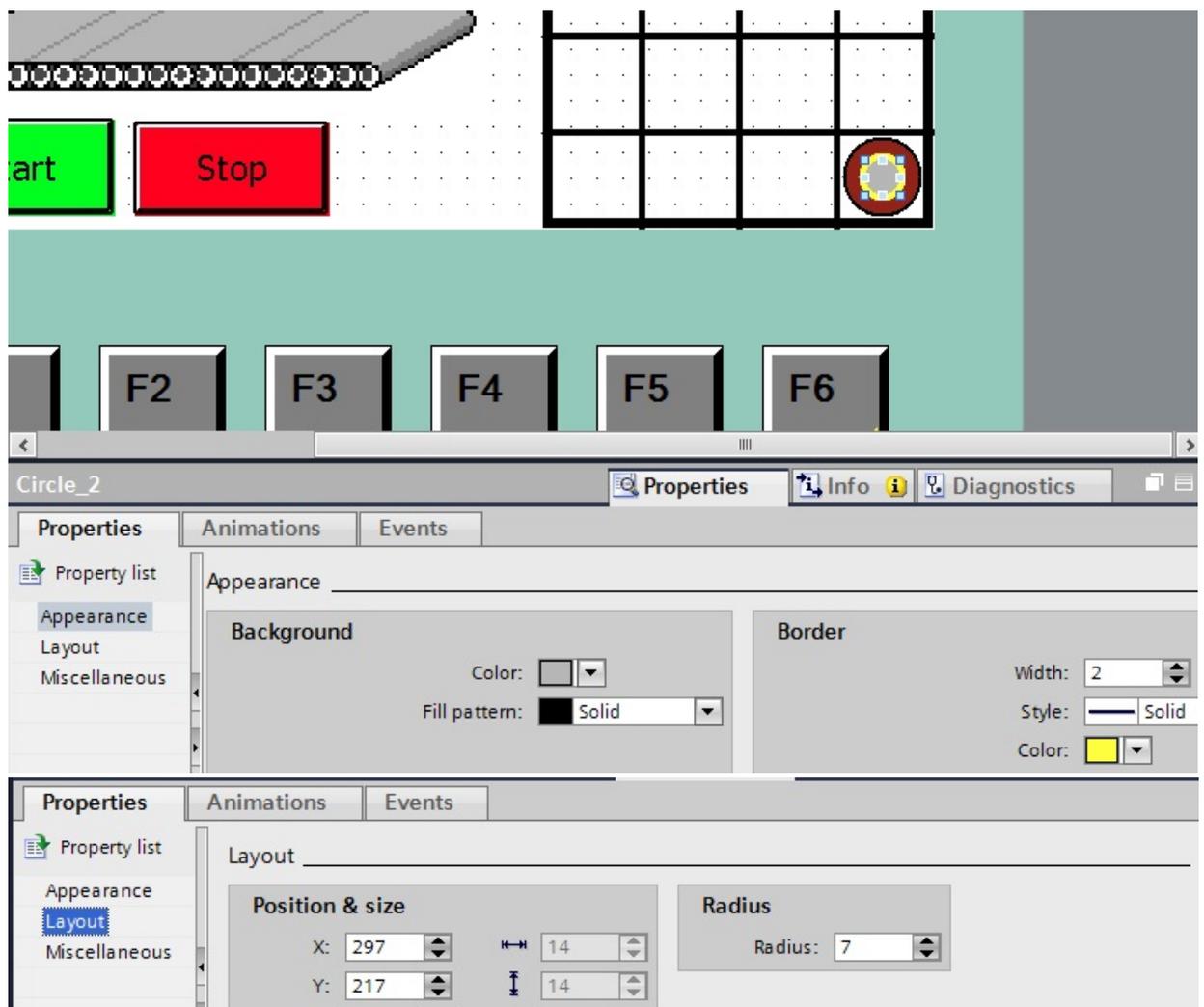
Ensuite, enregistrer le projet, le charger et le tester dans le Panel.

## 7.15 Dessin des bouteilles dans la caisse

Zoomer sur le coin en bas à droite de la caisse et dessinez-y un cercle.

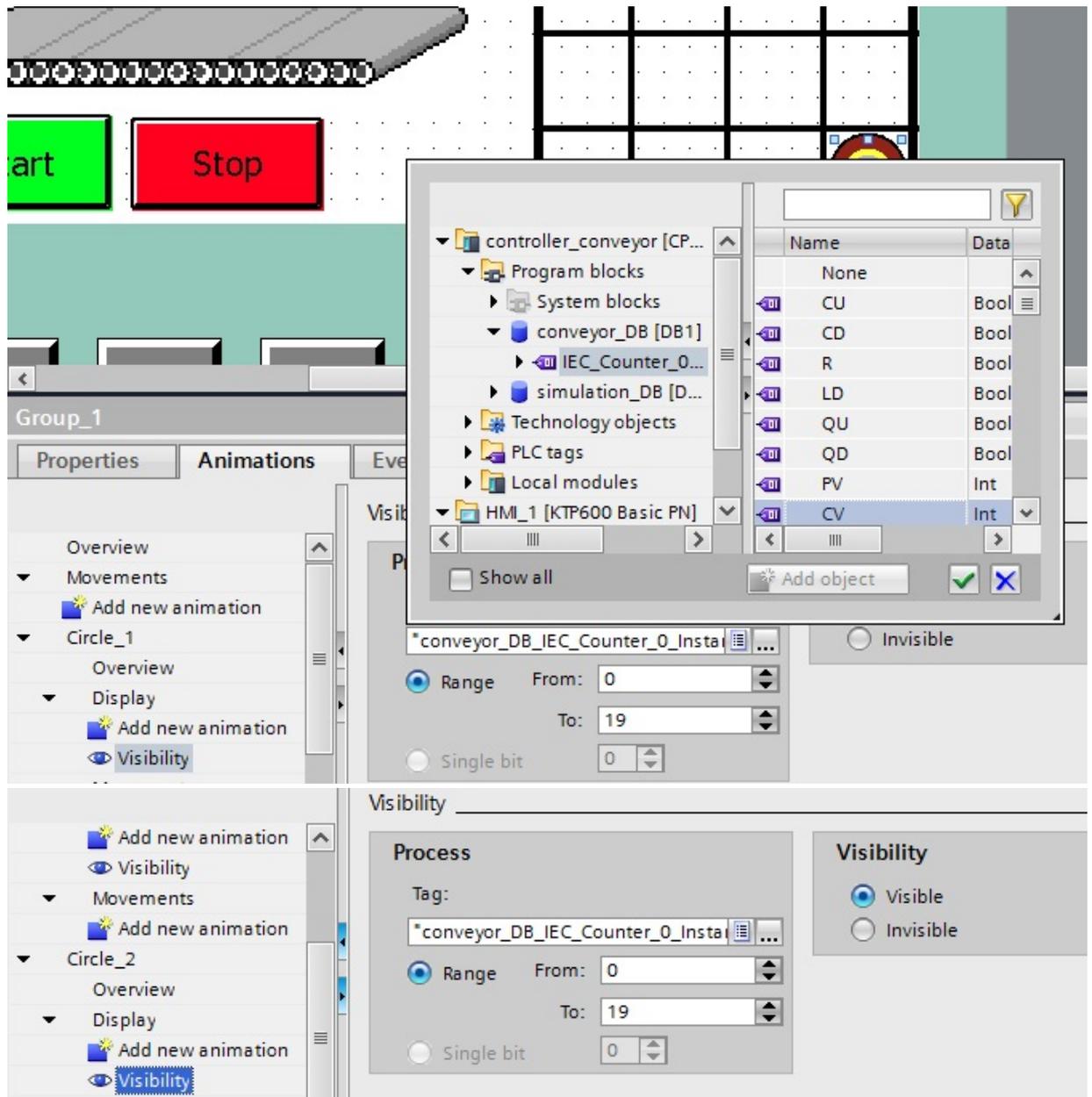


Dessiner un second cercle à l'intérieur du premier.



Grouper les 2 cercles.

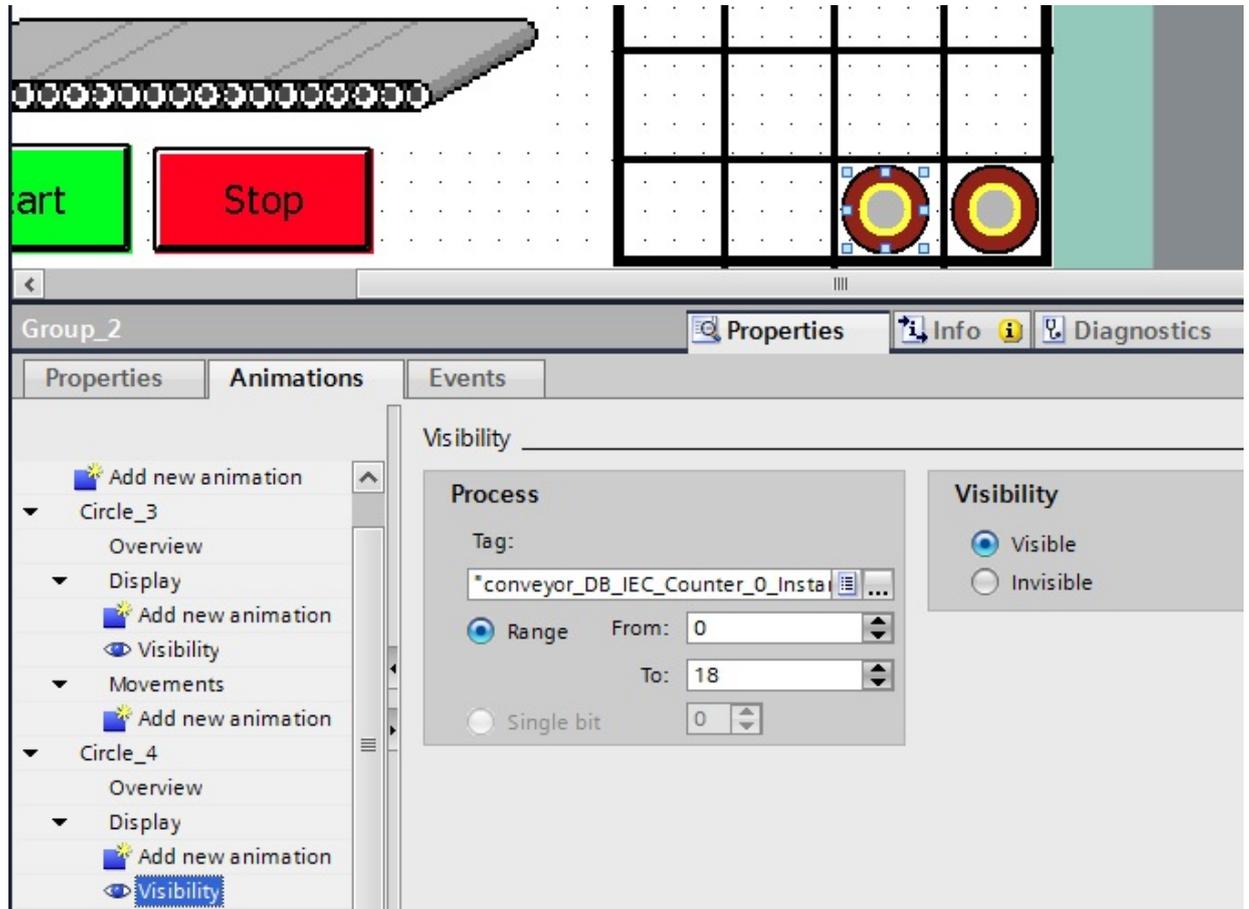
Sous Circle\_1 et Circle\_2, créer l'animation **Visibility (Visibilité)** avec la variable **conveyor\_DB\_IEC\_Counter\_0\_Instance\_CV** Range (page) **0 à 19** Visibles.



Copier-coller la bouteille.

Pour les deux cercles, sous Visibility, changer la plage de la variable

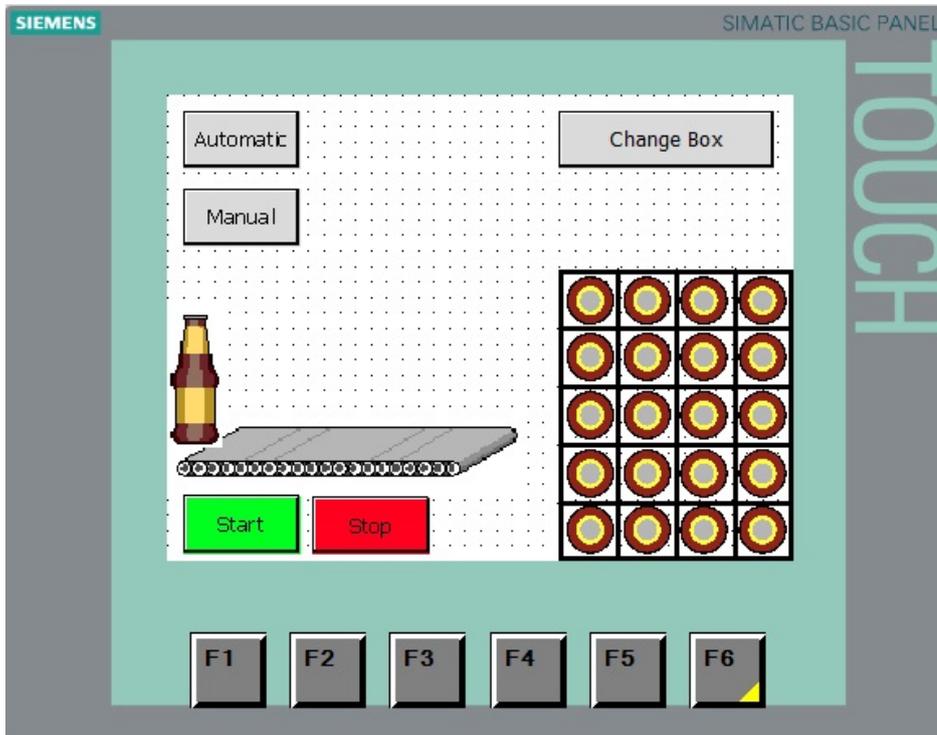
**Conveyor\_DB\_IEC\_Counter\_0\_Instance\_CV** sur **0 à 18** Visible.



Copier-coller les différentes bouteilles.

Pour l'animation **Visibility (Visibilité)** des deux cercles, baisser la plage (Range) to (à) de 1.

La dernière bouteille a une plage de 0 à 0.



Régler le cycle d'acquisition des nouvelles **variables IHM** sur **100 ms**.

Ensuite, enregistrer le projet, le charger et le tester dans le Panel.

