**Initiation à l’API Siemens S7-1200 et à l’interface**

**EazyPort USB**

**Introduction**

Les automatismes peuvent prendre des formes très variées selon les industries. Cependant, l’objectif reste le même à savoir : aider ou remplacer l’être humain dans les travaux pénibles, de précision ou répétitifs de manière à soulager la tâche de l’opérateur ou à obtenir une production plus constante et plus économique. Un système automatisé se compose de deux parties indépendantes qui dialoguent ensemble :

* Une partie commande PC.
* Une partie opérative PO.

La conception, l’étude et la réalisation d’un automatisme nécessite une démarche structurée qui fait appel à un outil de description des systèmes automatisés séquentiels dans l’ordre chronologique des étapes tels que : le chronogramme, l’organigramme et le GRAFCET.

1. **API Siemens S7-1200 :**
2. **Que signifie le terme API ?**

**API** est l’abréviation pour **A**utomate **P**rogrammable **I**ndustriel. Il s’agit d’un appareil qui commande un processus (par exemple une presse d’imprimerie pour l’impression des journaux, une installation de remplissage de sacs de ciment, une presse d’injection plastique, etc.…).  
Ceci est réalisé grâce aux instructions d’un programme stocké dans la mémoire de l’appareil.

1. **Comment l’API commande-t-il le processus ?**

L’API commande le processus en connectant des **actionneurs** via les points de connexion de l’API  
appelés **sorties** à une tension de commande de 24 V. Ceci permet de démarrer ou d’arrêter un  
moteur, de faire monter ou descendre des vannes ou d’allumer et éteindre des lampes.

1. **Comment l’API reçoit-t-il les informations sur les états du processus ?**

L’API reçoit les informations du processus à partir de ce qu’on appelle des **capteurs de signaux** qui  
sont câblés aux **entrées** de l’API. Ces capteurs de signaux peuvent être, par exemple, des capteurs qui reconnaissent si une pièce d'usinage se trouve à une position donnée ou de simples commutateurs ou boutons poussoirs, qui peuvent être ouverts ou fermés, appuyés ou relâchés. On distingue également entre les **contacts à ouverture (NF)** qui sont fermés au repos et les **contacts à fermeture (NO)** qui sont ouverts au repos.

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.

1. **Comment le SIMATIC S7-1200 adresse-t-il les signaux d’entrée/sortie ?**

La déclaration d'une entrée ou sortie donnée à l'intérieur d'un programme s'appelle l'adressage.  
Les entrées et sorties des API sont la plupart du temps regroupées en groupes de huit sur des modules d'entrées ou de sorties numériques. Cette unité de huit est appelée **octet**. Chaque groupe reçoit un numéro que l'on appelle l'**adresse d'octet**. Afin de permettre l'adressage d'une entrée ou sortie à l'intérieur d'un octet, chaque octet est divisé en huit **bits**. Ces derniers sont numérotés de 0 à 7. On obtient ainsi l'**adresse du bit…**  
Le SIMATIC S7-1200possède :

* 14 entrées numériques intégrées Octet 0 : Bits 0 à 7, et Octet 1 : Bits 0 à 5.
* 10 sorties numériques intégrées Octet 0 : Bits 0 à 7, et Octet 1 : Bits 0 à 1.

1. **Configuration et commande du SIMATIC S7-1200 :**

Le SIMATIC S7-1200 est un automate modulaire et tout un éventail de modules, décrits ci-après,   
l’accompagnent.

* Modules centraux CPU avec différentes capacités, entrées/sorties intégrées et interfaces  
  PROFINET (par exemple CPU 1214C).
* Module de puissance PM avec une entrée 120/230V CA, 50Hz/60Hz, 1.2A/0.7A, et une sortie 24V CC/2.5A.
* Signal Boards SB pour ajouter des entrées ou sorties analogiques ou numériques sans modifier la taille de la CPU. (Les Signal Boards peuvent être utilisés avec les CPU 1211C/1212C et 1214C).
* Modules d'entrées/sorties SM pour les entrées et sortie TOR et analogiques (pour les CPU 1212C, un maximum de 2 SM peut être utilisé, pour la 1214C un maximum de 8).
* Modules de communication CM pour la communication en série RS232 / RS 485 (Pour les CPU 1211C/1212C et 1214C, jusqu'à 3 CM peuvent être utilisés).
* Modules Compact Switch CSM avec 4 x prizes RJ45 10 / 100 Mbits/s.
* Cartes mémoire SIMATIC 2 Mo ou 24 Mo pour stocker les données du programme et pour le  
  remplacement aisé des CPU en cas de maintenance.

**Eléments importants de la CPU:**  
Avec une alimentation intégrée de 24V et des entrées et sorties numériques intégrées, la CPU du S7-  
1200 est prête à l’emploi, sans que des composants supplémentaires soient nécessaires.  
Pour communiquer avec l’appareil de programmation, la CPU est équipée d’un port TCP/IP intégré.  
Au moyen d’un réseau ETHERNET, la CPU est capable de communiquer avec des appareils de  
commande IHM et d’autres CPU.

* Alimentation 24V
* Borniers enfichables pour un câblage utilisateur (derrière les caches plastiques)
* DEL d'état pour les E/S intégrées et le mode de fonctionnement de la CPU
* Connexion TCP/IP (sous la CPU) Transfert du programme aux différentes CPU

La **carte mémoire SIMATIC MC (Memory Card)** stocke le programme, les données, les données système, les fichiers et les projets. Elle peut être utilisée pour les opérations suivantes :

Mise à jour du firmware des CPU, des modules de signal SM et des modules de communication CM.

1. **Aperçu de l'EasyPort USB:**

****

1. **Qu'est-ce que l'EasyPort USB ?**

L'interface de processus EasyPort USB sert à la transmission bidirectionnelle de signaux de processus entre un processus de commande réel en technique très basse tension (24 V=) et un PC. Afin d'exclure les rétroactions du processus sur le PC, la transmission des données entre l'EasyPort USB et le PC fait systématiquement appel à des interfaces à séparation galvanique. Les possibilités d'utilisation de l'EasyPort USB sont multiples:

* Commande d'un processus de commande réel par une commande tournant sur le PC.
* Commande d'un modèle de processus simulé par un API réel.
* Régulation d'un processus réel par un programme tournant sur le PC.
* Acquisition et dépouillement de mesures effectuées sur un processus réel.

1. **Comment utiliser l'EasyPort USB?**

Dans notre première manipulation, l'EasyPort USB relie le monde réel au monde virtuel du PC. L'interface de processus EasyPort USB dispose de 16 entrées et 16 sorties numériques (TOR) ainsi que de 4 entrées et 2 sorties analogiques.

Pour l'échange des signaux de processus entre l'EasyPort USB et le logiciel du PC, on dispose du serveur OPC EzOPC ou du contrôle ActiveX de l'EasyPort USB.

1. **Éléments de commande:**

Le panneau de commande de l'EasyPort USB comprend les éléments suivants:

* LED ERROR  
  Quand un court-circuit est détecté sur l'une des sorties, la LED rouge ERROR s'allume. Les sorties de l'EasyPort sont alors déconnectées. Cette LED s'allume aussi brièvement à la mise sous tension de l'EasyPort USB. Elle s'éteint au cours du test de mise sous tension.
* LED STATUS  
  La LED verte STATUS indique deux états:  
  – Clignotement à 1 Hz :  
  État après mise sous tension ; le module EasyPort USB ne communique pas encore.  
  – Clignotement modulé:  
  Le module EasyPort USB est adressé ; l'adresse est indiquée au rythme de 2 s par le nombre de signaux lumineux.
* Écran LCD  
  La première ligne affiche les tensions d'entrée ou de sortie du signal analogique sélectionné, sous forme de valeur et, en plus, de barographe.  
  La deuxième ligne indique le numéro de canal et le sens du signal.  
  On a : **In** = signal d'entrée, **Out** = signal de sortie.
* LED IN 0...7  
  Indication d'état des entrées numériques (TOR) par LED vertes
* LED OUT 0...7  
  Indication d'état des sorties numériques (TOR) par LED jaunes
* Touches  
  La touche permet de sélectionner le canal analogique affiché à l'écran LCD.  
  Le numéro de canal **Ch** possible pour signal d'entrée est: 0…3.  
  Le numéro de canal **Ch** possible pour signal de sortie est: 0…1.  
  La touche permet de sélectionner différentes unités physiques.  
  Les unités proposées sont: V, bar, PSI, MPa, l/min, °C.  
  L'actionnement simultané des deux touches fléchées active le réglage d'adresse.

**Manipulation 1**

**Buts :**

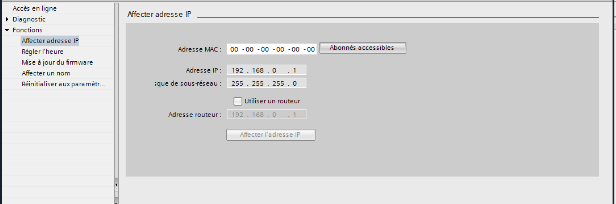
* Initiation à TIA Portal de Siemens pour la programmation de la gamme SIMATIC S7-1200.
* Exemple d’application :
* Fonction OR.
* Fonction AND.
* Fonction XOR.
* Etc…
* Utiliser les connaissances acquises pour simuler notre manipulation (Passage à niveau) avec un API réel et le logiciel EazyVeep.

1. **Initiation à TIA Portal :**

**Connexion à la CPU via le protocole TCP/IP :**

Pour programmer le SIMATIC S7-1200 à partir d’un PC, d’une PG ou d’un ordinateur portable, vous avez besoin d’une connexion TCP/IP. Pour que le PC et SIMATIC S7-1200 puissent communiquer, il est important que leurs adresses IP correspondent. Il faut tout d'abord expliquer ici comment l'adresse IP de l'ordinateur peut être paramétrée.

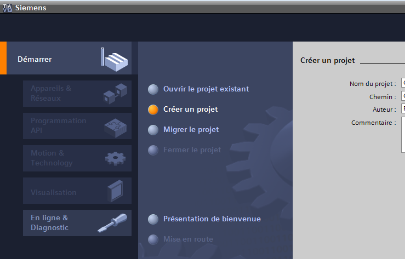
1. Appelez les **Connexions réseau** à partir du **Panneau** **de configuration** puis sélectionnez les **Propriétés** de la connexion LAN. Démarrer →Paramètres → Panneau de configuration → Connexions réseau → Connexion locale → Propriétés.
2. Sélectionnez les **Propriétés** du **Protocole Internet** **(TCP/IP).**
3. Vous pouvez ensuite paramétrer l'**Adresse IP** et le **Masque** **de sous-réseau** et les appliquer en cliquant sur **OK** →Utiliser l'adresse IP suivante → Adresse IP : 192.168.0.1 → Masque  
   de sous-réseau 255.255.255.0 → OK → Fermer.
4. Faire une double-clique sur **Totally Integrated Automation Portal** pour activer l'utilitaire. (→ TIA  
   Portal V13).
5. Sélectionner la commande de menu **En ligne et diagnostique**, puis ouvrir la **Vue du projet**.
6. Sélectionnez ensuite dans le navigateur du projet sous **Accès en ligne**, la carte réseau paramétrée précédemment. Si vous cliquez sur **Mettre à jour** **les abonnés accessibles**, vous verrez l’adresse MAC du SIMATIC S7-1200 connecté. Sélectionnez ici **En ligne et diagnostic** → Accès en ligne → … Connexion réseau.
7. Sous **Fonctions**, vous verrez l’option **Affecter adresse IP**. Entrez l’**adresse IP** et le **Masque sous-réseau**. Cliquez ensuite sur **Affecter l’adresse IP** pour que cette nouvelle adresse soit affectée au SIMATIC S7-1200. → Fonctions →Affecter l’adresse IP → Adresse IP : 192.168.0.1 → Masque de sous-réseau : 255.255.255.0 → Affecter l’adresse IP



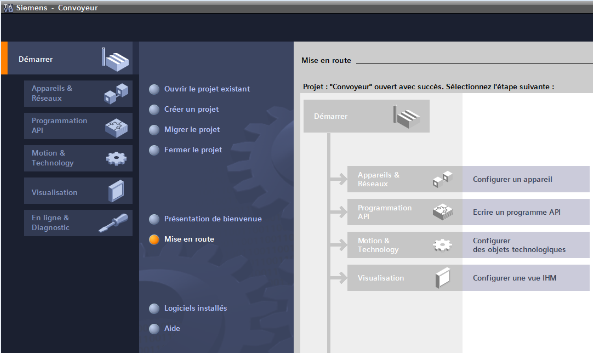
1. **Exemples d’applications :**

Les étapes ci-dessous montrent comment créer un projet pour SIMATIC S7-1200 et programmer la  
solution pour cette application.

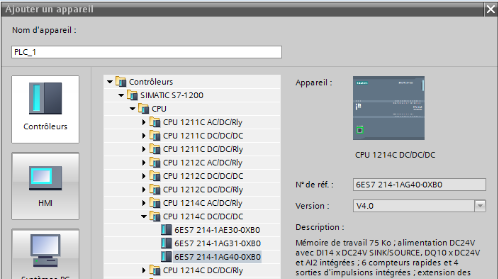
1. L’outil que nous allons utiliser est **Totally Integrated Automation Portal**, que l’on appelle ici d’un double-clique. (→ Totally Integrated Automation Portal V13)
2. Les programmes pour SIMATIC S7-1200 sont gérés sous forme de projets. Nous allons maintenant créer un nouveau projet via la vue du portail →Créer un projet → Passage à niveau → Créer.

****

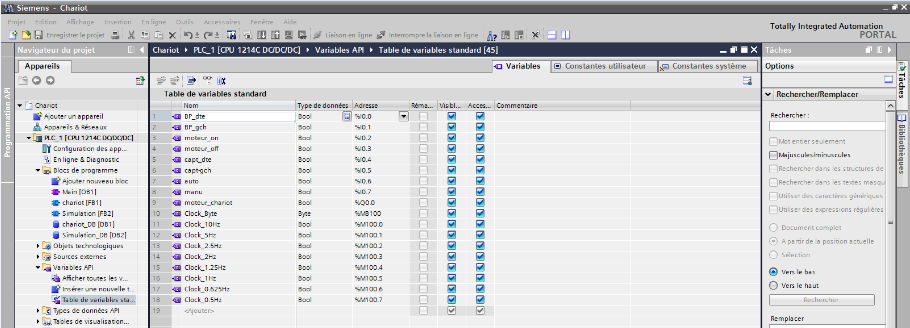
1. L'option **Mise en route** est proposée pour la configuration. En premier lieu, nous allons **configurer un appareil**. → Mise en route → Configurer un appareil.

****

1. Puis, nous allons **ajouter un appareil** avec le **nom d’appareil ‘Manipulation1’**. Dans le catalogue, choisir la **CPU 1214C DC/DC/DC** avec la référence correcte. → Ajouter un appareil → Manipulation1 → CPU 1214C → 6ES7 232-4HA40-0XB0→ Ajouter.

****

1. Le logiciel passe automatiquement vers la vue du projet avec la configuration matérielle ouverte. Ici, on peut ajouter des modules supplémentaires depuis le catalogue du matériel (fenêtre de droite) ; les adresses d’entrée/sortie peuvent être réglées dans la **vue d’ensemble des appareils**. Dans notre cas, les entrées intégrées à la CPU ont des adresses allant de %E 0.0 à %E 1.5 et les sorties intégrées des adresses allant de %A 0.0 à %A 1.1 → Vue d’ensemble des appareils → DI14/DO10 → 0…1
2. Afin que le logiciel puisse accéder par la suite à la bonne CPU, il convient de paramétrer son  
   adresse IP et le masque de sous-réseau. → Propriétés → Général → Interface PROFINET → Adresse IP : 192.168.0.1 → Masque de sous-réseau : 255.255.255.0
3. Puisque de nos jours, on programme avec des variables plutôt qu’avec des adresses absolues, on doit spécifier ici les **variables globales de l’API**. Ces variables globales de l'API sont des noms descriptifs accompagnés de commentaires qui définissent chaque entrée et sortie utilisée dans le programme. Plus tard, pendant la programmation, on pourra accéder à ces variables API globales via leurs noms. Ces variables globales peuvent être utilisées partout dans le programme, dans tous les blocs. **Manipulation1 [CPU1214C** **DC/DC/DC]** puis **Variables API**. Avec un double-clique, ouvrir la **table de variables** **standard** et saisir, comme montré ci-dessous, les noms des entrées et des sorties. → Manipulation1 [CPU1214C DC/DC/DC] → Variables API → Table de variables standard

****

**Programmation de l’API :**

Nos premiers exemples seront des portes logiques élémentaires avec utilisation dans la programmation des **API.**  
**Tableau d'affectations :**

|  |  |  |
| --- | --- | --- |
| **Symboles** | **Adresses** | **Commentaires** |
| A | %E0.0 | Entrée 1 |
| B | %E0.1 | Entrée 2 |
| C | %E0.2 | Entrée 3 |
| OR | %A0.0 | Sortie OR |
| AND | %A0.1 | Sortie AND |
| XOR | %A0.2 | Sortie XOR |

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

* Pour charger le programme entier dans la CPU, cliquer sur l'icône Charger dans l’appareil. → Manipulation1.
* Dans la boîte de dialogue suivante, sélectionnez **PN/IE** comme type pour l'interface PG/PC puis les cartes réseau installées auparavant comme interface PG/PC. Après la **Mise à jour** des abonnés accessibles, vous devez voir votre CPU avec l'adresse 192.168.0.1 et pouvoir la sélectionner comme appareil cible. Cliquez ensuite sur → Type de l'interface PG/PC:  
  PN/IE → Interface PG/PC : …… → Mise à jour.
* Si le chargement s'est correctement déroulé, le résultat s'affiche dans une nouvelle fenêtre. Cliquez ensuite sur **Terminer**. → Terminer.
* Démarrer la CPU en cliquant sur l'icône
* Cliquez sur l’icône Activer/désactiver visualisation du programme. Cette commande permet de surveiller l’état des variables d'entrée et de sortie sur le schéma pendant le test.

1. **Manipulation: Passage à niveau**
2. **Introduction :**

A un passage à niveau un train peut venir de chaque direction. Lorsqu’un train arrive, les barrières de sécurité doivent être descendues et les feux de voyantisation routière doivent passer au rouge. Lorsque le train a quitté le passage à niveau et qu’aucun train n’arrive en sens inverse, la barrière remonte et les feux de signalisation sont commutés au blanc.

**Déclarations des Entrées/Sorties :**

|  |  |  |
| --- | --- | --- |
| **Symboles** | **Adresse** | **Commentaire** |
| Train\_dte | %E0.2 | Train de droite |
| Train\_gch | %E0.3 | Train de gauche |
| Bar\_dte\_haut | %E0.4 | Barrière droite en haut |
| Bar\_gch\_haut | %E0.6 | Barrière gauche en haut |
| Bars\_descendants | %A0.0 | Descente des barrières |
| Signal\_blanc | %A0.1 | Feux de signalisation blanc active |
| Signal\_rouge | %A0.2 | Feux de signalisation rouge active |

1. **Travaille demandée :**
2. Programmer le cycle décrit en utilisant le langage de contact(CONT).
3. Transférer l’application dans l’automate SIMATIC S7-1200 1214C et la simuler avec l’interface Eazyport USB.

**Manipulation 2**

**Buts :**

* Utiliser les connaissances précédentes pour réaliser un exemple plus réaliste dans le contexte industriel.
* Introduction aux Interfaces Homme Machine(HMI).

1. **Contrôle-commande avec WinCC :**  
   **Description du système**  
   Sachant que les processus 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 processus (machine/système). Le contrôle effectif du processus 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 processus. Au Runtime, le projet est exécuté en mode processus.

WinCC gère les tâches suivantes :

**• Représentation du processus**  
Le processus est représenté sur le pupitre opérateur. Si, par exemple, un changement intervient dans le processus, l’affichage est mis à jour sur le pupitre opérateur.  
**• Commande du processus**  
L’opérateur peut commander le processus 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 processus, une alarme se déclenche automatiquement. Par exemple, quand une limite fixée est dépassée.  
**• Archivage des valeurs de processus et des alarmes**  
Le système IHM peut archiver des alarmes et des valeurs de processus. Cela vous permet de documenter les caractéristiques du processus 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 processus sous forme de protocole. Vous pouvez ainsi afficher les données de production à chaque changement d'équipe.  
**• Gestion des paramètres du processus et des machines**  
Le système IHM peut enregistrer les paramètres de processus 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.

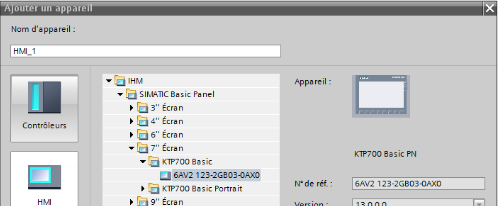
1. **Configuration:**  
   La visualisation du processus est créée sous le logiciel de configuration STEP7 V13 à l'aide de la version intégrée de WinCC. Les valeurs du processus sont représentées via des écrans et des objets sur l’écran. Les valeurs par défaut peuvent être transférées à l'automate en utilisant les éléments de commande. La communication entre le pupitre opérateur et la machine ou le processus 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 processus est enregistrée et chargée dans le Panel KTP700 Basic color PN.**  
**Après l’allumage du Panel, la commande du convoyeur peut être contrôlée et commandée.**

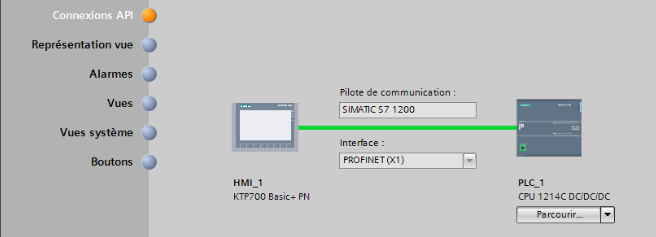
1. **Insertion d’un Panel KTP700 PN dans un projet :**  
   Là, sous une même interface, les éléments tels que le contrôleur, la visualisation et la mise en réseau de la solution d’automatisation sont créés, paramétrés et programmés. Les outils en ligne sont disponibles pour les diagnostics d’erreur.

Pour créer un nouveau Panel dans le projet :

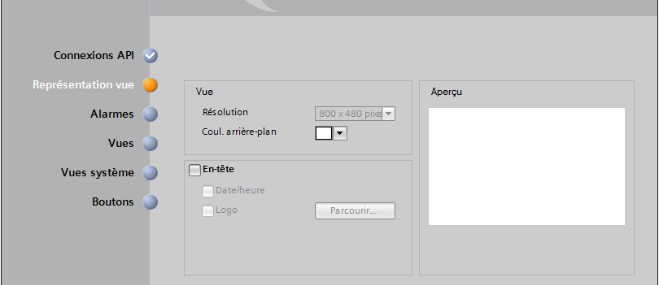
* Ouvrir la fenêtre de sélection en double-cliquant sur **(Ajouter un appareil)**. Sous SIMATIC HMI, 7“ Display, choisir **KTP700 Basic PN**. Cocher la case **(Ouvrir la vue des appareils)**. Cliquez sur le bouton **OK**.



* Sous Select PLC (sélectionner API), choisir **(commande\_convoyeur)**. Cliquer ensuite sur le bouton **(Suivant)**.

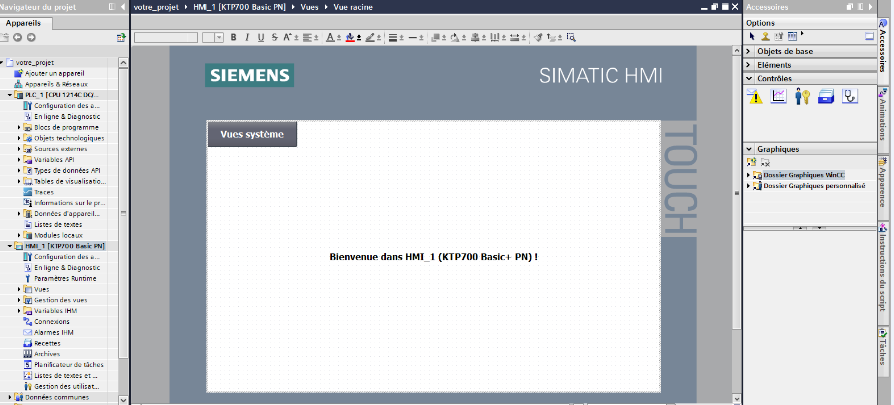


* Sous (Représentation vue), modifier la couleur d'arrière-plan et choisir **blanc**, puis désactiver la case à cocher **(En-tête)**. Cliquer ensuite sur le bouton **(Suivant)**.



* Désactiver toutes les cases à cocher sous (Alarmes). Cliquer ensuite sur le bouton **(Suivant)**.
* Sous (Navigation écran), il est possible de créer une arborescence de menus. Pour notre exemple, la vue (Racine) est suffisante. Cliquer ensuite sur le bouton **(Suivant)**.
* Sous (écrans système) activer **(Modes opératoires)** et **(Arrêter Runtime)**. Cliquer ensuite sur le bouton **(Suivant)**.
* Enfin, il est possible de placer des touches systèmes préconfigurées. Désactiver toutes les cases. Cliquer ensuite sur le bouton **(Terminer)**.

L’interface de WinCC est maintenant ouverte avec la vue racine:



1. **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)**.

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.

1. **Connexions aux automates S7 :**

Pour pouvoir assurer l'affichage et la commande des valeurs de processus 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 **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 **Abonnés accessibles**  
Cliquer sur le bouton **Afficher**.

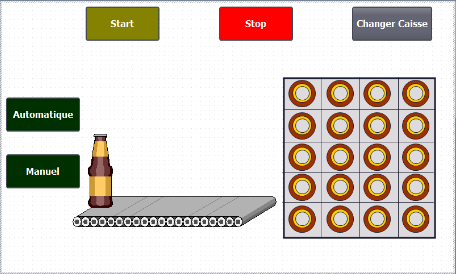
1. **Affecter l’adresse IP :**

Après avoir entré l’adresse MAC, l’adresse IP peut être affectée sous **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**.

1. **Manipulation: Commande d'un convoyeur avec compteur et multi-instance**
2. **Introduction :**

Pour visualiser notre processus avec WinCC, un compteur et une multi-instance doivent être ajoutés. 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’.



**Déclarations des Entrées/Sorties :**

|  |  |  |
| --- | --- | --- |
| **Symbole** | **Adresse** | **Commentaire** |
|  |  |  |
| S1 | %E0.0 | Bouton mode manuel, NO |
| S2 | %E0.1 | Bouton mode automatique, NO |
| S3 | %E0.2 | Bouton Marche, NO |
| S4 | %E0.3 | Bouton Arrêt, NF |
| S5 | %E0.4 | Bouton S5 NO Réinitialiser le compteur / nouvelle caisse |
| B0 | %E0.5 | Capteur B0 NO Compteur bouteilles |
| M1 | %A0.0 | Moteur du convoyeur M1 |

1. **Travaille demandée :**
2. Programmer l’application décrite en utilisant le langage de contact(CONT).
3. Transférer l’application dans l’automate SIMATIC S7-1200 1214C et la simuler avec l’IHM.

**NB :**

**On va utiliser un bloc fonctionnelle FB pour le ‘convoyeur’ et un autre pour la ‘simulation’**

**Manipulation 3**

**But**

* Mise en service du variateur SIMATIC G110 CPM110.
* Controller et commander le système moteur-variateur via l’automate S7-1200 et l’IHM.

1. **Module SIMATIC G110 CPM110 :**
2. **Aperçu :**

SINAMICS G110 est un variateur de fréquence dont la fonctionnalité de base couvre les besoins d’entraînement à vitesse variable dans un large champ d’applications industrielles. Le variateur compact SINAMICS G110 fonctionne en loi *U/f* sur les réseaux monophasés 200 V à 240 V.  
Il constitue la solution «budget» idéale en entrée de gamme de la famille de variateurs SINAMICS.

1. **Domaines d’application :**

SINAMICS G110 convient particulièrement aux applications de pompage et de ventilation, aux applications d’entraînement dans une grande diversité de secteurs tels que l’agroalimentaire, le textile, l’emballage. Il convient également aux applications de manutention, de manœuvre de portails d’usine et de garage et comme entraînement universel pour les supports publicitaires mobiles.

1. **Construction :**

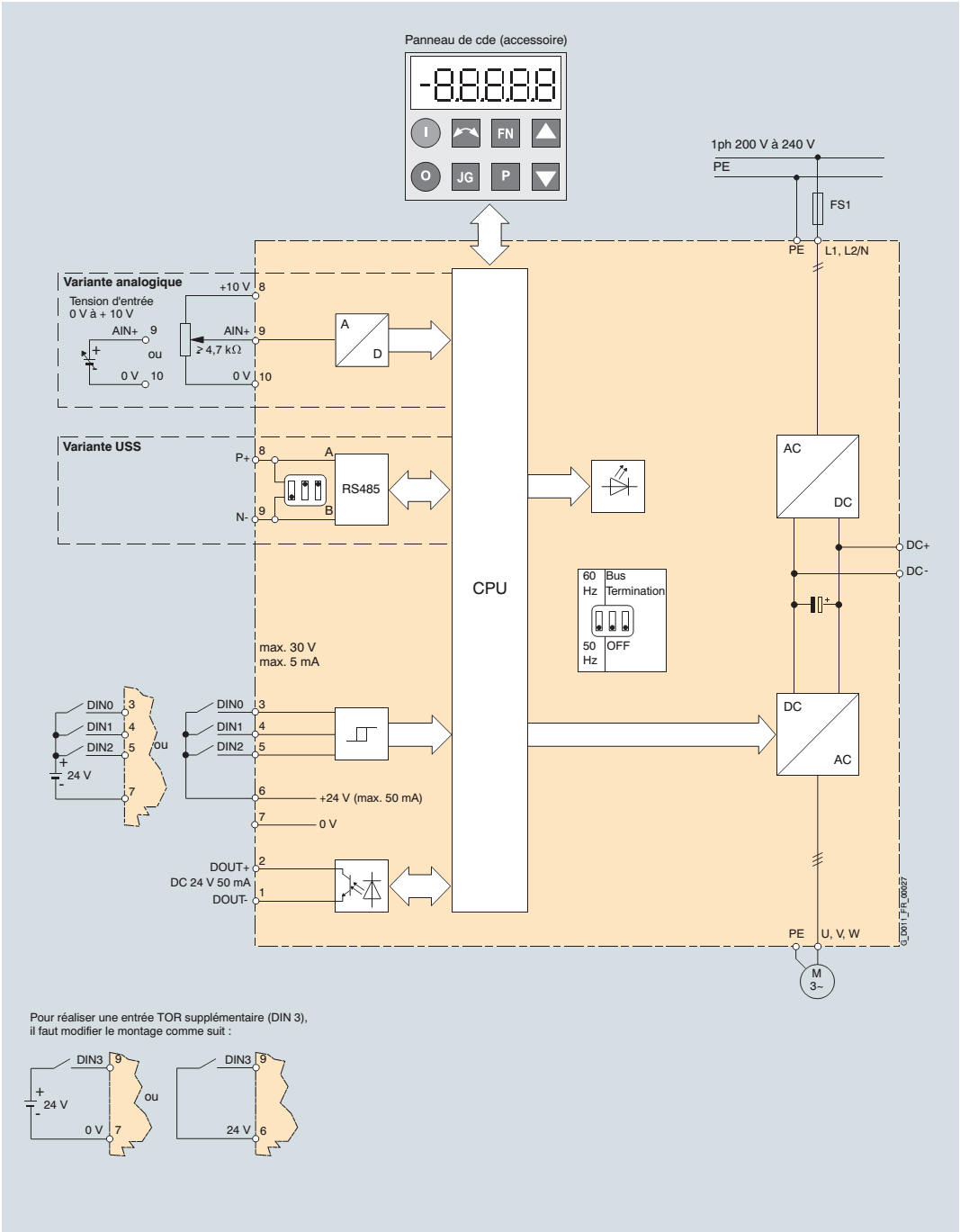
Les modèles encastrables de la série SINAMICS G110 comportent le module de commande et le module puissance et confèrent au variateur de la version CPM 110 (Controlled Power Module = module de puissance contrôlé) sa conception compacte et performante. Ils utilisent la technologie IGBT la plus récente et une commande numérique à microprocesseur.  
La famille des variateurs SINAMICS G110 comporte les versions et exécutions suivantes :

* La version analogique est disponible dans les exécutions suivantes:  
  - sans filtre CEM, avec radiateur.  
  - filtre CEM intégré de classe A/B, avec radiateur.  
  - sans filtre CEM, avec radiateur plat (taille A seulement).  
  - filtre CEM intégré de classe B, avec radiateur plat (taille A seulement).
* La version USS (RS485) est disponible dans les exécutions suivantes :   
  - sans filtre CEM, avec radiateur.  
  - filtre CEM intégré de classe A/B, avec radiateur.  
  - sans filtre CEM, avec radiateur plat (taille A seulement).  
  - filtre CEM intégré de classe B, avec radiateur plat (taille A seulement).

**NB :**

**Nous allons utiliser la version analogique.**

1. **Schéma bloc :**



1. **BOP : touches et fonctions**



|  |  |  |
| --- | --- | --- |
| **Champ/touche** | **Fonction** | **résultats** |
|  | Visu d’état | L'afficheur LCD indique les réglages actuels du variateur. |
|  | Mise en marche du moteur | Cette touche permet de démarrer le variateur. Par défaut, elle est désactivée. Pour l'activer : P0700 = 1 ou P0719 = 10 ... 15 |
|  | Mise à l’arrêt du moteur | ARRET1: L'actionnement de cette touche entraîne l'arrêt du moteur selon la rampe paramétrée. Par défaut, cette touche est désactivée. Pour l'activer : P0700 = 1 ou P0719 = 10 ... 15 ARRET2: Le double actionnement de cette touche (ou si elle est maintenue enfoncée) entraîne l'arrêt du moteur par un ralentissement naturel. Cette fonction est toujours activée. |
|  | Inversion du sens de marche | Cette touche inverse le sens de rotation du moteur. L'inversion est indiquée par le signe moins (-) ou par un point décimal clignotant. Par défaut, cette touche est désactivée. Pour l'activer : P0700 = 1 ou P0719 = 10 ... 15. |
|  | Marche par à-coups du moteur | A l'état „prêt à l'enclenchement“ du variateur, l'actionnement de cette touche provoque la mise en marche du moteur et sa rotation à la fréquence de marche par à-coups prédéfinie. Le moteur s'arrête dès que la touche est relâchée. Cette touche est inopérante lorsque le moteur est en marche. |
|  | fonctions | Cette touche peut être utilisée pour l'affichage d'informations supplémentaires. Maintenue enfoncée pendant deux secondes pendant le fonctionnement du variateur, cette touche permet de visualiser les informations suivantes indépendamment du paramètre actuellement sélectionné : 1. Tension du circuit intermédiaire (symbolisé par un d - unité V). 2. Fréquence de sortie (Hz) 3. Tension de sortie (symbolisé par un o - unité V) 4. La valeur sélectionnée sous P0005 (si P0005 est réglé pour afficher une des grandeurs ci-dessus (1 - 3), celle-ci ne sera pas affichée). L'actionnement répété a pour effet de faire défiler l'affichage des grandeurs ci-dessus dans l'ordre indiqué. **Fonction de saut** Au départ de chaque paramètre (rxxxx ou Pxxxx), une pression brève sur la touche Fn provoque le saut immédiat à r0000. Vous pouvez alors modifier un autre paramètre. Après le retour à r0000, un nouvel actionnement de la touche Fn provoque le retour au point de départ. **Acquittement** En cas de messages d'alarme et d'erreur, ceux-ci peuvent être acquittés en actionnant la touche Fn. |
|  | Accès aux paramètres | Cette touche permet d'accéder aux paramètres. |
|  | Incrémenter une valeur | Cette touche augmente la valeur affichée. |
|  | Décrémenter une valeur | Cette touche diminuer la valeur affichée. |

1. **Mise en service du module :**
2. **Mise en service rapide :**

La mise en service rapide a pour effet d'adapter le variateur au moteur et de régler certains paramètres technologiques importants. La mise en service rapide est inutile si les caractéristiques assignées du moteur (moteur Siemens 1LA à 4 pôles, couplage étoile, spécif. variateur) contenues dans le variateur coïncident avec celles inscrites sur la plaque signalétique du moteur.

**Départ**

|  |  |
| --- | --- |
| **P0010** | **Filtre des paramètres de mise en service \***  0 Prêt  1 Mise en service rapide  30 Réglages d'usine  **REMARQUE**  Le variateur peut être mis en service très rapidement et facilement en réglant  P0010 = 1 |
| **P0100** | **Europe / Amérique du Nord**  (fréquence du secteur)  0 Europe [kW], 50 Hz  1 Amérique du Nord [hp], 60 Hz  2 Amérique du Nord [kW], 60 Hz  **REMARQUE**  Avec le réglage P0100 = 0 ou 1, la position de  l'interrupteur DIP 50/60 Hz détermine la valeur de  P0100 |
| **P0304** | **Tension assignée du moteur** (à relever sur la plaque signalétique en V)  Tension nominale du moteur indiquée sur la plaque signalétique. Le diagramme suivant montre une plaque signalétique typique avec la position des caractéristiques essentielles du moteur. |
| **P0305** | **Courant assignée moteur**  (à relever sur la plaque signalétique en A) |
| **P0307** | **Puissance assignée moteur**  (à relever sur la plaque signalétique en kW/hp)  Si P0100 = 0 ou 2, la valeur saisie est en kW, si P0100 = 1, elle est en hp. |
| **P0308** | **Cos Phi assigné du moteur**  (à relever sur la plaque signalétique en cos Phi)  Le réglage 0 entraîne un calcul interne de la valeur.  P0100 = 1: P0308 sans objet, aucune saisie nécessaire. |
| **P0309** | **Rendement assigné du moteur**  (à relever sur la plaque signalétique en %)  Le réglage 0 entraîne un calcul interne de la valeur.  P0100 = 0, 2: P0309 sans objet, aucune saisie nécessaire. |
| **P0310** | **Fréquence moteur assignée**  (à relever sur la plaque signalétique en Hz)  Le nombre de paires de pôles est recalculé automatiquement dès que le  paramètre est modifié. |
| **P0311** | **Vitesse assignée moteur**  (à relever sur la plaque signalétique en tr/min)  **REMARQUE**  La saisie est obligatoire en cas de compensation du glissement. |
| **P0335** | **Refroidissement du moteur**  (Sélectionne le système de refroidissement moteur.)  0 Auto ventilé  1 Moto ventilé |
| **P0640** | **Facteur de surcharge du moteur,%**  Définit la limitation du courant de surcharge du moteur en [%] par rapport à  P0305. |
| **P0700** | **Sélection source de commande**  (voir chapitre 6.2.2 "Sélection source de commande")  0 Réglages usine par défaut  1 BOP (clavier)  2 Borne  5 USS |
| **P1000** | **Sélection consigne de fréquence**  (voir chapitre 6.2.5 "Sélection consigne de fréquence")  1 Consigne MOP  2 Consigne analogique  3 Fréquence fixe  5 USS |
| **P1080** | **Fréquence min.**  (fréquence minimale du moteur en Hz)  Définition de la fréquence minimale de fonctionnement du moteur indépendamment de la consigne de fréquence. La valeur entrée ici est valable pour les deux sens de rotation. |
| **P1082** | **Fréquence max.**  (fréquence maximale du moteur en Hz)  Définition de la fréquence maximale sur laquelle le moteur sera limité indépendamment de la consigne de fréquence. La valeur entrée ici est valable pour les deux sens de rotation. |
| **P1120** | **Temps de montée**  (temps d'accélération en s)  Temps nécessaire au moteur pour accélérer de 0 à la fréquence maximale du moteur (P1082) en l'absence de lissage. |
| **P1121** | **Temps de descente**  (temps de décélération en s)  Temps nécessaire au moteur pour décélérer de la fréquence maximale du moteur  (P1082) à 0 en l'absence de lissage. |
| **P1135** | **Temps de descente ARRET3**  (temps de descente pour arrêt rapide en s)  Définit le temps de descente de la fréquence maximum jusqu'à 0 pour l'ordre ARRET3. |
| **P1300** | **Mode de commande**  (type de régulation souhaité)  0 U/f avec caract. linéaire  2 U/f ave caract. parabolique  3 U/f avec caract. programmable |
| **P3900** | **Fin du mode mise en service rapide**  (démarrage du calcul du moteur)  0= Pas de mise en service rapide (pas de calcul des paramètres du moteur....).  1= Fin de la mise en service rapide avec réinitialisation sur réglage usine de tous les autres réglages (recommandée)  2= Fin de la mise en service rapide avec réinitialisation sur réglage usine des réglages E/S.  3= Fin de la mise en service rapide sans réinitialisation de tous les autres réglages.  **REMARQUE**  Si P3900 = 1, 2, 3 o P0340 est réglé en interne sur 1 et les données correspondantes sont calculées (voir liste des paramètres P0340) |

1. **Copie du jeu de paramètres avec le BOP:**

Un jeu de paramètres peut être lu dans un SINAMICS G110 et transféré, puis sauvegardé dans le BOP (uploader). Par la suite, le jeu de paramètres peut être chargé dans un autre SINAMICS G110 (downloader). Pour copier un jeu de paramètres d'un SINAMICS G110 sur un autre, utilisez la procédure suivante :   
**Uploader (SINAMICS G110 o BOP):**

1. Enfichez le BOP sur le SINAMICS G110, dont vous voulez copier les  
   paramètres.
2. peut Assurez-vous que le variateur être arrêté.
3. Arrêtez le variateur.
4. Réglez P0003=3.
5. Réglez P0010 = 30 (mode copie).
6. Réglez le paramètre P0802 = 1 pour commencer le transfert des paramètres du variateur dans le BOP.
7. Pendant le transfert (upload), le message "busy" est affiché.
8. Pendant le transfert, ni le variateur, ni le BOP ne réagissent sur une quelconque entrée.
9. Lorsque le transfert est terminé avec succès, le BOP revient à l'affichage normal ; ensuite, le variateur passe automatiquement à l'état "Prêt".
10. Si le transfert n'a pas été terminé avec succès :
11. recommencez le transfert (upload) ou effectuez une "réinitialisation sur les réglages usine".
12. Maintenant, le BOP peut être retiré du variateur.

**Downloader (BOP o SINAMICS G110):**

1. Enfichez le BOP sur le SINAMICS G110 sur lequel le jeu de paramètres doit être chargé.
2. Assurez-vous que la tension réseau est appliquée au variateur.
3. Réglez P0003 = 3.
4. Réglez P0010 = 30 (mode copie).
5. Réglez P0803 = 1 pour commencer le chargement à partir du BOP vers le SINAMICS G110.
6. Pendant le chargement, le message "busy" est affiché.
7. Pendant le chargement (download), ni le variateur ni le BOP ne réagissent sur une entrée quelconque.
8. Lorsque le chargement est terminé avec succès, le BOP revient à l'affichage normal ; ensuite, le variateur passe automatiquement à l'état "Prêt".
9. Si le chargement (download) n'a pas été terminé avec succès : recommencez le chargement ou effectuez une "réinitialisation sur les réglages usine".
10. Maintenant, le BOP peut être retiré du variateur.
11. **Moteurs triphasés à induction :**

Le moteur triphasé à induction, également dit asynchrone, est de nos jours le type de moteur le plus communément utilisé dans les applications industrielles. Le moteur à cage, en particulier, est certainement le plus employé parmi les moteurs électriques dans ces types d’application.

1. **Principe de fonctionnement :**

La section électrique du moteur triphasé à inductionse compose, comme le montre la **Figure**, d’un stator fixe, d’un enroulement triphasé alimenté par l’alimentation triphaséeet d’un rotor*.* Il n’existe aucune connexion électrique entre le stator et le rotor. Les courants dans le rotor sont induits à travers l’entrefer du côté du stator. Aussi bien le stator que le rotor sont réalisés en tôle  
extrêmement magnétique en vue de minimiser le courant de Foucault et les pertes d’hystérésis.

1. **Le stator :**

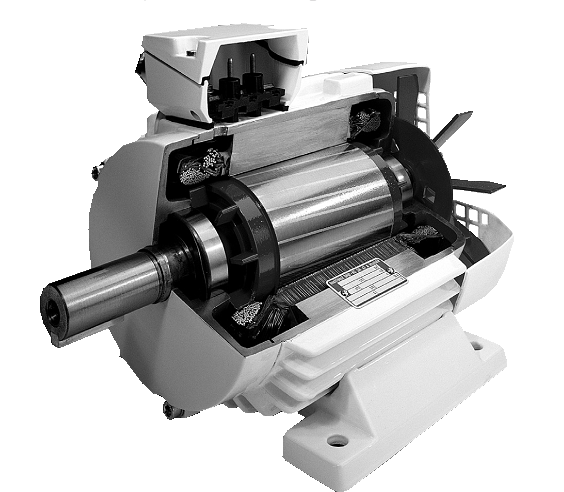
Le stator comporte trois enroulements individuels qui se chevauchent et sont décalés par un angle électrique de 120°. Lorsqu’il est relié à l’alimentation, le courant entrant magnétise ce stator en premier lieu. Ce courant (courant magnétisant)génère un champ tournantqui tourne à une vitesse dite synchrone appelée vitesse**ns**.

1. **Le rotor :**

Dans les moteurs à induction à cage, le rotor consiste en une série ou paquet de tôles cylindriques à encoches avec des barres en aluminium unies à l’avant par des anneaux de sorte à former une cage fermée. Le rotor des moteurs triphasés à induction est souvent dit ancre en raison de la forme des rotors appliqués aux tous premiers appareils électriques. Dans un appareil électrique, l’enroulement du rotor ou ancre est induit par le champ magnétique à la différence des moteurs à induction triphasés où cette fonction est remplie par le champ tournant statorique.

1. **Glissement :**

On définit glissement **s** la différence entre la vitesse synchrone ns et la vitesse **n** de fonctionnement nominal n qui est normalement exprimée en pourcentage. Selon les moteurs, ce glissement nominal oscille entre 3% et 10%. Le glissement est l’une des caractéristiques les plus importantes d’un moteur à induction.

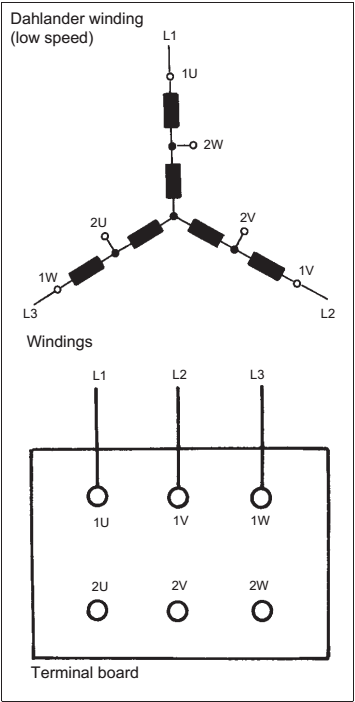


**Figure: Un moteur triphasé à cage**

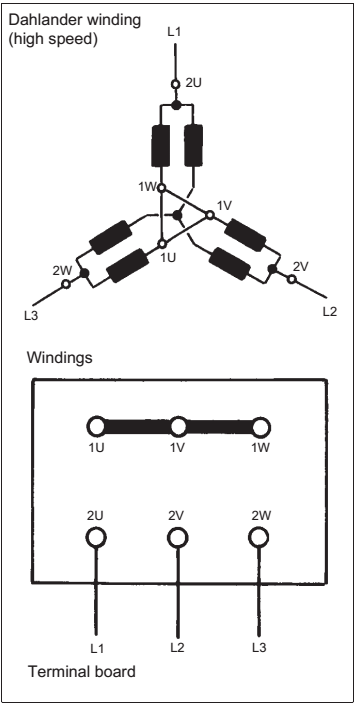
1. **Moteur à deux vitesses avec enroulements Dahlander :**

Pour l’une des vitesses, le réseau est connecté sur les 3 bornes correspondantes. Pour la seconde, celle-ci sont reliées entre elles, le réseau étant branché sur les trois autres bornes.  
L’ordre d’alimentation des phases doit être respecté pour ne pas avoir une inversion du sens de rotation au moment du passage de la petite vitesse à la grande vitesse.

1. **Couplage triangle série : petite vitesse**



1. **Couplage étoile parallèle : grande vitesse**



1. **Manipulation : contrôler un moteur avec l’IHM et le S7-1200 via un variateur**
2. **Introduction :**

On veut commander un moteur à enroulement Dahlander par une consigne analogique premièrement en boucle ouvert puis en boucle fermer. Tous les entrées/sorties de notre application Doivent être affichées sur l’IHM.

1. **Travaille demandée :**
2. Configurer le variateur convenablement à notre application.
3. Réaliser le programme STEP7 et l’interface IHM convenable.

**Manipulation 4**

**But :**

* Apprendre à configurer une liaison de transport ISO-on-TCP pour un échange de données entre deux automates SIMATIC S7-1200.
* Se familiariser avec les fonctions de base de l'environnement de développement S7-SCL.
* Prendre connaissance des outils aidant la recherche des erreurs de programmation.

1. **Langage de programmation S7-SCL :**

SCL (Structured Control Language) est un langage de programmation évolué basé sur le PASCAL et  
permettant une programmation structurée. Le langage est conforme aux normes DIN EN-61131 -3 (CEI  
61131 -3) relatives aux commandes séquentielles du langage SFC "Sequential Function Chart". En plus  
des éléments typiques à l'API, tels que les entrées, sorties, temporisations, mémentos, appels de blocs,   
S7-SCL contient également des éléments de langage de programmation évolué. Il prend en charge le concept de blocs de STEP 7. Il est ainsi possible d'utiliser un langage de programmation des blocs  
standardisé en plus des langages de programmation LIST, CONT, LOG. De ce fait, S7-SCL complète et  
élargit les logiciels de programmation STEP 7 utilisant LIST, CONT, LOG.

Il n'est pas nécessaire de créer chaque fonction, vous pouvez utiliser des blocs préprogrammés, comme  
les fonctions systèmes et les blocs fonctionnels système qui se trouvent dans le système d'exploitation  
de l'unité centrale.

Vous pouvez mixer des blocs programmés avec S7-SCL avec des blocs CONT, LIST et LOG. Cela  
signifie qu'un bloc programmé avec S7-SCL peut appeler un autre bloc programmé en langage CONT,   
LIST ou LOG. Inversement, les blocs S7-SCL peuvent également être appelés dans des programmes  
écrits en CONT, LIST et LOG.

Les fonctions de test de S7-SCL permettent de rechercher les erreurs logiques de programmation dans  
une compilation correcte.

1. **L'environnement de développement S7-SCL :**

Afin d'utiliser au mieux S7-SCL, nous avons créé un environnement de développement spécialement  
conçu pour répondre à la fois aux particularités de S7-SCL et à celles de STEP 7. Cet environnement  
est composé d'un éditeur/compilateur et d'un débogueur.

**Editeur / compilateur**  
L'éditeur S7-SCL est un éditeur de texte permettant de traiter tout type de texte. Il sert principalement à créer et éditer les blocs utilisés dans les programmes STEP 7. Le contrôle de syntaxe intervient durant la saisie, ce qui réduit les risques d'erreur de programmation. Les erreurs de syntaxe sont présentées sous différentes couleurs.

L'éditeur offre les possibilités suivantes :

* Programmation d'un bloc S7 dans le langage S7-SCL.
* Ajout d'éléments de langage et d'appels de bloc par simple glissé-déposer.
* Contrôle de syntaxe directement en cours de programmation.
* Configuration de l'éditeur en fonction des besoins spécifiques, p.ex. Colorisation des différents éléments de langage en fonction de leur syntaxe.
* Vérification du bloc terminé par compilation.
* Affichage des erreurs et avertissements survenus pendant la compilation.
* Localisation de l'erreur dans le bloc, éventuellement avec description de l'erreur et proposition de correction.

**Débogueur**  
Le débogueur S7-SCL permet de contrôler l'exécution d'un programme dans l'AS et de détecter ainsi les erreurs de logique.

S7-SCL dispose à cet effet de deux types de test :

* Visualisation par étape.
* Visualisation en continue.

En mode "visualisation par étape", le programme est exécuté dans son déroulement logique. Vous  
pouvez exécuter l'algorithme instruction par instruction et observer la modification des variables dans une fenêtre de résultats

La "visualisation en continu" permet de tester un groupe d'instructions dans un bloc. Pendant le test, les valeurs des variables et des paramètres s'affichent chronologiquement et sont actualisées, autant que possible, de manière cyclique.

1. **Exemple d’application : commande d’une presse**

Notre application va consister à programmer une commande de presse.

Une presse avec un capot de protection doit être activée avec un bouton START uniquement si la  
protection est fermée. Cette condition est surveillée à l'aide d'un capteur Protection fermée Capt1.  
Si c’est le cas, un distributeur 5/2 M0 alimentant le vérin de la presse est activé, afin que la forme  
plastique puisse ensuite être pressée.  
La presse doit se retirer de nouveau quand le bouton ARRET D’URGENCE (contact NF) est actionné,   
quand le capteur Protection fermée B1 ne répond plus, ou quand le capteur Vérin tige sortie Capt2 répond.

**Table d’affectation**

|  |  |  |
| --- | --- | --- |
| **Symbole** | **Adresse** | **Commentaire** |
| Start | %E0.0 | Bouton de démarrage S3 (NO) |
| Capt1 | %E0.1 | Capteur Protection fermée (NO) |
| Capt2 | %E0.2 | Capteur vérin A tige sortie (NO) |
| Arrêt\_uegence | %E0.3 | Bouton ARRET D’URGENCE (NF) |
| M0 | %A0.0 | Sortir tige du vérin A |

1. **Communication entre deux SIMATIC S7-1200:**

Dans ce module, l’étudiant va apprendre à configurer une liaison de transport ISO-on-TCP pour un  
échange de données entre deux automates SIMATIC S7-1200, et à programmer les échanges de  
données dans le programme de contrôle en utilisant les blocs de communication TSEND\_C et TRCV\_C.

1. **Application à réaliser :**

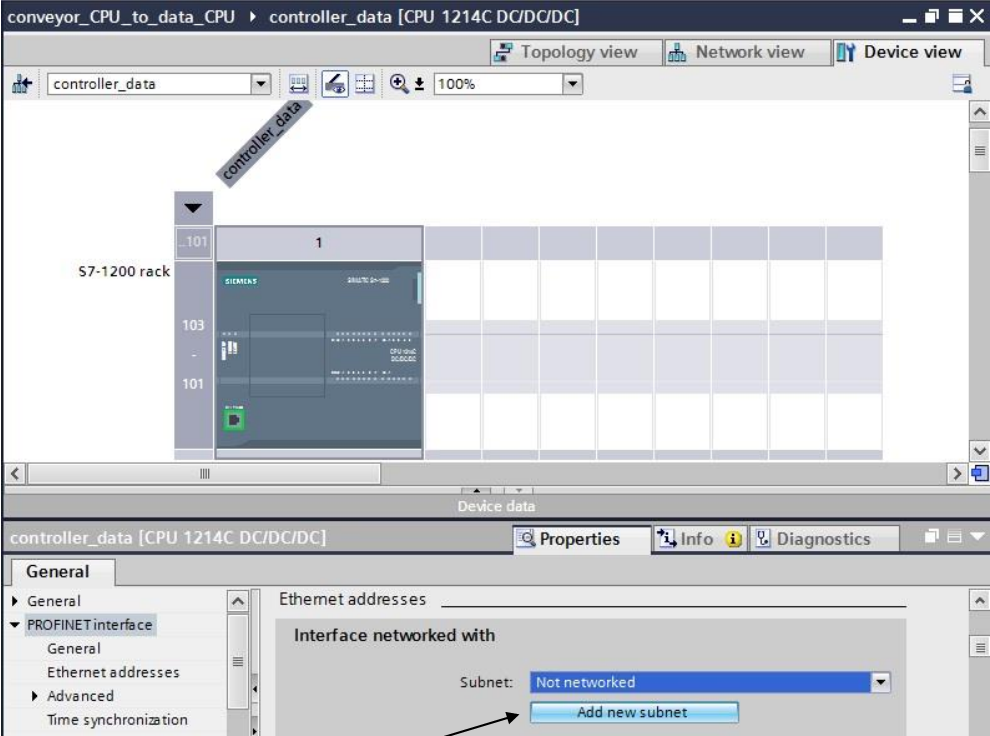
Les états des signaux d’entrée doivent être transmis à une deuxième CPU pour la suite du traitement et être visualisés ici.

Les entrées de la première CPU sont dans un premier temps écrites dans un tampon d’envoi.  
Ensuite, le bloc émetteur TSEND\_C envoie le contenu du tampon d’envoi dans la seconde CPU au  
moyen d’une liaison ISO-on-TCP.

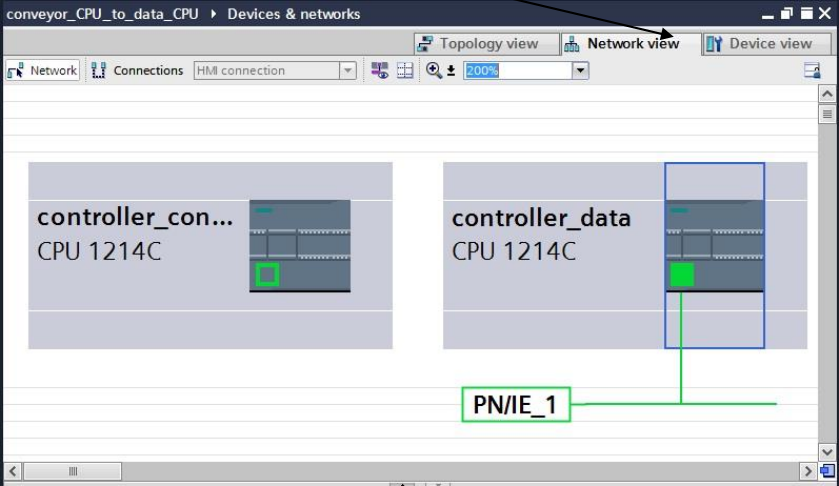
Dans la deuxième CPU, les données reçues sont écrites dans le tampon d’entrée via le bloc récepteur TRCV\_C. Le contenu du tampon de réception est ensuite visualisé sur les sorties de la CPU.

1. **Mise en réseau des automates :**

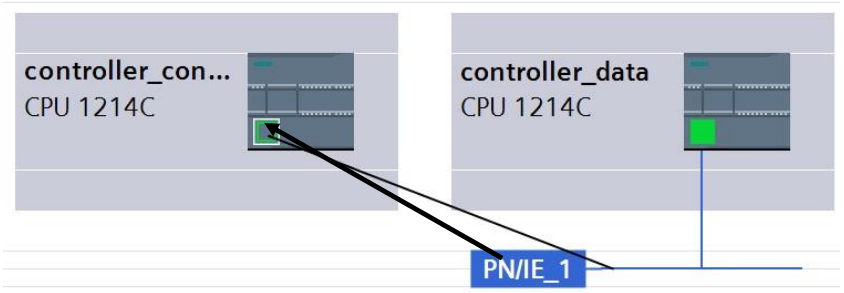
* Ajouter une deuxième CPU.
* Dans la fenêtre en bas de la vue des appareils, sous l’onglet **Propriétés**, sélectionner **Interface PROFINET**.



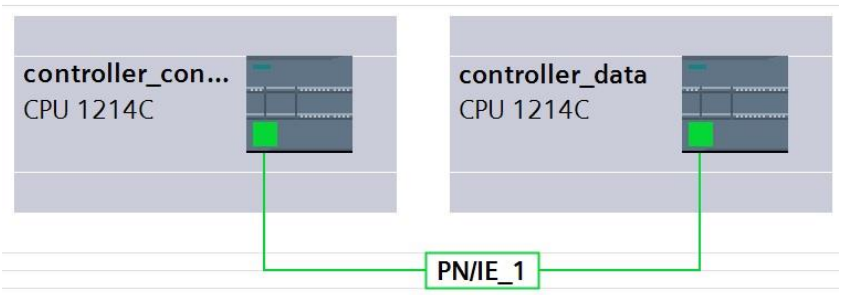
* Cliquez ensuite sur **Ajouter nouveau sous-réseau**. Entrer l'adresse IP **192.168.0.5** et le masque de sous-réseau **255.255.255.0**
* Passez dans la vue de réseau.



* Faire glissez la ligne du réseau **PN/IE\_1** vers le carré vert de Commande\_presse.



* Les deux CPU sont maintenant interconnectées.



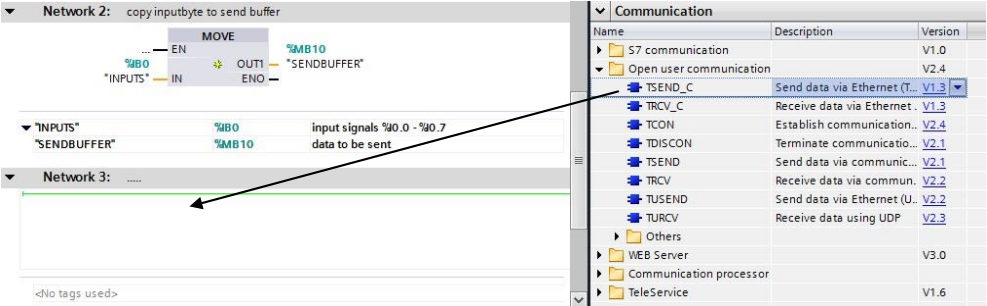
* Dans le navigateur du projet, sélectionner **Commande\_press** et cliquer sur **Charger dans l’appareil**.
* Dans le navigateur du projet, sélectionner **Commande\_donnees** et cliquer sur **Charger dans l’appareil**.

1. **Programmation des blocs de programme :**
2. **Programme pour la Commande presse :**

Nous allons maintenant appeler et paramétrer les blocs de programme nécessaires à la communication entre les deux CPU.  
En appelant les blocs de communication TSEND\_C et TRCV\_C, une connexion entre les deux CPU est  
automatiquement établie. La connexion est créée, établie et gérée automatiquement jusqu’à ce qu'une opération ou une CPU mise sur STOP rompe la liaison.

**Programmation**

* Le bloc d’émission TSEND\_C est appelé dans le bloc OB1. Dans le réseau 2, créer l’instruction MOVE. Glisser-déposer le bloc TSEND\_C dans le réseau 3.



* Dans la fenêtre suivante, confirmer l’ajout du bloc suggéré. Cliquer sur **Instance unique**, puis sur **OK**.

Le bloc de données est généré et incorporé automatiquement.  
Sous Propriétés, sélectionner les paramètres de connexion. Sous Paramètres de connexion, choisir le partenaire et renseigner les champs en couleur. Pour chaque automate, choisir un nouveau bloc de données et saisir directement **1** dans chaque champ ID de connexion.

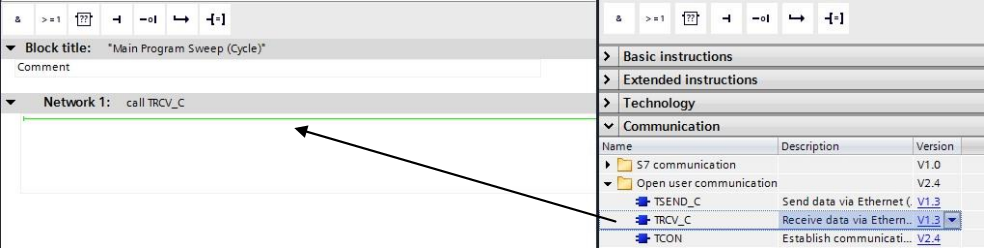
La tâche d'émission (**REQ**) du bloc TSEND\_C doit être cyclique, c’est la raison du lien avec les  
paramètres de sortie (**DONE**).

Agrandir la taille du bloc et entrer les paramètres du bloc soit directement, soit dans la fenêtre des  
propriétés.

La programmation de la commande de presse est terminée.

1. **Programme pour la Commande\_donnees :**

* Le bloc d’émission TRCV\_C est appelé dans le bloc OB1. Ouvrir le bloc Main[OB1] de la commande de données. Glissez-déposez le bloc **TRCV\_C** dans le réseau 1.



* Dans la fenêtre suivante, confirmer l’ajout du bloc suggéré. Cliquer sur **Instance unique**, puis sur **OK**.

Le bloc de données est généré et incorporé automatiquement.  
Sous Propriétés, sélectionner les paramètres de connexion.  
Sélectionner le bloc de données **controller\_data\_Connection\_DB** déjà existant uniquement sous les  
paramètres de connexion de l'automate local, sinon un nouveau bloc de données est généré.  
Ensuite, sélectionner le partenaire de liaison et son bloc de données associé.

* Agrandir la taille du bloc et entrer les paramètres du bloc soit directement, soit dans la fenêtre des propriétés.
* Dans le réseau 2, créer l’instruction **MOVE**. La programmation de la commande des données est terminée.

1. **Diagnostic et recherche d'erreurs dans SIMATIC S7-1200 :**

**Diagnostic appareil :**

* Dans **Configuration des appareils**, sous **Diagnostic**, on obtient des informations sur les états des différents composants. (→Configuration des appareils → Diagnostic).
* Faire un clic droit sur la CPU et choisir **En ligne et diagnostic** pour consulter les détails la concernant. (→En ligne et diagnostic).
* Dans la partie droite, le panneau de commande CPU s'affiche, ainsi que le temps de cycle et la mémoire.
* Des informations générales sur la CPU sont disponibles. (→Général).
* S'il y a des informations de diagnostic, elles sont affichées dans l'état du diagnostic et dans le tampon de diagnostic. (→ Etat du diagnostic → Tampon de diagnostic).
* Le tampon de diagnostic constitue l'aide la plus importante pour les diagnostics d'erreur.  
  (→ Tampon de diagnostic).
* On dispose également d'informations sur le temps de cycle du programme traité.  
  (→ Temps de cycle).
* La capacité de la mémoire est indiquée en détail. (→ Mémoire).
* On y trouve également des informations sur l'interface de la CPU.
* Sous Fonctions, vous pouvez affecter l'adresse IP. (→ Fonctions → Affecter l'adresse IP).
* Sous Fonctions, vous pouvez régler l'heure. (→ Régler l'heure).
* Vous pouvez également restaurer les paramètres d'usine. (→ Restaurer les paramètres d'usine)
* Il est également possible de vérifier et d'affecter le nom d'appareil PROFINET.

**Comparaison hors ligne/en ligne :**

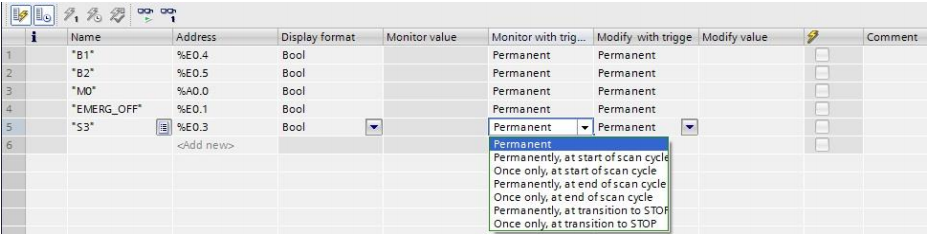
Il est souvent important de savoir si les données enregistrées correspondent aux données chargées  
dans le contrôleur.

En premier lieu, supprimer la négation d'ARRET D'URGENCE sur ET dans le bloc **Program presse** **[FC1]**.  
Enregistrer le bloc **Program presse [FC1]**, mais **ne pas** le charger dans le contrôleur.

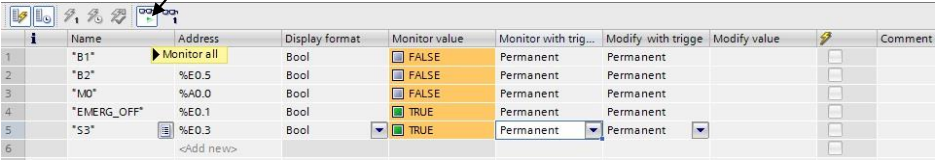
* Pour faire la comparaison, sélectionner **Commande Presse**, puis **Comparer** → **En ligne/hors ligne**. (→ Commande Presse → Comparer → En ligne/hors ligne)
* Si des différences dans les blocs sont affichées, pour lancer une comparaison détaillée. (→ Program presse → Lancer une comparaison détaillée)
* Le bloc est affiché sous sa version hors ligne et en ligne, les différences sont signalées.  
  (→ Résultat de la comparaison)

**Visualisation et forçage de variables :**

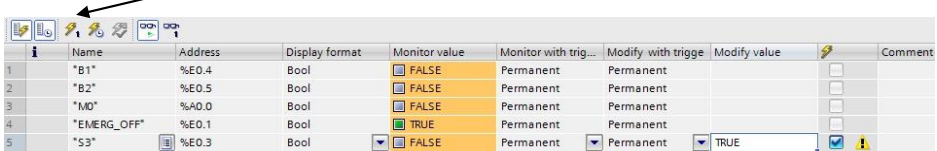
* Pour visualiser et forcer les variables, vous devez **ajouter une nouvelle table de visualisation**. (→Ajouter nouvelle table de visualisation)
* Vous pouvez saisir les variables dans la table ou sélectionner la table de variables standards et faire glisser les variables à visualiser de la vue de détail sur la table de visualisation. (→Table de variables standard)
* Les colonnes suivantes peuvent être affichées : Toutes les colonnes de forçage et Toutes les  
  colonnes du mode étendu. Choisir le point de déclenchement de la visualisation.  
  (→ Permanent)



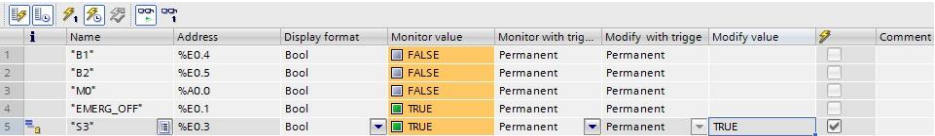
* Cliquer ensuite sur Visualiser toutes les valeurs de manière unique et immédiate ou sur  
  Visualiser tout. (→ Visualiser tout)



* Entrer les **valeurs de forçage** voulues et les activer. Cliquer ensuite sur pour **Forcer toutes**  
  **les valeurs activées de manière unique et immédiate** ou sur pour **Forcer toutes les valeurs**  
  **actives par "Forçage avec condition de déclenchement"**. (→ TRUE → Force toutes les  
  valeurs actives par "Forçage avec condition de déclenchement")



* Confirmer l'avertissement en cliquant sur **Oui**. (→ Oui)
* L'entrée est activée, bien que la borne soit hors tension.

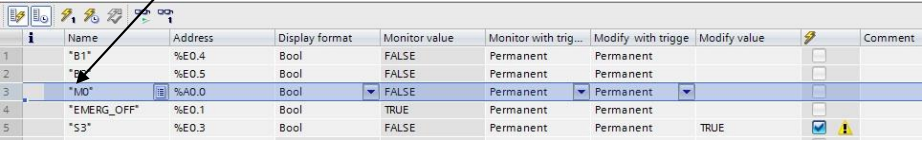


**Remarque :**  
Si la table de visualisation est fermée ou si la connexion avec l'API est perdue, toutes les variables  
forcées deviennent inactives.

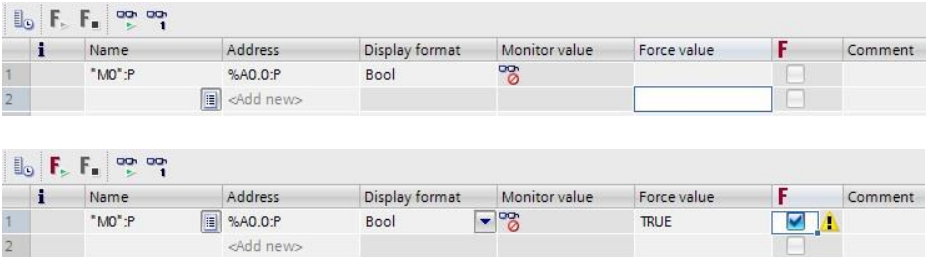
**Forçage permanent de variables :**

La fonction Forçage permanent permet de renseigner les variables avec une valeur fixe.  
Les valeurs de forçage permanent sont prédéfinies comme avec la fonction "Forçage de variables",  
mais à l'inverse, elles sont conservées après l'arrêt ou la mise hors tension de la CPU.

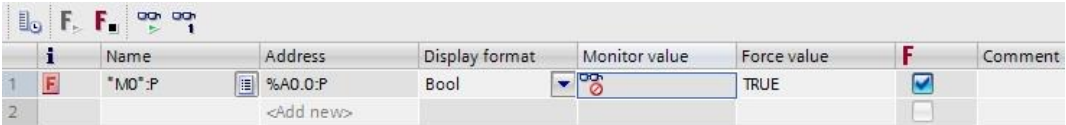
* Double-cliquer sur Force table (Table de forçage permanent) pour réaliser le forçage permanent. (→Table de forçage permanent)
* Copier l'opérande "M0" avec l'adresse %A0.0 à partir de la table de visualisation. (→ M0)



* Ajoutez l'opérande M0 dans la table de forçage permanent. Pour le forçage permanent, les opérandes sont saisis avec un accès direct à la périphérie (syntaxe : **%A0.0:P**)

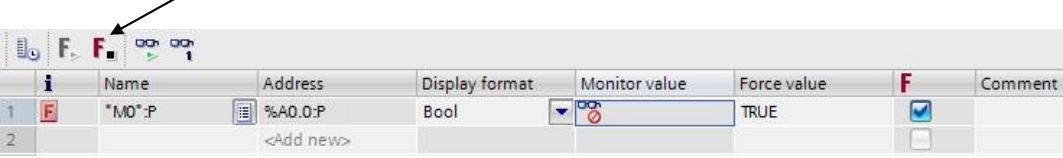


* Entrer les **valeurs** voulues et les activer. Ensuite, cliquer sur **Démarrer forçage permanent ou remplacer**. Cette action écrit la nouvelle demande forcée dans la CPU.  
  (→%A0.0:P → TRUE → → Démarrer forçage permanent ou remplacer)
* Confirmer l'avertissement en cliquant sur **Oui**. (→ Oui)



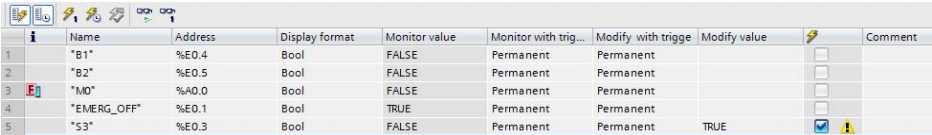
**Remarque :**  
Si la table de visualisation est fermée ou si la connexion avec l'API est perdue, **le forçage permanent**  
**reste actif** et la **DEL FRCE** jaune sur la CPU s'allume.

* Pour **quitter le forçage permanent**, cliquer sur : (→ Quitter forçage permanent)



Le forçage permanent est arrêté et la **DEL FRCE** jaune sur la CPU s'éteint.

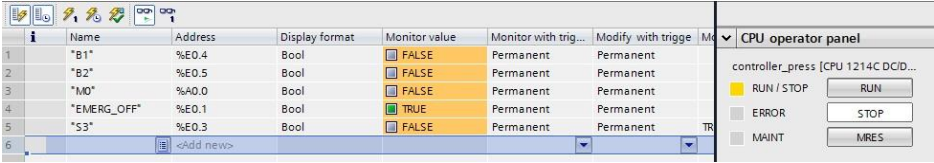
Si une demande de forçage permanent existe déjà, elle est signalée par une icône dans la table de  
visualisation.



**Débloquer les sorties de périphérie :**

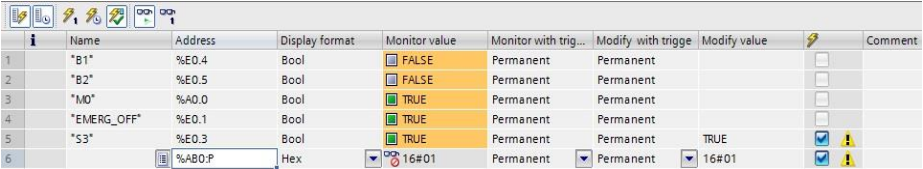
La fonction **Débloquer PA** permet de désactiver le blocage de validation des sorties (BASP) qui  
empêche l'émission de signaux de sortie lorsque la CPU est en mode STOP.  
Après le déblocage PA, les sorties commandées sont actives lorsque la CPU est en mode STOP. Les  
modules analogiques sortent leur dernière valeur.  
Pour utiliser cette fonction, saisir l'opérande dans la table de visualisation au format requis, ainsi que la valeur de forçage.

* Mettre la CPU en **mode STOP** et cliquer sur le bouton **Débloquer PA**



* Confirmez l'avertissement en cliquant sur **Oui.** (→ Oui)

**L'opérande doit être adressé en tant que périphérie au format Byte, Word ou DWord (%AB0:P,**  
**%AW0:P, %AD0:P).**



* Cliquer sur Forcer variable immédiatement.

Il n'est certes pas possible de visualiser les sorties de périphérie, mais vous pouvez en visualiser  
complément la mémoire image des sorties (MIS) dans la mémoire de la CPU (également en mode  
STOP).

