الجمهورية الجزائرية الديمقراطية الشعبية
République Algérienne démocratique et populaire

وزارة التعليم العالي و البحث العلمي
Ministère de l'enseignement supérieur et de la recherche scientifique

جامعة سعد دحلب البليدة
Université SAAD DAHLAB de BLIDA

كلية التكنولوجيا
Faculté de Technologie

قسم الإلكترونيك
Département d'Électronique

Mémoire de Master

Filière Électronique
Spécialité Automatique et Informatique Industrielle (AII)

Présenté par

DALEM NAZIM ABDELAZIZ

&

ATTIG NOUSSAIBA

Développement d'une station de supervision industrielle à base d'équipement Siemens.

Proposé par : BENNILA NOUREDDINE

Année Universitaire 2018/2019

Remerciements

Nous remercions, Dieu, le tout puissant pour nous avoir donné la foi qui nous a guidé jusqu'à la réalisation et l'aboutissement de ce projet.

Nous tenons à exprimer nos remerciements les plus distingués : à notre encadreur Mr Noureddine Bennila.

De nous avoir fait l'honneur d'assurer l'encadrement de notre travail, nous vous sommes très reconnaissants d'avoir veillé à son élaboration en ne ménageant aucunement votre temps et vos conseils.

Nous tenons à remercier vivement messieurs les membres du jury d'avoir consacrer leur temps à la lecture de ce manuscrit, et d'accepter de juger et d'évaluer ce travail.

Nous voudrions remercier nos chers parents et nos Familles qui nous ont Soutenus dans nos études.

Nous remercions les membres de jury, qui nous ont fait l'honneur de participer au jugement de ce travail.

Par ailleurs, nos remerciements vont à l'ensemble de nos professeurs, qu'ils veuillent bien trouver ici l'expression de nos sentiments les plus respectueux.

Dédicace

J'ai le grand honneur de dédier ce modeste travail :

A ma maman qui m'a soutenu et encouragé durant ces années d'étude.

Qu'elle trouve ici le témoignage de ma profonde reconnaissance.

A mes frères, mes grade parent et Ceux qui ont partagé avec moi tous les moments d'émotion lors de la réalisation de ce travail Ils m'ont chaleureusement supporté et encouragé tout au long de mon parcours.

A tous mes camarades et mes amies qui m'ont toujours encouragé et à qui je souhaite plus de succès

A tous ceux j'aime

Merci !

- Noussaiba

Dédicace

Je dédie ce travail à mon père et à ma chère mère qui m'a amené à être qui je suis aujourd'hui.

Je dédie ce travail à la famille et aux amis qui m'ont toujours soutenu.

Mon pays qui m'a tant donné.

Moi-même pour cela j'ai mis le travail et l'effort.

Et bien sûr, Dieu qui guide nos cœurs à travers les heureux et les tristes.

- Nazim Abdelaziz

ملخص: لقد أنشأنا قناة اتصال بين عدة أجهزة تحكم قابلة للبرمجة من سلسلة S7-300 باستخدام تقنيات الاتصالات PROFIBUSو MPI ثم أنشأنا نظام يتحكم في نقل البيانات اللازمة للعرض والتحكم فيها إلى برنامج WinCC .

لقد بحثنا في استخدامات ووظائف أجهزة متعددة، مثل المستشعرات الصناعية، محولات التردد، والمحركات. تركزت جميعها حول تكاملها مع وحدات التحكم، والاتصال بالبرامج المخصصة في معالجة وعرض المعلومات على HMI.

أيضًا ، كعرض لما تعلمناه ، قمنا بتطوير تطبيق تحكم في حركة عربة من خلال محول تردد من Schneider، جنبًا إلى جنب مع مستشعر ضوئي يستخدم لوظائف متعددة.

كلمات المفاتيح :مولد تردد.

Résumé : Nous avons établi un bus de communication entre plusieurs CPU S7-300 utilisant les technologies de communication PROFIBUS et MPI et transféré les données nécessaires à l'affichage et au contrôle en temps réel au logiciel WinCC.

Nous avons examiné les utilisations et les fonctionnalités de nombreux équipements tels que les capteurs industriels, les EFV et les moteurs. Tout était centré sur leur intégration aux automates programmables, la connexion à un logiciel dédié, le traitement et l'affichage d'informations sur un pupitre opérateur.

Nous avons également développé une application utilisant un automate Siemens S7-300, qui contrôle le déplacement d'un chariot via un EFV Schneider, ainsi qu'un photodétecteur utilisé pour de multiples fonctionnalités.

Mots clés : S7-300, EFV, WinCC.

Abstract : We have established a communication bus between multiple S7-300 CPUs using PROFIBUS and MPI communication technologies and transferred the necessary data for display and control in real time to the WinCC software.

We have looked into the uses and functionalities of multiple equipment such as industrial sensors,VFDs , and motors. All of which was centered around their integration to programmable logic controllers, connection to dedicated software, treatment and display of information on an HMI.

In addition, and as a showcase for what we have learned, we have also developed an application using a Siemens S7-300 PLC which controls the movement of a cart through a Schneider VFD, together with a photosensor used for multiple functionalities.

Keywords : VFD, PROFIBUS, MPI.

Listes des acronymes et abréviations :

API : Automate Programmable Industriel.

PLC : Programmable Logic Controller.

TIA : Totally Integrated Automation.

MPI : Multi-Point Interface.

PROFIBUS : Process Field Bus.

VFD : Variable Frequency Drive.

EFV : Variateur électronique de Vitesse.

CPU : Central Processing Unit.

IHM : Interface Homme-Machine

HMI : Human Machine Interface.

Table des matières

Liste des Figures

Liste des tableaux

Introduction générale

Dans l'environnement de l'automatisation des processus industriels, la communication entre les dispositifs qui interviennent dans le contrôle de ces systèmes est un élément clé pour permettre non seulement un fonctionnement correct, mais également la supervision et le contrôle de ces processus.

Les communications entre les appareils s'effectuent en utilisant différents protocoles de communication industriels. Un protocole de communication est un ensemble de règles permettant le transfert et l'échange de données entre les périphériques.

Au fur et à mesure des progrès technologiques dans le domaine de l'électronique, le contrôle automatisé des processus industriels s'est imposé aux méthodes de production traditionnelles.

La prochaine étape vers une industrie entièrement automatisée consistait à interconnecter ces systèmes automatisés. Cette étape a permis, outre une gestion plus efficace des processus de production, une meilleure disponibilité des informations provenant des appareils de terrain, de manière centralisée au niveau de l'usine.

Dans le cadre de ce travail, nous avons établi une communication entre plusieurs automates Siemens de la série S7, via un support de communication et un protocole exclusif à la sélection Siemens. Ensuite, nous avons mis en place un système de supervision des fonctionnalités individuelles que nous avons programmées dans chaque automate.

Ce document va être divisé en trois chapitres. Le premier chapitre traitera des systèmes de supervision, en commençant par une introduction aux systèmes automatisés, les hiérarchies trouvées dans l'environnement de communication industrielle, puis le niveau spécifique de supervision qui nous intéresse dans le travail.

Le deuxième chapitre portera sur le matériel que nous avons utilisé dans les différents automates et sur le processus du type de communication que nous avons utilisé.

Le troisième chapitre sera consacré au logiciel, à la programmation des automates, aux fonctionnalités des IHM créées, puis les résultats du travail.

Chapitre 1 Les systèmes de supervision industriels

1.1 Introduction

La supervision est une technique industrielle de suivi et de pilotage informatique de procédés de fabrication automatisés. La supervision concerne l'acquisition de données (mesures, alarmes, retour d'état de fonctionnement) et des paramètres de commande des processus généralement confiés à des automates programmables.

Dans l'informatique, la supervision est la surveillance du bon fonctionnement d'un système ou d'une activité [2].

1.2 Les systèmes automatisés

Un système automatisé est un ensemble d'éléments qui effectue des actions sans intervention de l'utilisateur : c'est l'opérateur. Celui-ci se contente de donner des ordres de départ et si besoin d'arrêt. [23]

Certains objectifs notables que ces systèmes tentent d'atteindre comprennent:

- ✓ Eliminer les tâches répétitives
- ✓ Simplifier le travail de l'humain
- ✓ Augmenter la sécurité
- ✓ Accroitre la productivité
- ✓ Economiser les matières premières et l'énergie
- ✓ S'adapter à des contextes particuliers
- ✓ Maintenir la qualité [3].

1.2.1 Les automates programmables

Un automate est un dispositif capable d'assurer, sans intervention humaine, un enchainement d'opérations, correspondant à la réalisation d'une tache donnée. C'est-à-dire, un appareil conçu pour remplacer l'homme dans l'exécution de certaines tâches

Un automate programmable industriel (API) est une machine électronique spécialisée dans la conduite et la surveillance en temps réel de processus industriels et tertiaires. Il exécute une suite d'instructions introduites dans ses mémoires sous forme de programmes, et s'apparente par conséquent aux machines de traitement de l'information [4].

a. Avantages de l'utilisation d'un API dans l'automatisation

- Nature flexible: un modèle de API peut être utilisé pour différentes opérations selon les besoins.
- Facile à installer et à dépanner: Dans les systèmes à relais câblés, le temps d'installation est supérieur à celui des panneaux de commande basés sur un API.
- Rentable: La technologie avancée et la grande production d'API le rendent moins cher que les autres systèmes à contrôleur ou à relais.
- Fonction de simulation: le logiciel de programmation d'automate est fourni avec les fonctions de simulation par défaut.
- Méthodes de programmation simples: L'API dispose de méthodes de programmation simples lui permettant de programmer des automates de type Ladder ou Booléen.
- Facilité de maintenance: Par rapport aux systèmes de contrôle tels que les systèmes à relais ou à micro-contrôleurs, les coûts de maintenance des API sont faibles.
- Documentation: Le programmeur peut programmer et imprimer facilement les programmes de l'automate pour une utilisation ultérieure.

b. inconvénients de l'utilisation des automates

- Les constructeurs d'automates proposent uniquement une architecture en boucle fermée.

- Les API sont propitiatoires, ce qui signifie que les logiciels et les pièces d'un fabricant ne peuvent pas être facilement utilisés en combinaison avec une partie d'un autre fabricant.

- un certain nombre de modules optionnels doivent être ajoutés pour optimiser la flexibilité et les performances [32].

c. la structure des API

Les automates programmables sont constitués de quelques composants différents; Modules d'entrée et de sortie (E / S), une alimentation, un dispositif de programmation et une unité centrale de traitement (CPU). La CPU dispose également d'une mémoire pour les fonctions logiques et de tables de données pour le stockage d'E / S.

d. Types d'automates

Sur la manière moins subtile de classer les API, il y a les automates compacts et modulaires.

- Dans un **automate compact**, la section d'entrée et la section de sortie de l'automate sont intégrées dans le microcontrôleur lui-même.

Cela signifie que chaque type de sortie ou d'entrée est fixe et déterminé par le fabricant. De plus, le nombre d'entrées et de sorties ne peut pas être étendu dans ce type d'automate.

- L'**API modulaire** est un type qui permet son expansion grâce à l'utilisation de modules, d'où le terme « modulaire».

Les modules apportent à l'automate des fonctionnalités supplémentaires, telles que la possibilité d'augmenter le nombre d'unités d'E / S, des modules à co-processeur (les modules à co-processeur sont des micro-ordinateurs programmables qui étendent les capacités et la fonctionnalité d'un système d'automate. Un module à co-processeur est contrôlé par la CPU).

L'alimentation, le module de communication et le module d'entrée / sortie sont tous distincts du microcontrôleur actuel. Vous devez donc les connecter manuellement les uns aux autres pour créer votre système de contrôle API.

e. Principe de fonctionnement de l'API

Tous les automates ont quatre étapes de base d'opération répétées plusieurs fois par seconde. Initialement, lorsqu'il est allumé pour la première fois, il vérifie si son matériel et son logiciel sont défectueux. S'il n'y a pas de problèmes, toutes les entrées sont copiées et leurs valeurs sont copiées dans la mémoire. C'est ce que l'on appelle l'analyse d'entrée. En utilisant uniquement la copie en mémoire des entrées, le programme logique sera résolu une fois, on appelle cela l'analyse logique. Lors de la résolution de programme logique, les valeurs de sortie ne sont modifiées que dans la mémoire temporaire. Lorsque l'analyse de programme est terminée, les sorties sont mises à jour à l'aide des valeurs temporaires en mémoire. Cette opération est appelée analyse de sortie. L'API redémarre maintenant le processus en lançant un auto-contrôle des défauts. Ce processus se répète généralement 10 à 100 fois par seconde[24].

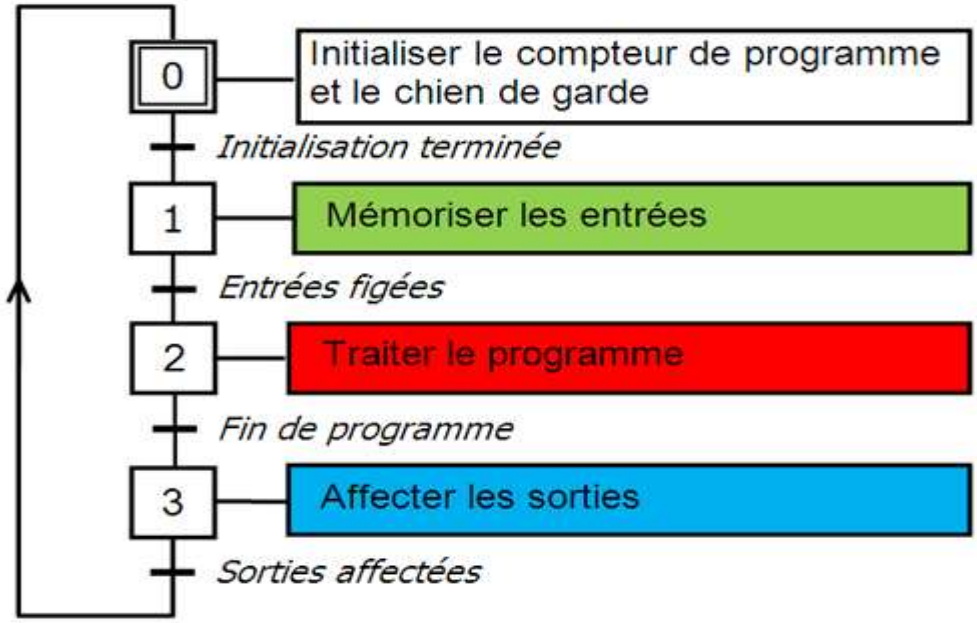


Figure 1. 1 fonctionnement cyclique d'un API

Chien de garde :

Un chien de garde, en anglais *watchdog*, est un circuit électronique ou un logiciel utilisé en électronique numérique pour s'assurer qu'un automate ou un ordinateur ne reste pas bloqué à une étape particulière du traitement qu'il effectue. C'est une protection destinée généralement à redémarrer le système, si une action définie n'est pas exécutée dans un délai imparti. [33]

f. Fabricants de API

La marque de l'automate est sélectionnée en fonction du lieu de travail. Certains endroits utilisent plusieurs fabricants dans une même installation. Certains fabricants notables sont énumérés ci-dessous.

- **Allen-Bradley**

Rockwell Automation et sa gamme de contrôleurs Allen-Bradley sont l'un des plus importants fabricants d'API utilisés aux États-Unis[25]. Les automates Allen Bradley sont disponibles dans différentes tailles, ils sont disponibles dans les tailles grande, moyenne et micro. Un automate de grande taille nommé PAC (Controllable Automation

Controllers) est capable de contrôler une installation entière si elle est correctement conçue.

Ceci est réalisé avec ses contrôleurs et le logiciel ControlLogix, GuardPLC pour les systèmes de sécurité et le progiciel SoftLogix.

Le SLC500 et MicroLogix sont de petits automates, ils sont utilisés pour des applications plus petites.

La programmation pour les contrôleurs Allen-Bradley se présente sous la forme de RSLogix 5000 et du nouveau Studio 5000.

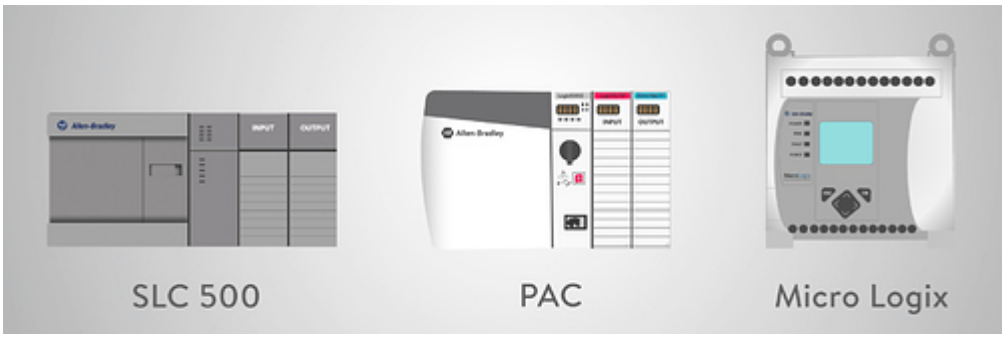


Figure 1. 2 Automates Allen-Bradley

- **ABB**

ABB propose la série AC500, avec des automates petits et grands.

La série AC500 se distingue des autres marques par son concept «tout module sur tout processeur». Cela signifie que quel que soit le processeur dont vous disposez pour votre automate AC500, vous pouvez utiliser n'importe quel module ABB AC500.

Figure 1. 3 Automates ABB

- **Mitsubishi API**

Mitsubishi API a la série MELSEC. Liste en tant que série MELSEC iQ, série MELSEC L, série MELSEC F, série MELSEC A.

La série iQ appartient à la dernière génération, la série F à des automates compacts peu coûteux et la série MELSEC QS / WS est conçue pour le contrôle de la sécurité.

- **Siemens**

Siemens est une multinationale allemande et probablement le plus grand nom du secteur de l'automatisation.

Siemens et sa gamme de contrôleurs SIMATIC (Simatic S7-1200, Simatic S7-1500, Simatic S7-300, Simatic S7-400.) Offrent une option aux utilisateurs de tous les besoins.

SIMATIC signifie Siemens Automatic. Les contrôleurs les plus récents disposent de fonctionnalités standard très intéressantes, notamment les connexions Ethernet TCP / IP simplifiées et les communications Profinet IO. Profibus est inclus ou facilement ajouté avec un module.

En ce qui concerne le logiciel SIMATIC, Siemens offre une excellente efficacité tout au long du processus d'automatisation. Leur SIMATIC STEP7 permet aux utilisateurs de configurer, programmer, tester et diagnostiquer les contrôleurs de base et avancés.

Figure 1. 4 Automates Siemens

1.2.2 Logiciel TIA Portal

La plateforme « Totally Intergrated Automation Portal » est le nouvel environnement de travail Siemens qui permet de mettre en œuvre des solutions d'automatisation avec un système d'ingénierie intègre comprenant les logiciels SIMATIC Step7 et SIMATIC WinCC.

SIMATIC STEP 7, intégré à TIA Portal, est le logiciel de configuration, programmation, vérification et diagnostic de tous les automates SIMATIC. Doté d'un grand nombre de fonctions conviviales, SIMATIC STEP 7 garantit une efficacité nettement supérieure pour toutes les tâches d'automatisation, qu'il s'agisse de la programmation, de la simulation, de la mise en service ou de la maintenance **[8].**

a. WinCC sur TIA portal

WinCC (TIA portal) est un logiciel d'ingénierie pour la configuration de pupitres SIMATIC, de PC industriel SIMATIC et de PC standard par le logiciel de visualisation. Le SIMATIC WinCC dans le TIA portal fait partie d'un nouveau concept d'ingénierie intégré qui offre un environnement d'ingénierie homogène pour la programmation et la configuration de solution de commande, de visualisation d'entrainement **[8].**

Avec SIMATIC WinCC Professional, Siemens propose un système SCADA parfaitement intégré au TIA Portal, SIMATIC WinCC Runtime Professional est un système de contrôle et de surveillance basé sur PC permettant la visualisation et le contrôle par l'opérateur des processus, des flux de production, des machines et des installations dans tous les secteurs [26].

Les outils industriels tels que les automates programmables, les capteurs et les systèmes de supervision communiquent les données de différentes manières et pour cela, ils nécessitent des protocoles et des niveaux de réseaux de communication différents.

1.3 Les réseaux locaux industriels

Un réseau local industriel (RLI) est un système de communication entre plusieurs équipements de type industriel (capteurs, automates, actionneurs, ...) dans une zone géographique limitée (un « terrain »). On parle aussi de « bus de terrain » ou de « réseau de terrain ». Elle est en première approximation un réseau local utilisé dans une usine ou tout système de production pour connecter diverses machines afin d'assurer la commande, la surveillance, la supervision, la conduite, la maintenance, le suivi de produit, la gestion, en un mot, l'exploitation de l'installation de production.

Un processus de fabrication nécessite le téléchargement d'un programme sur un automate programmable ; il doit être transmis sans erreur le plus rapidement possible, sans toutefois qu'un léger retard soit trop préjudiciable si le processus physique pendant ce temps est dans un état stable. L'opération n'est pas critique du point de vue temporel, mais elle doit être effectuée sans erreur.

De tels besoins nécessitent d'être pris en charge que ce soit au niveau physique ou au niveau protocoles : liaison et application. Au niveau physique les réseaux locaux industriels doivent être dotés de moyens résistant aux perturbations, aux chocs, à la chaleur, …etc. tel que les câbles et les connecteurs blindés. Les moyens de communication utilisés à chaque niveau doivent répondre en termes de débit aux besoins de ce niveau **[10].**

1.3.1 Le rôle des réseaux industriels

Un réseau industriel joue le même rôle qu'un réseau normal. Le but premier est toujours de transmettre des informations entre plusieurs machines. Lorsque l'on parle de réseaux, on sous-entend généralement que les machines sont des ordinateurs. Lorsque l'on parle des réseaux industriels, il s'agit de faire communiquer des machines qui ne sont plus seulement des ordinateurs. On fait communiquer des appareils différents tels que des ordinateurs, des automates programmables, des capteurs, des actionneurs, et des régulateurs [10].

Les réseaux permettent aux entreprises de :

• Partager des ressources (imprimantes, disque dur, processeur, etc.)

• Réduire les coûts.

• Augmenter la fiabilité : dupliquer les données et le traitement sur plusieurs machines. Si une machine tombe en panne une autre prendra la relève.

• Fournir un puissant média de communication (e-mail, conférence virtuelle, etc.) [10].

1.3.2 Les types de réseaux industriels

a. Introduction

Le modèle OSI est une base de référence pour identifier et séparer les différentes fonctions d'un système de communication basé sur une structure en couches. Chaque couche (matérielle, logicielle) assure un ensemble de fonctions spécifiques. Chaque couche utilise les services de la couche immédiatement inférieure pour rendre à son tour un service à la couche immédiatement supérieure. Un protocole est le langage commun (règles de dialogue) que doivent connaître et utiliser deux couches homologues (couche de même niveau) pour dialoguer.

Le modèle OSI possède 7 couches ou niveaux qui définissent les fonctions des protocoles de communication qui vont de l'interface physique à l'interface des applicatifs utilisant le réseau. En raison de son apparence, la structure est très souvent appelée pile ou pile de protocoles.

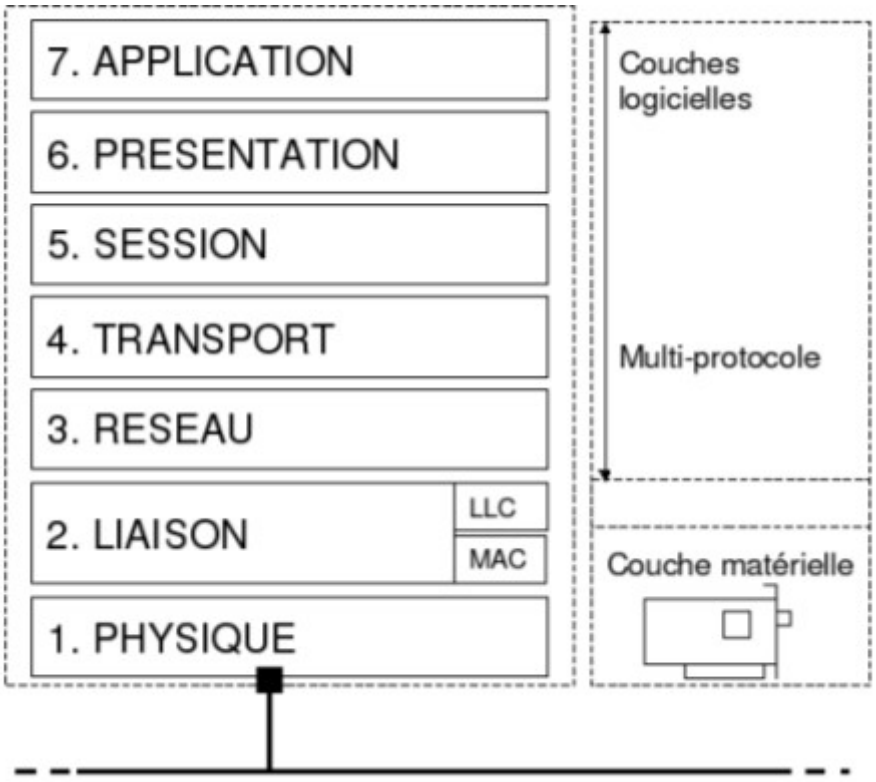


Figure 1. 5 couches de modèle OSI

La couche (1) : physique Rassemble les moyens électriques, mécaniques, optiques ou hertziens par l'quels les informations sont transmises. Les unités sont donc des bits (0 ou 1).

La couche (2) : liaison de donnée Gère la fiabilité de transfert des bits d'un nœud à l'autre du réseau, comprenant entre autres les dispositifs de détection et correction d'erreur, ainsi que le système de partage des supports. L 'unité de donné à ce niveau est appelée une trame.

La couche (3) : réseau Aiguille les données à travers un réseau à commutation. L'unité de données s'appelle en général un paquet.

La couche (4) : transport Regroupe les règles de fonctionnement de bout en bout, assurant ainsi la transparence du réseau vis-à-vis des couches supérieures. Elle traite notamment l'adressage, l'établissement des connexions et la fiabilité de transport.

La couche (5) : session Réunit les procédures de dialogue entre les applications, établissement et interruptions de la communication, cohérence et synchronisation des opérations.

La couche (6) : présentation Traite les formes de représentation des données permettant la traduction entre machines différentes.

La couche (7) : application Source et destination de toutes les informations à transporter, la couche application rassemble toutes les applications qui ont besoin de communiquer par le réseau : messagerie électronique, transfert de fichiers, gestionnaire de base de données **[11].**

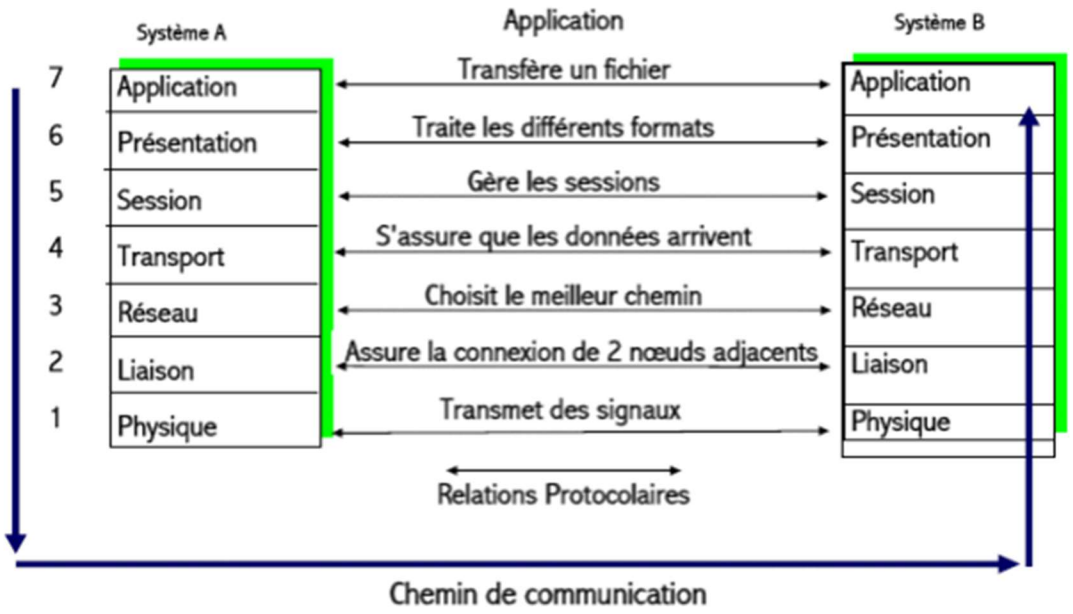


Figure 1. 6 relation protocolaire dans les couches de modèle OSI

- Ce système (Modél OSI) est très pratique pour diviser les protocoles de communication industriels et la section suivante se concentrera sur les principaux types.

b. Les types des bus de communication

Un Bus est un ensemble de conducteurs commun à plusieurs circuits permettant d'échanger des données. Les échanges sont régis par un protocole.

On distingue néanmoins par complexité décroissante :

Réseaux locaux industriels (data bus): communication entre l'automatisme et le monde informatique [27]

→ Quantités importantes d'information (messages, fichiers)

→ Temps de réaction de 1 s à 10 s (temps non critique)

→ Longue distance possible

Bus de terrain (field bus): réseaux entre unités de traitement (automates programmables, superviseurs, commandes numériques ...), (device bus) bus et réseaux pour la périphérie d'automatisme (variateurs, robots, axes ...) permet la communication d'unités de traitement pour la coordination des automatismes distribués

→ Quantité relativement faible de données < 256 octets

→ Temps de réaction < 100 ms (notion d'événements temps réel)

→ Distance < 1 km

Bus capteur/actionneur (sensor bus): interface avec les capteurs/actionneurs, relie entre eux des nœuds à intelligence limitée ou nulle

→ Niveau bits

→ Temps de réaction < 10 ms (contrainte temps réel)

→ Distance < 100 m

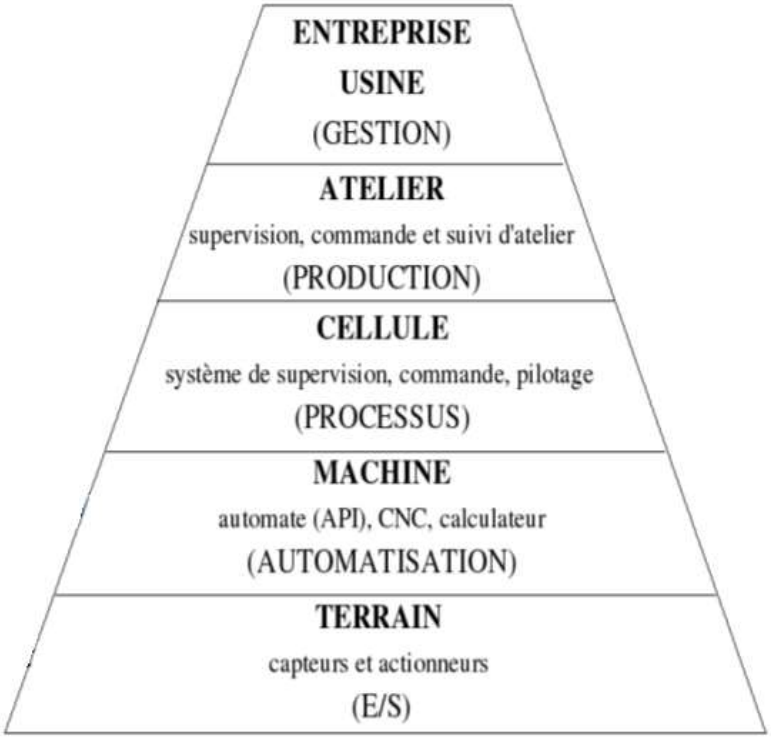


Figure 1. 7 Pyramide du CIM (Computer-integrated manufacturing)

1.3.3 Bus de terrain

Un bus de terrain est un système de communication numérique dédié qui respecte le modèle d'interconnexion des systèmes ouverts (OSI) de l'Organisation de Standardisation Internationale (ISO 7498 – 1983).

Un bus de terrain est basé sur la restriction du modèle OSI à 3 couches :

- Couche Application
- Couche Liaison
- Couche Physique

C'est un réseau bidirectionnel, sériel, multibranche (multidrop), reliant différents types d'équipements : E/S déportées, Capteur / Actionneur, Automate programmable (API), CNC, Calculateur, PC Industriel, ...[27]

1.3.4 Protocoles de communication Siemens

Dans le deuxième chapitre, nous expliquerons comment nous avons utilisé la communication de Siemens.

Voici quelques exemples de supports de communication Siemens remarquables.

a. Multi-Point Interface (MPI)

MPI (Multi-Point Interface) est l'interface intégrée pour les produits SIMATIC :

- contrôleurs
- panneaux
- Appareils de programmation / PC

Avec MPI, des sous-réseaux avec les propriétés suivantes sont établis :

- petite portée
- Peu de nœuds
- Petites quantités de données [28]

MPI offre une capacité réseau simple avec les services suivants :

- communication PG / OP
- communication S7
- communication de base S7
- Global Data Communication (GD)

MPI prend en charge des vitesses de transmission allant de 187,5 kbps à 12 Mbps. Les adresses des nœuds MPI doivent être uniques et définies avec le périphérique de programmation PC.

MPI est basé sur la norme PROFIBUS (CEI 61158 et EN 50170) et prend en charge les topologies de bus suivantes :

● Ligne

● étoile

● arbre

b. PROFIBUS

PROFIBUS est un système de bus qui met en réseau des systèmes d'automatisation et des appareils de terrain compatibles avec PROFIBUS. En tant que moyen de communication pour le terrain, PROFIBUS est une partie importante de Totally Integrated Automation (TIA)

Les différents réseaux de communication peuvent être utilisés indépendamment les uns des autres ou peuvent être combinés les uns avec les autres.

- **PROFIBUS DP** (distributed I/O) est un réseau de communication pour le niveau terrain (selon CEI 61158-2 / EN 61158-2). avec les protocoles d'accès hybrides bus à jeton et maître-esclave. La mise en réseau se fait au moyen de lignes à deux fils ou de câbles à fibres optiques. Des vitesses de transmission de données de 9,6 kbps à 12 Mbps sont possibles) [29].

- **PROFIBUS PA** est le PROFIBUS pour Process Automation (PA). Il connecte le protocole de communication PROFIBUS DP avec la technologie de transmission MBP (Manchester Bus Powered) à la norme IEC 61158-2.

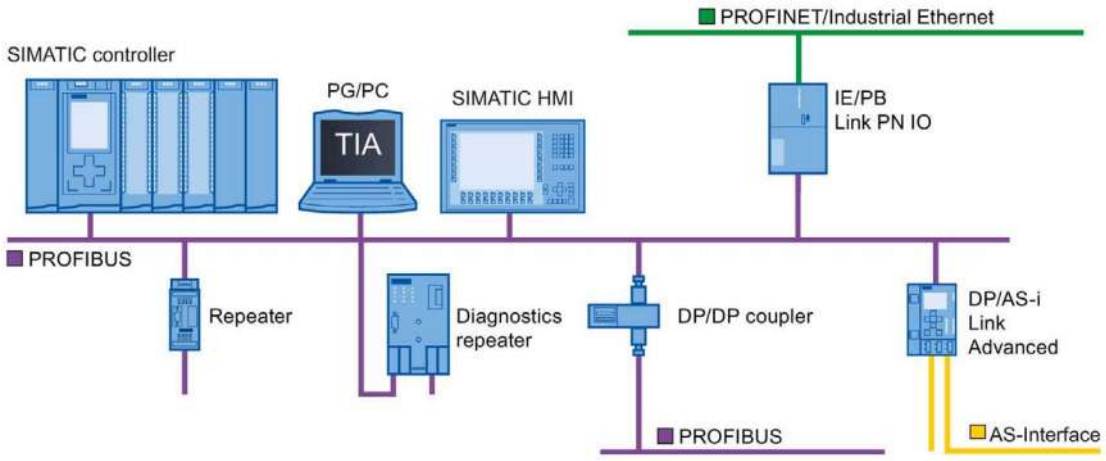


Figure 1. 8 Options de connexion à PROFIBUS DP

c. PROFINET

Dans le cadre de Totally Integrated Automation (TIA), PROFINET IO constitue le développement continu et cohérent de:

● PROFIBUS DP.

● Ethernet industriel.

Cela garantit la migration en douceur de PROFIBUS DP dans le monde PROFINET.

PROFINET IO utilise un matériel et des logiciels Ethernet traditionnels pour définir un réseau structurant l'échange de données, les alarmes et les diagnostics.

En utilisant des modules de communication et des composants logiciels adaptés, vous pouvez utiliser une station PC en tant que contrôleur PROFINET IO[30].

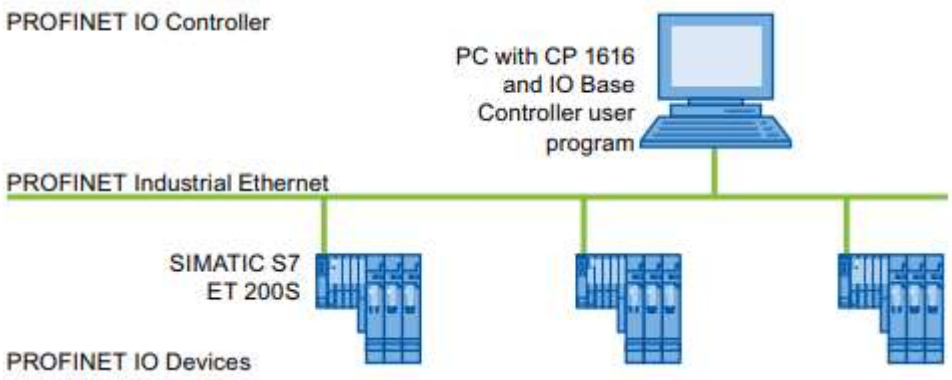


Figure 1. 9 CP 1616 en tant que contrôleur PROFINET IO

1.4 Structure de la supervision des procédés industriels

1.4.1 Définition de la Supervision Industrielle

La supervision industrielle permet de suivre en temps réel une installation ou une machine industrielle. Elle permet d'avoir un affichage dynamique du processus avec les différentes alarmes, défauts et événements survenant pendant l'exploitation de la machine. Les procédés de supervision actuelle se basent sur les architectures de systèmes distribués permettant la surveillance ou le monitoring à distance.

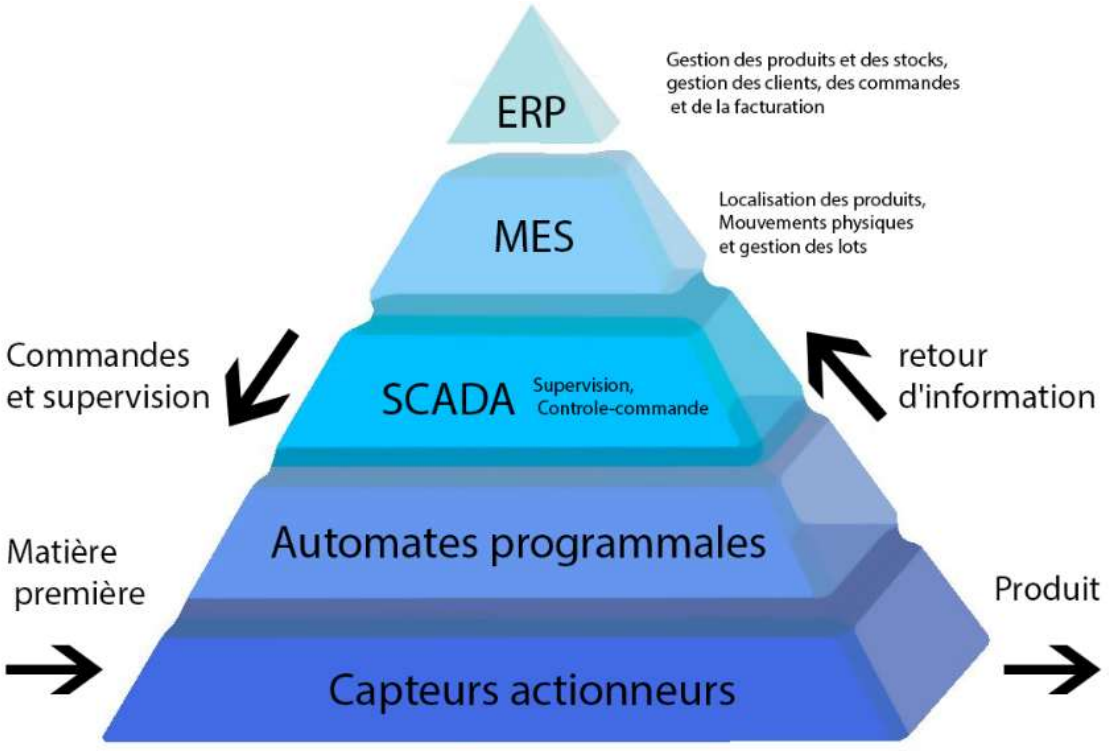


Figure 1. 10 La supervision dans la hiérarchie d'une entreprise manufacturière

ERP: Le système de gestion ERP (Entreprise Ressource Planning) qui est l'outil du responsable de production, lui permettant la prise de décision et le planning des opérations de production suite à une analyse des données de production.

MES: Le système d'exécution de la production (MES) qui permet de recevoir des données en flux direct à partir des systèmes de supervision et des opérateurs. Il conserve la trace de toutes les informations de production en temps réel et permet d'effectuer des activités d'analyse. Bien que le MES se situe à un niveau supérieur de la supervision, cette dernière prend en charge les besoins du MES. Le MES est un système informatique dont les objectifs sont d'abord de collecter en temps réel les données de production de tout ou partie d'une usine ou d'un atelier. Ces données collectées permettent ensuite de réaliser un certain nombre d'activités d'analyse afin de fournir les informations nécessaires à l'optimisation des activités de production, depuis la création de l'ordre de fabrication jusqu'au produit fini.

SCADA et DCS (les systèmes de contrôle commande):

Les systèmes de contrôle distribués (DCS) sont utilisés dans les applications industrielles et de génie civil pour surveiller et contrôler les équipements distribués avec intervention humaine à distance.

Un DCS est un système de contrôle de processus qui utilise un réseau pour interconnecter des capteurs, des contrôleurs, des terminaux de commande et des actionneurs. Un DCS contient généralement un ou plusieurs ordinateurs pour le contrôle et utilise principalement des interconnexions et des protocoles propriétaires pour les communications.

DCS est un terme très large qui décrit des solutions à travers une grande variété d'industries, y compris:

* Les réseaux électriques et les centrales électriques

* Systèmes de contrôle environnemental

* Des signaux de trafic

* Systèmes de gestion de l'eau

* Raffinage et usines chimiques

* Fabrication de produits pharmaceutiques

SCADA est l'acronyme de Supervisory Control And Data Acquisition. Le SCADA peut être appelé Interface Homme-Machine (IHM) en Europe. Le terme fait référence à un système de mesure (et de contrôle) distribué à grande échelle. Les systèmes SCADA sont utilisés pour surveiller ou contrôler les processus chimiques, physiques ou de transport.

Le terme SCADA se réfère généralement à un système central qui surveille et contrôle un site complet. La majeure partie du contrôle du site est effectuée automatiquement par une unité terminale à distance (RTU) ou par un automate programmable (API). Les fonctions de contrôle de l'hôte sont presque toujours limitées aux fonctions de base de dépassement ou de supervision du site.

Un système SCADA comprend 2 sous-ensembles fonctionnels : la commande et la surveillance.

La commande :

Les fonctions de commande en marche normale sont :

• L'envoi de consignes vers le procédé dans le but de provoquer son évolution.

• L'acquisition de mesures ou de compte-rendu permettant de vérifier que les consignes envoyées vers le procédé produisent exactement les effets escomptés.

• L'envoi vers le procédé d'ordres prioritaires permettant de déclencher des procédures de sécurité (arrêts d'urgence par exemple).

La surveillance :

La partie surveillance d'un superviseur a pour objectifs :

• Recueillir en permanence tous les signaux en provenance du procédé et de la commande.

• Reconstituer l'état réel du système commandé.

• Fait toutes les inférences nécessaires pour produire les données utilisées pour dresser des historiques de fonctionnement.

• Mettre en oeuvre un processus de traitement de défaillance le cas échéant

• La détection d'un fonctionnement ne correspondant plus à ce qui est attendu.

1.5 Conclusion

Ce chapitre a été une introduction à l'automatisation industrielle, en mettant l'accent sur les aspects les plus importants de l'automatisation.

Le travail que nous avons effectué se situe au niveau du terrain et au niveau SCADA de la hiérarchie de supervision, et dans le chapitre suivant, nous allons nous concentrer sur ce sujet.

Chapitre 2 Les composants utilisés

2.1 Introduction

Dans ce chapitre, nous allons donner des détails sur les équipements que nous avons utilisés pour réaliser un exemple d'utilisation de la communication entre des automates.

2.2 CPU SIMATIC S7-312C

Le API S7 312c est la CPU que nous avons utilisée pour la communication et la programmation du travail.

Sa référence et ses caractéristiques :

Nº de référence : 6ES7 312-5BF04-0AB0

Mémoire de travail de 64 Ko ; 0,2 ms/kilo-instructions ; DI10/DO6 intégrées ; 2 sorties d'impulsions (2,5 kHz) ; 2 voies de comptage et de mesure avec codeurs incrémentaux 24 V (10 kHz) ; interface MPI ; configuration sur une rangée pouvant comporter jusqu'à 8 modules. [18]

Figure 2. 1 CPU SIMATIC S7-312C

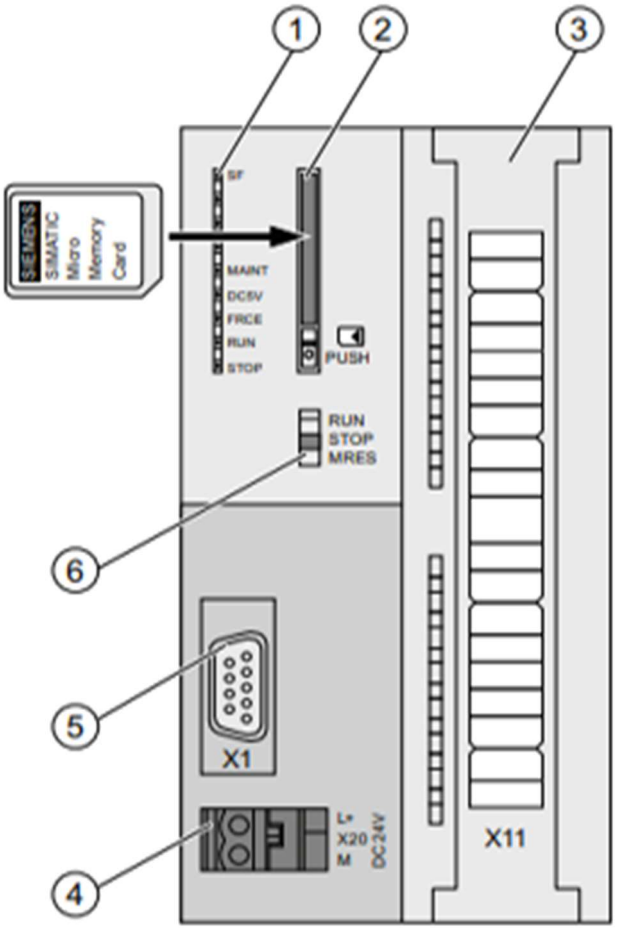


Figure 2. 2 Vue de face du module CPU SIMATIC S7 312C

1- Indicateurs d'état et d'erreur

2- Emplacement pour la carte mémoire SIMATIC

3- bornes des entrées et sorties intégrées

4- connexion d'alimentation

5- interface x1(MPI)

6- sélecteur de mode

Commutateur de mode de fonctionnement

Vous pouvez régler le mode de fonctionnement de la CPU via le connecteur de mode de fonctionnement.

Position	Signification	Explications
RUN	Mode de fonctionnement RUN	La CPU traite le programme utilisateur.
STOP	Mode de fonctionnement STOP	La CPU ne traite aucun programme utilisateur.
MRES	Effacement général	Position du commutateur de mode de fonctionnement pour l'effacement général de la CPU. L'effacement général à l'aide du commutateur de mode de fonctionnement nécessite une séquence d'actions particulière de votre part.

Tableau 2. 1 position du commutateur de mode fonctionnement

Désignation de led	Couleur	Sens
SF	Rouge	Défaut matériel ou erreur logicielle
MAINT	Jaune	Entretien demandé

DC5V	Verte	L'alimentation 5v pour les processeurs cpu et s7-300 est OK
FRCE	Jaune	Led est allumé : travail de force active Led clignote à 2hz : fonction de test flash de nœud
STOP	Jaune	CPU en STOP, en HOLD ou au Start-up La led clignote à une cadence de 0,5Hz lorsqu'une réinitialisation de la mémoire est Demandée et à 2hz pendant la réinitialisation

Tableau 2. 2 Indicateurs d'état et d'erreur.

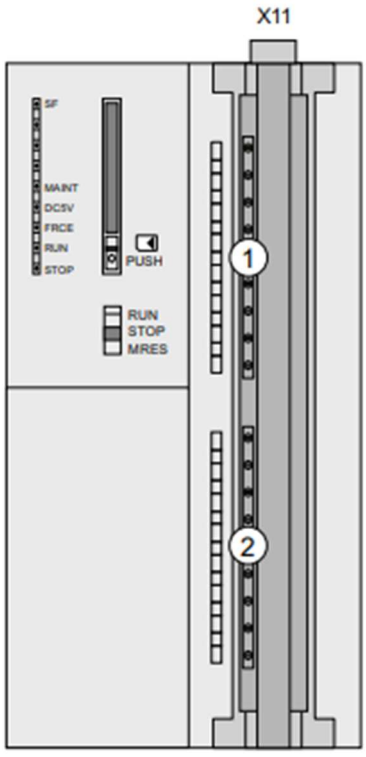


Figure 2. 3 les entrées /sorties numériques intégrées du processeur avec portes avant ouvertes.

1- entrées numériques (broches 2 à 10)

2- entrées digital (11) sorties numériques (broches 14 à 19)

2.3 Supports de communication

Siemens propose une grande variété de méthodes de communication, dont les principales sont PROFINET / ethernet industriel, les réseaux locaux sans fil industriels IWLAN, PROFIBUS et l'interface multipoint (MPI). Dans notre travail, nous avons utilisé le MPI pour la communication entre les CPU via un système de câblage PROFIBUS. Et un adaptateur MPI entre l'un des automates et le PC.

2.3.1 Adaptateur MPI

Le Multi Point Interface - Siemens (MPI) est une interface propriétaire des automates programmables industriels SIMATIC S7.

Nous utilisons cet adaptateur pour la communication entre le processeur et l'ordinateur, ces utilisations consistent à télécharger des programmes, à diagnostiquer des modules et à établir une connexion avec l'IHM exécutée sur le PC.

L'adaptateur MPI (Figure 2.15) se connecte entre le PC (via le port USB) et l'API (port DB9).

Figure 2. 4 Un Adaptateur MPI.

Le réseau MPI électrique est construit le plus souvent avec un câble blindé et torsadé à

deux fils et peut être réalisé jusqu'à une longueur de 50 m. Ces 50 m sont mesurés sur la distance premier participant - dernier. MPI sert également pour la communication avec les composants mis en place pour 'servir et visualiser', ainsi que pour la communication entre deux automates[15].

2.3.2 PROFIBUS (Process Field Bus)

Est le nom d'un type de bus de terrain propriétaire et de son protocole, inter-automates et de supervision. Il est devenu peu à peu une norme de communication dans le monde de l'industrie ces dix dernières années [22].

PROFIBUS utilise une topologie de réseau de bus, un câble PROFIBUS (Figure 2.4) se connecte à un connecteur (Figure 2.18) qui se connecte à un automate. il existe différents types et générations de connecteurs et ceux de la figure 2.17 sont ceux utilisés dans notre travail.

Nous avons utilisé le socket PG pour établir une connexion MPI entre le PC et l'API.

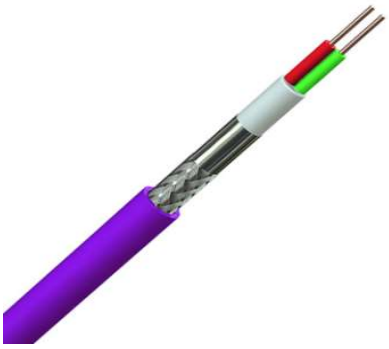


Figure 2. 5 Coupe détaillée montrant les composants constituant le câble PROFIBUS

Sans PG Socket Avec PG Socket

Figure 2. 6 Connecteur PROFIBUS

Figure 2. 7 un connecteur Profibus ouvert et un câble Profibus

a. Terminaison de câbles Profibus

Pour minimiser les réflexions du signal, le câble Profibus doit être terminé à la fin. La réflexion du signal se produit lorsque, dans un signal transmis par un support de transmission, tel qu'un câble de cuivre ou une fibre optique, une partie de l'énergie du signal peut être renvoyée au lieu d'être renvoyée complètement le long du câble jusqu'à la seconde extrémité. Tout comme la mise à la terre, le problème est tel qu'une mauvaise terminaison des câbles Profibus empêchera le bus de fonctionner.

Une terminaison Profibus est réalisée en insérant une résistance de 220 Ω à chaque extrémité de la ligne Profibus. Ainsi, vous insérez une résistance de 220 Ω dans la

première et une dans la dernière station. L'utilisation d'une résistance de 220 Ω s'explique par le fait que les deux résistances de 220 Ω sont connectées en parallèle. La connexion en parallèle des deux résistances donne une résistance totale de 110 Ω. La résistance de boucle du câble Profibus DP standard est de 110 Ω / km. Sachez qu'un câble Profibus PA standard a une résistance de boucle de 44 Ω / km. [31]

La plupart des connecteurs Profibus ont une option de terminaison. Cela signifie que vous n'avez pas besoin de connecter votre propre résistance manuellement. Vous pouvez activer ou désactiver la terminaison dans le connecteur en basculant le commutateur, généralement placé en haut du connecteur. Ceci est utile de deux manières. Tout d'abord, cela rend la terminaison très facile, car vous pouvez effectuer toutes vos connexions, puis basculer les commutateurs à l'endroit où vous devez raccorder le câble Profibus. La deuxième raison est que le dépannage devient plus facile. Sans rien déconnecter, vous pouvez vérifier si la résiliation est effectuée correctement.

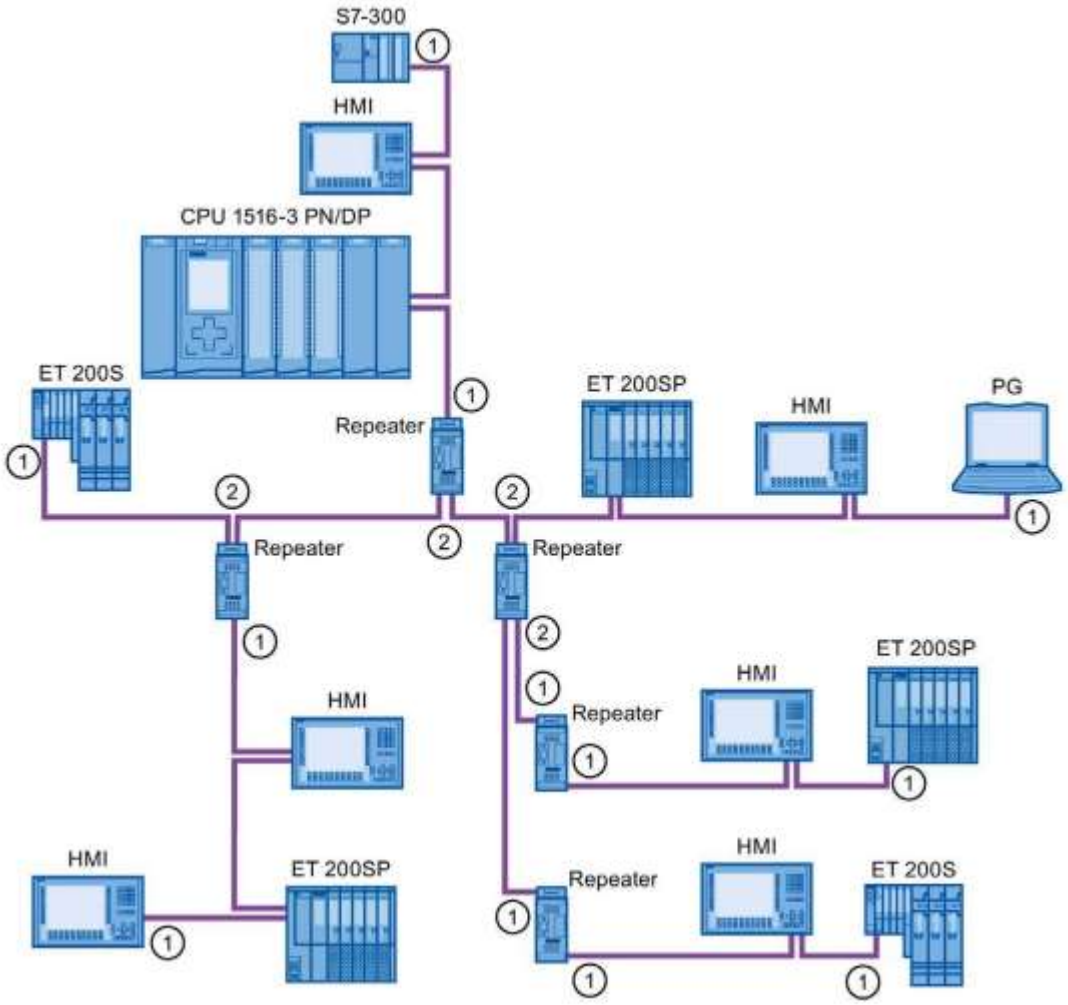


Figure 2. 8 exemple de configuration avec cinq répéteurs RS485

1: Connecter la résistance de terminaison

2: Ne pas connecter la résistance de terminaison

2.4 Moteur asynchrone (machine à induction)

Pour déplacer le chariot contrôlé par le premier automate, nous utilisons un moteur asynchrone, le moteur asynchrone est, de beaucoup, le moteur le plus utilisé dans l'ensemble des applications industrielles, du fait de son faible cout, de son faible encombrement, de son bon rendement et de son excellente fiabilité.

La machine asynchrone est constituée de deux éléments principaux :

Stator : constitué de trois enroulements (bobines) parcourus par des courants alternatifs triphasés et possède p paires de pôles ("nombre de bobinage triphasé au sein dans le stator")

Le rotor : Partie tournante du moteur. Le rotor peut être constitué par un bobinage triphasé, mais, le plus souvent, Il est constitué d'une masse métallique dont de l'aluminium pour l'alléger. On parle alors de rotor à cage d'écureuil[14].

fiche technique du moteur utilisé:

Model: Carpanelli - M56B4

Courant Nominal: 0.78A

Tension: 230V

Puissance: 0.09 KW

RPM à 50hz: 1380 RPM

2.5 Le variateur de vitesse

Altivar12 Variateurs de vitesse pour moteurs asynchrones

Figure 2. 9 le variateur de Vitesse Altivar12

Utilisé pour contrôler le démarrage / arrêt, l'accélération / décélération, le couple, la vitesse et le sens de rotation du moteur ; et pour la surveillance générale des entrées données au moteur à travers l'écran en vedette[21].

Avant d'utiliser ce VFD pour contrôler le moteur, nous avons programmé certains paramètres qui conviennent le mieux à notre travail.

Le reste de cette section traitera des bases de la programmation de ce VFD et des paramètres et valeurs que nous avons modifiés. En commençant par un schéma décrivant l'interface et les boutons les plus importants pour manipuler le variateur.

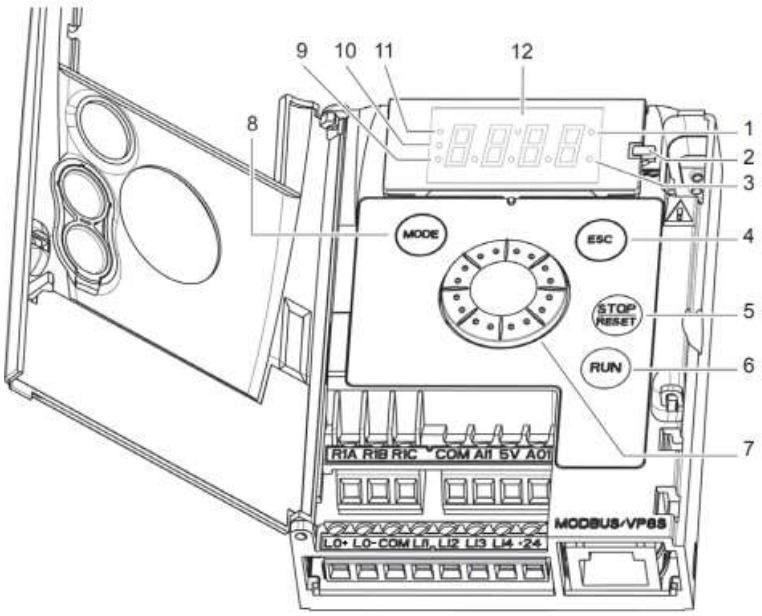


Figure 2. 10 schéma de l'interface VFD

1. DEL de valeur (a) (b). **2.** DEL de charge. **3.** DEL d'unité (c)

4. **Bouton ESC** : permet de quitter un menu ou un paramètre ou de mettre fin à la valeur affichée afin de revenir à la valeur précédente se trouvant en mémoire. En configuration locale, une pression de 2 s sur le bouton ESC commute entre les modes contrôle et programmation.

5. Bouton STOP/RESET : arrête le moteur (peut être dissimulé par la porte si la fonction est désactivée). Nota : voir les instructions sur le retrait du cache de « RUN/STOP ». Permet de réarmer un défaut.

6. Bouton RUN : lance l'exécution en configuration locale et en configuration à distance si la fonction est configurée (peut être dissimulé par la porte si la fonction est désactivée).

7. Bouton de navigation

Agit comme un potentiomètre en configuration locale et en configuration à distance si la fonction est configurée.

- Molette servant à la navigation lorsqu'elle est tournée dans le sens horaire ou anti-horaire.

- et à la sélection / validation par simple pression. cette action est représentée par ce symbole 

8. Bouton MODE

Permet de passer d'un des modes Référence, Surveillance ou programmation à un autre. Une pression de 3 s sur le bouton MODE commute entre les configurations locale et à distance.

9. DEL du mode CONFIGURATION (b)

10. DEL du mode SURVEILLANCE

11. DEL du mode REFERENCE

12. Afficheur 4 fois 7 segments

Configuration utilisée :

Nous passons à la configuration A DISTANCE en maintenant le bouton MODE pendant trois secondes à partir de l'écran rdY.

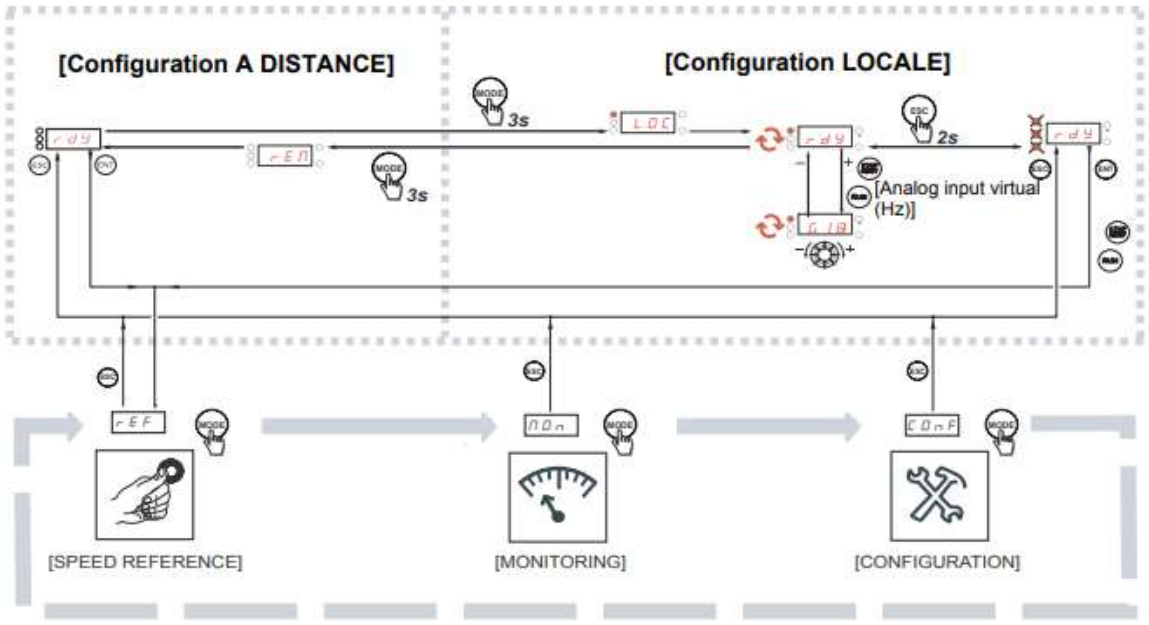


Figure 2. 11 Un diagramme qui montre une mappage détaillé pour naviguer entre les modes

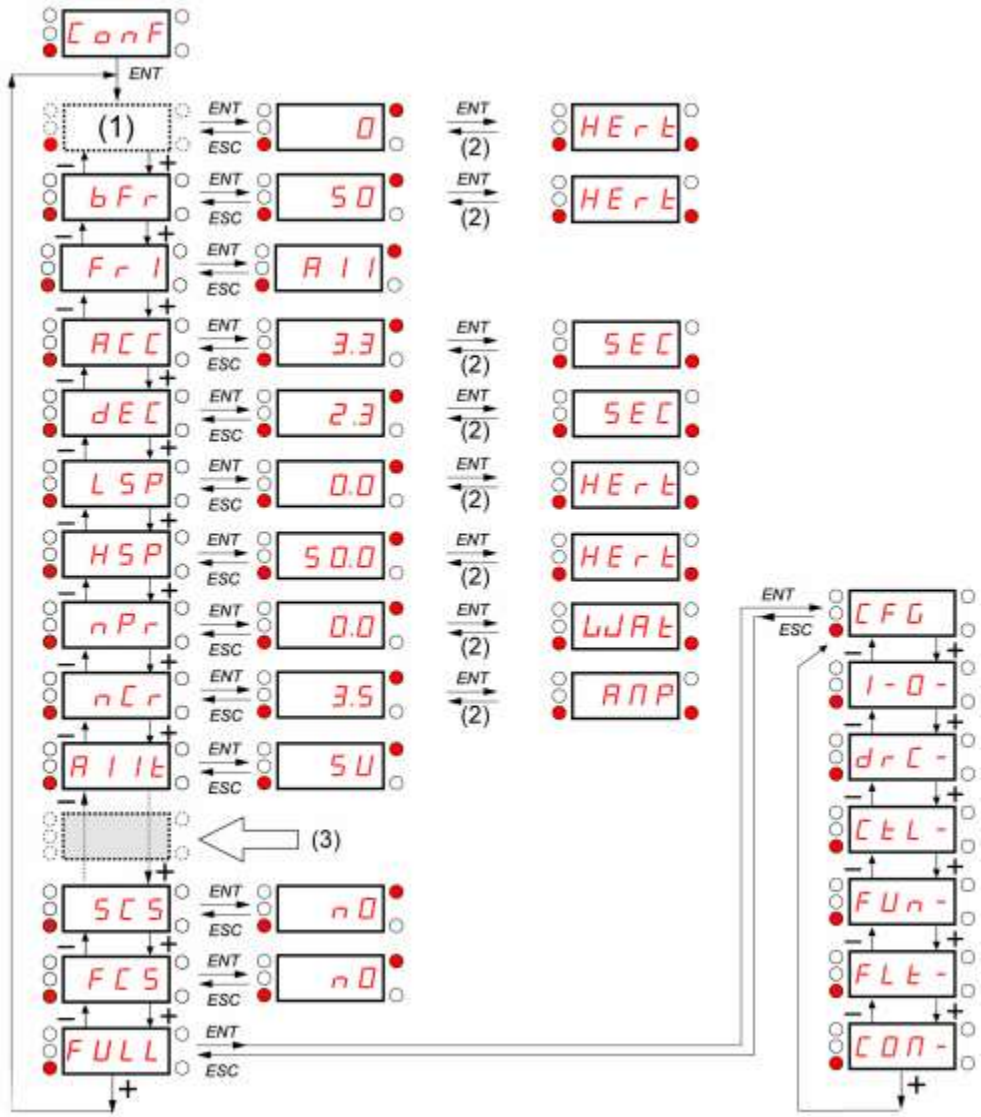
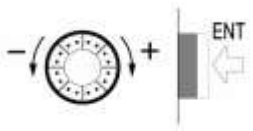


Figure 2. 12 Le diagramme qui montre un mappage pour naviguer entre les paramètres de menu ConF



Nous utilisons Pour la navigation.

C'est une bonne idée de réinitialiser un VFD utilisé aux réglages d'usine avant de l'utiliser.

Pour que nous arrivions à ce paramètre, nous passons par *ConF >> FCS*.

nO: Fonction inactive. FCS passe automatiquement à nO dès que l'une des opérations

suivantes a été effectuée.

rEC1: La configuration actuelle devient identique à la configuration de sauvegarde précédemment enregistrée par SCS. FCS passe automatiquement à nO dès que cette action a été effectuée. rEC1 n'est visible que si la sauvegarde a été effectuée. Si cette valeur s'affiche, InI1 n'est pas visible.

InI: La configuration actuelle devient identique au réglage usine. Si cette valeur s'affiche, InI1 n'est pas visible.

InI1: La configuration actuelle devient identique à la configuration de sauvegarde précédemment définie dans le logiciel SoMove. Si cette valeur s'affiche, Ini et rEC1 ne sont pas visibles.

Nous avons configuré le circuit de commande avec LI1 pour une rotation en avant et LI2 pour une rotation en arrière.

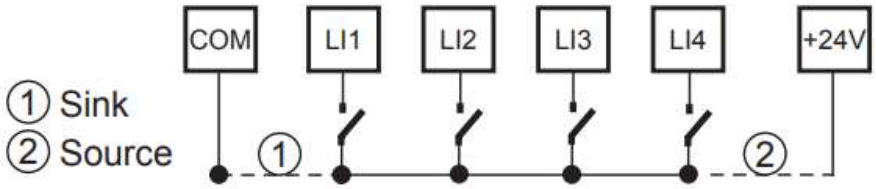


Figure 1. 11 Logic Inputs

Pour ce faire, nous allons au paramètre tCC et le mettons à 2C

ConF > FULL > I_O- > tCC.

2C: Commande à 2 fils, l'état ouvert ou fermé de l'entrée commande la marche ou l'arrêt.

Exemple de câblage « source » :

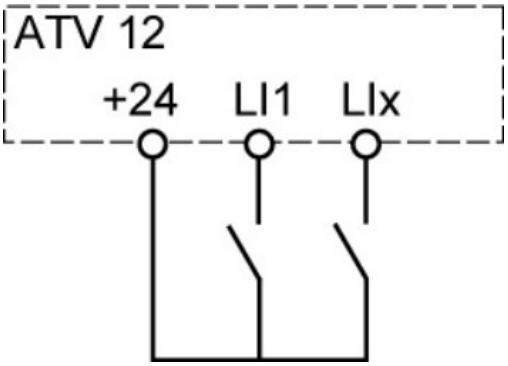


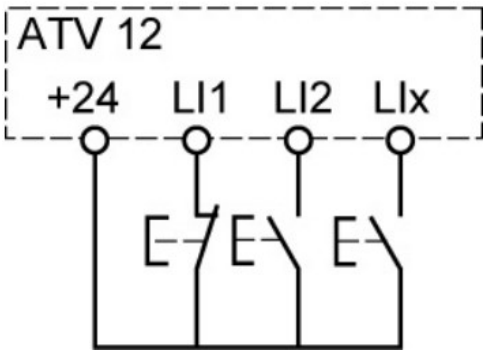
Figure 1. 12 Logic Input Configuration

LI1 : avant

LIx : arrière

3C: Commande à 3 fils, une impulsion « avant » ou « arrière » suffit pour commander le démarrage et une impulsion « arrêt » suffit pour commander l'arrêt.

Exemple de câblage « source » :



LI1 : arrêt

LI2 : avant

LIx : arrière

Pour configurer LIx, allez au paramètre rrS.

ConF> FULL> FUn->rrS.

LI1 à LI4 : choix de l'entrée attribuée à la marche arrière

nO: v Fonction désactivée

LIH: L1 actif à l'état haut

L2H: L2 actif à l'état haut

L3H: L3 actif à l'état haut

L4H: L4 actif à l'état haut

Le signal de sortie de l'API qui contrôle la vitesse du moteur est connecté à AI 1 et délivre une tension de 0 à 10 V.

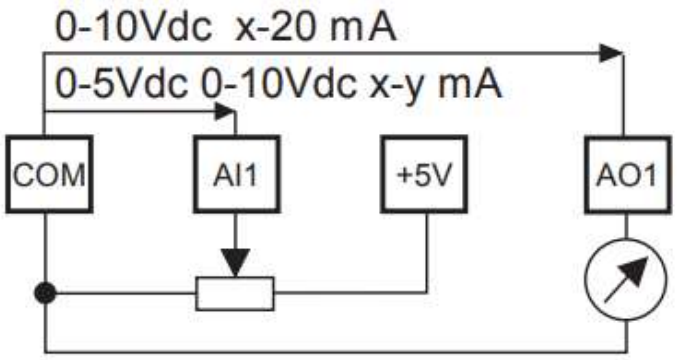


Figure 2. 13 Entre analogique de variateur de vitesse

pour cela nous devons choisir un type d'entrée approprié pour AI1.

Pour configurer le type Ai1, nous naviguons vers AI1t**.** *ConF > FULL > I_O- > AI1- > AI1t.*

Cette fonction sert d'interface entre le signal d'entrée analogique et la valeur interne du variateur.

5U: v Tension : 0-5 V

10U: v Tension : 0-10 V

0A: Courant : x-y mA. Plage déterminée par les paramètres Valeur mini AI1 CrL1 et Valeur maxi AI1 CrH1.

pour améliorer la précision et le temps de réponse des opérations effectuées via le VFD,

nous réduisons le temps d'accélération et de décélération à l'aide des paramètres trouvés dans les paramètres du VFD.

ConF> FULL > Fun- > rPt- > ACC.

Temps d'accélération entre 0 Hz et la Fréq. nom. mot FrS. Assurez-vous que cette valeur est compatible avec l'inertie entraînée.

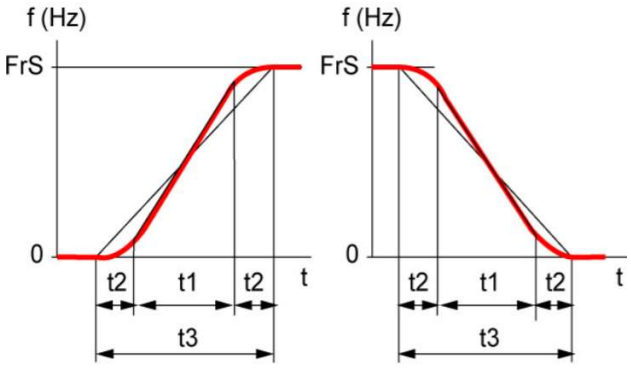
ConF> FULL > Fun- > rPt- > dEC.

Temps de décélération de la Fréq. nom. mot FrS à 0 Hz. Assurez-vous que cette valeur est compatible avec l'inertie entraînée

rPt: pour changer la rampe d'accélération ou de décélération.

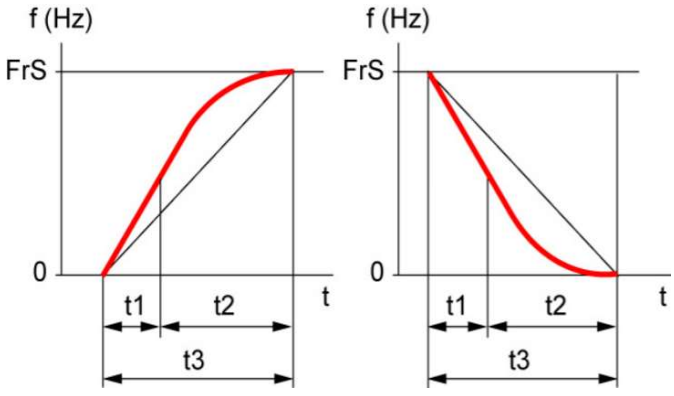
Lin : Linéaire

S : en S



Le coefficient d'arrondissement est fixe, t1 = 0,6 fois temps rampe réglé (linéaire) t2 = 0,4 fois temps rampe réglé (arrondi) t3 = 1,4 fois temps rampe réglé

U : en U



Le coefficient d'arrondissement est fixe, t1 = 0,5 fois temps rampe réglé (linéaire) t2 = temps rampe réglé (arrondi) t3 = 1,5 fois temps rampe réglé

2.6 Module de sorties analogiques

Pour contrôler la vitesse du moteur à travers VFD et à partir d'un signal provenant de l'automate programmable, nous devons lui envoyer un signal analogique. Etant donné que l'API n'est pas en mesure de produire une sortie analogique, nous avons besoin d'un module de sortie analogique pour cela.

Le module de sortie analogique convertit la valeur d'un TAG établi dans l'API en un signal anlogique sous forme de tension ou de courant.

Module de sorties analogiques SM 332 ; AO 2 x 12 bits

Nº de référence : 6ES7332-5HB01-0AB0

Figure 2. 14 représentation de module analogique

Propriétés :

● 2 sorties dans un groupe

● Les sorties sont sélectionnables voie par voie en tant que

– Sortie de tension

– sortie de courant

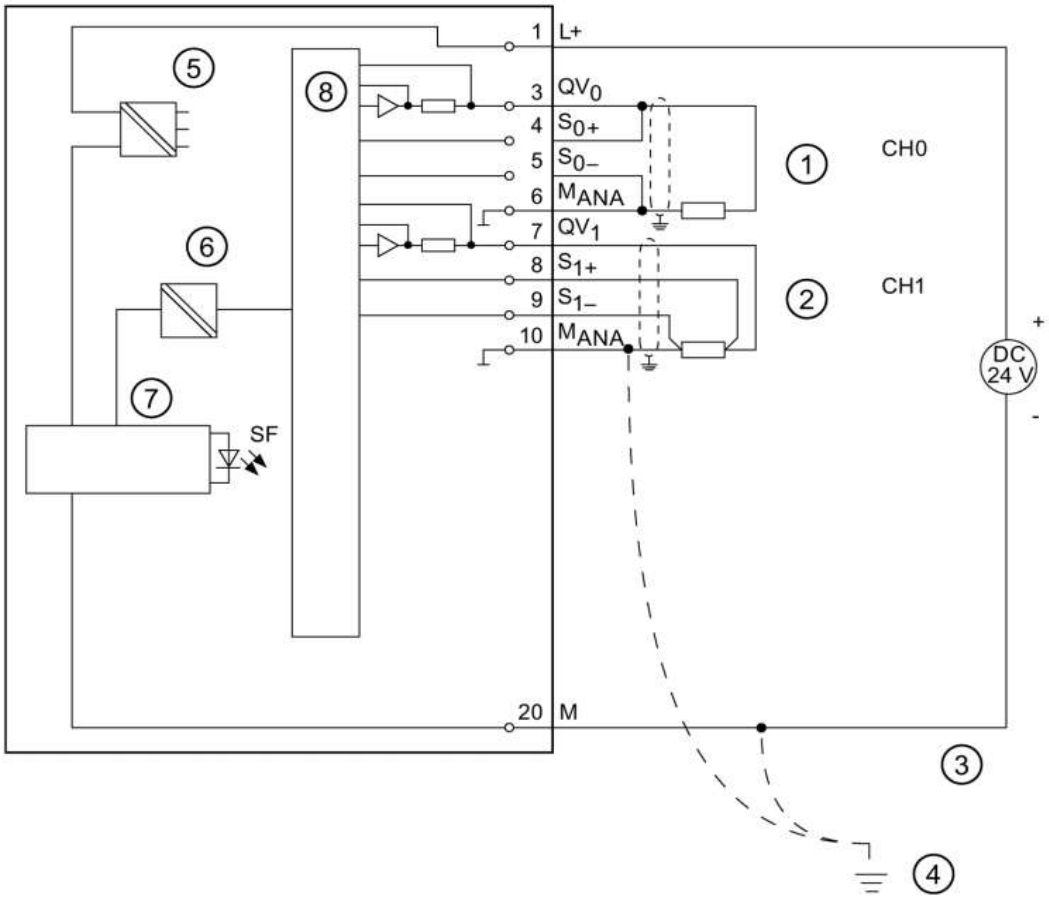
● résolution 12 bits

● Diagnostic paramétrable et alarme de diagnostic

● Avec séparation galvanique par rapport au couplage de bus interne et à la tension D'alimentation

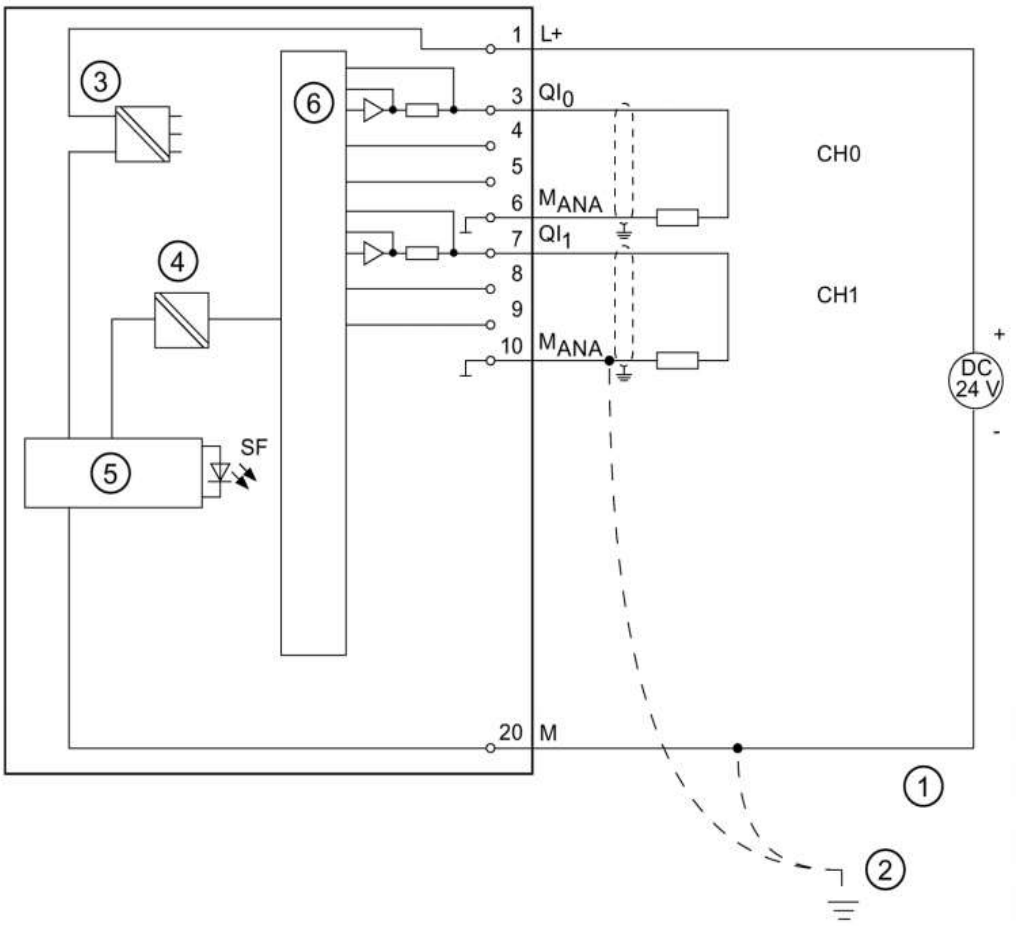
● prend en charge la fonction reparamétrage en MARCHE

Le circuit de sortie d'un signal sous forme de tension (Figure 2.13) ou de courant (Figure 2.14) est présenté ci-dessous.



① Montage 2 fils : sans compensation de la résistance de câble
② Montage 4 fils : avec compensation de la résistance de câble
③ Equipotentialité
④ Terre fonctionnelle
⑤ Alimentation interne
⑥ Séparation de potentiel
⑦ Coupleur de bus interne
⑧ Convertisseur numérique-analogique

Figure 2. 15 montage 2 et 4 fils pour sortie de tension [19]



① Equipotentialité
② Terre fonctionnelle
③ Alimentation interne
④ Séparation de potentiel
⑤ Coupleur de bus interne
⑥ Convertisseur numérique-analogique

Figure 2. 16 le connecteur de sortie de courent

2.7 Capteur Photoélectrique

Est un équipement utilisé pour mesurer la distance, l'absence ou la présence d'un objet à l'aide d'un émetteur de lumière, souvent infrarouge, et d'un récepteur photoélectrique.

Nous utilisons un capteur photoélectrique pour diverses utilisations.

L'initialisation de la position du chariot. Le repositionnement du chariot. Corriger les erreurs et détecter les dangers.

Figure 2. 17 Photocellule FAR2/BP-OE

Un détecteur photoélectrique réalise la détection d'une cible (objet ou personne) au moyen d'un faisceau lumineux. Ses deux constituants de base sont un émetteur et un récepteur de lumière.

Les caractéristiques :

1-un détecteur infrarouge émetteur et récepteur dans le même boitier.

2-la série FA.

3-délivrant un courant maximal de 100 mA et dont la distance de détection ne dépasse pas 100mm.

4- alimenté par une tension de 10V à 30V [13].

Le principe de fonctionnement

En l'absence de cible, le faisceau lumineux n'arrive pas sur le récepteur. Quand une cible pénètre dans le faisceau, elle renvoie ce derniersur le récepteur :

Lumière sur le récepteur = détection

Composé de quatre files **:**

-Deux fils d’alimentation.

-Deux fil pour le signal (signal et signal signal inversé)

2.8 Codeur rotative incrémental

Nous avons utilisé un encodeur pour nous informer de la position du chariot. L'encodeur est un type de capteur pour fournir des informations sur sa rotation.

Il contient trois voies, une pour les signaux A, B et une indiquant un tour complet. Pour notre codeur, les voies A et B ont une précision de 1,44 degrés, ce qui signifie qu'il délivre 250 impulsions pour une révolution complète dans A et B chaque

Figure 2. 18 Hohner 10-1x941-250.an01

Il consiste de cinq fils de connexion dont la signification et le brochage et donner dans le tableau :

A	Bleu	Signal
B	Maron	---
C	Blanc	---
D	Noir	VCC
F	Gris	Masse

Tableau 2. 3 Brochage incrémental de l'encodeur

2.8.1 Principe de fonctionnement

Pour connaître la direction du codeur, vous pouvez vérifier l'état d'une piste puis vérifier l'état de la piste correspondante.

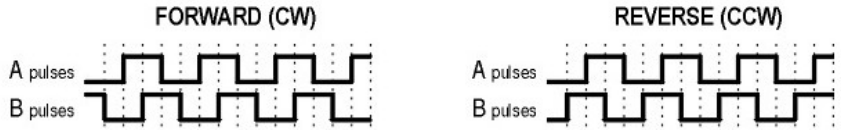


Figure 2. 19 le Principe de fonctionnement d'un codeur

Dans l'exemple de gauche, lorsque le signal A est positif, la piste B correspondante est à 0, ce qui suggère une rotation en avant (sens horaire).

Dans l'exemple à droite, lorsque le signal A est sur un front positif, la piste B correspondante est à 1, ce qui suggère une rotation arrière (sens inverse des aiguilles d'une montre)

Si vous appliquez cela en commençant par le signal B, vous devez basculer entre les résultats [20].

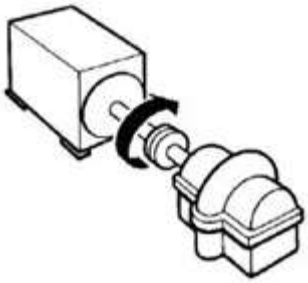


Figure 2. 20 Rotation horaire

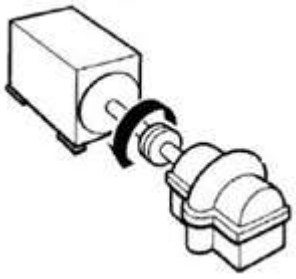


Figure 2. 21 Rotation antihoraire

(Le moteur est en haut à gauche)

Chapitre 3 Développement dans le Soft

3.1 Introduction

Il est maintenant plus facile de développer pour les opérations industrielles en utilisant le logiciel Totally Integrated Automation de Siemen

TIA est fondée sur des solutions pour l'ingénierie intégrée, la communication industrielle, la gestion de données industrielles, la sécurité industrielle, la sécurité intégrée

Ce chapitre vise à donner une vision plus claire des concepts relatifs à la programmation et à la conception IHM du travail.

3.2 Initiations sur la programmation du travail

Cette section traitera de certains concepts nécessaires au développement dans l'espace logiciel Siemens, qui ont tous été utilisés dans le développement de notre projet.

3.2.1 Blocs d'organisation

Les blocs d'organisation (OB) représentent l'interface entre le système d'exploitation et le programme utilisateur. Une tâche précise incombe à chaque bloc d'organisation. Vous pouvez composer le programme utilisateur CONT de votre CPU S7 des blocs d'organisation (OB) dont vous avez besoin pour votre solution d'automatisation. Dans le cas le plus simple, il s'agit des blocs d'organisation destinés : à la mise en route (OB100, OB101), au programme principal cyclique (OB1) et au traitement des erreurs (OB80 à OB87, OB121, OB122), dans le cas où une erreur ne doit pas entraîner l'arrêt de votre CPU. Il existe d'autres blocs d'organisation vous permettant de traiter des alarmes de la CPU ou du processus. Vous pouvez programmer chaque bloc d'organisation en tant que programme structuré en créant des fonctions (FC) et des blocs fonctionnels (FB) et

en appelant ceux-ci dans la section d'instructions. Lors de l'appel de bloc, vous devez fournir des données correspondant aux paramètres déclarés.

a. blocs d'organisation utilisés

Programme cyclique (OB 1)

Le système d'exploitation de la CPU S7 traite l'OB 1 en mode cyclique : Une fois que le traitement de l'OB 1 est achevé, le système d'exploitation recommence son traitement. Le traitement cyclique de l'OB 1 débute à la fin de la mise en route. Vous pouvez appeler des instructions dans l'OB 1.

Démarrage complet ou à chaud (OB 100) [16]

C'est le plus courammant utilisé,il est compatible avec tous les types de processeurs,seules les données rémanentes sont conservées.Les variables qui ont été configurées comme rémanentes conservent donc leur dernière valeur,les autres variables ou blocs non rémanentes sont réinitialisées.Un OB100 est généralement utilisé pour initialiser des variables ou modules d'E/S au démarrage du programme.

3.2.2 Introduction en les FC et FB

a. Fonctions (FC)

Les fonctions (Conformément à CEI 1131-3, une fonction (FC) est un bloc de code sans mémoire. Elle permet de transmettre des paramètres dans le programme utilisateur. C'est la raison pour laquelle les fonctions conviennent à la programmation de fonctions complexes récurrentes, telles que les calculs.) (FC) sont des blocs de code sans mémoire. Elles n'ont pas de mémoire de données dans laquelle il est possible d'enregistrer les valeurs de paramètres de bloc. C'est pourquoi des paramètres effectifs doivent être fournis à tous les paramètres formels lors de l'appel d'une fonction.

Pour enregistrer les données durablement, les fonctions disposent de blocs de données (Zone de données du programme utilisateur contenant des données utilisateur. Il existe des blocs de données globaux auquel il est possible d'accéder à partir de n'importe quel bloc de code et des blocs de données d'instance affectés à un appel de FB spécifique.) globaux.

Domaine d'application

Une fonction contient un programme qui est exécuté lorsque la fonction est appelée par un autre bloc de code. Les fonctions peuvent par exemple servir dans les cas suivants :

- Retourner des valeurs de fonction au bloc appelant, par ex. pour les fonctions mathématiques
- Exécuter des fonctions technologiques, par ex. commandes uniques avec combinaisons binaires

Une fonction peut être appelée plusieurs fois à différents endroits d'un programme. Ainsi vous simplifiez la programmation de fonctions utilisées fréquemment.

b. Blocs fonctionnels (FB)

Les blocs fonctionnels sont des blocs de code qui mémorisent durablement leurs paramètres d'entrée, de sortie et d'entrée/sortie dans des blocs de données d'instance afin qu'il soit possible d'y accéder même après le traitement de blocs. C'est pourquoi ils sont également appelés "Blocs avec mémoire".

Les blocs fonctionnels peuvent aussi travailler avec des variables temporaires. Cependant, les variables temporaires ne sont pas enregistrées dans la DB d'instance mais disponibles uniquement tout le temps d'un cycle.

Domaine d'application

Les blocs fonctionnels contiennent des sous-programmes qui sont exécutés lorsqu'un bloc fonctionnel est appelé par un autre bloc de code. Un bloc fonctionnel peut être appelé plusieurs fois à différents endroits d'un programme. Ainsi vous simplifiez la programmation de fonctions utilisées fréquemment.

Instances de blocs fonctionnels

L'appel d'un bloc fonctionnel est une instance. Chaque instance d'un bloc fonctionnel requiert un bloc de données d'instance dans lequel sont mémorisées les valeurs spécifiques de l'instance des paramètres formels déclarés dans le FB.

Le bloc fonctionnel peut mémoriser ses données spécifiques aux instances dans un bloc de données d'instance dédié ou dans un bloc de données d'instance du bloc appelant.

Types d'accès

S7-1200 et S7-1500 offrent deux possibilités d'accès différentes pour les blocs de données d'instance rattachés à un bloc fonctionnel lors de l'appel :

- Des blocs de données avec un accès optimisé

  Les blocs de données avec accès optimisé n'ont pas de structure de mémoire fermement définie. Dans la déclaration, les éléments de données ne reçoivent qu'un nom symbolique et pas d'adresse fixe dans le bloc.

- Blocs de données à accès standard (compatible avec S7-300/400)

  Les blocs de données avec accès standard ont une structure de mémoire fixe. Dans la déclaration, les éléments de données reçoivent aussi bien un nom symbolique qu'une adresse fixe dans le bloc.

3.2.3 Langues de programmation utilisées

a. Langage de programmation CONT (Ladder)

Le schéma à contacts CONT est un langage de programmation graphique. La syntaxe de ses instructions s'inspire des schémas à relais : CONT permet de suivre facilement le flux d'énergie circulant via des entrées, des sorties et des opérations entre les barres d'alimentation.

Le langage de programmation CONT met à votre disposition tous les éléments nécessaires à la création d'un programme utilisateur complet. Il dispose d'un jeu d'opérations très important. Vous disposez de diverses opérations de base différentes ainsi que d'une large palette d'opérandes et d'adressages. Cela vaut également pour le concept des fonctions et des blocs fonctionnels qui vous permettent de structurer clairement un programme CONT.

b. Langage de programmation SCL (Structured Control Language)

SCL (Structured Control Language) est un langage de programmation évolué proche du PASCAL, conforme à une norme sur les automates programmables. La norme DIN EN 61131-3 (qui correspond à la norme internationale CEI 1131-3) normalise les langages de programmation destinés aux automates programmables. La partie traitant du « texte structuré » constitue le fondement de SCL. Vous trouverez plus de détails à ce sujet dans la table de correspondance à la norme dans le fichier NORM_TBL.WRI (anglais) ou NORM_TAB.WRI (allemand) de STEP 7. Outre les éléments du langage évolué, SCL dispose également d'éléments de langage spécifiques aux automates programmables, comme les entrées, sorties, temporisations, mémentos, appels de blocs etc.. SCL complète et élargit ainsi le spectre du logiciel de programmation STEP 7 avec ses langages de programmation CONT, LOG et LIST.

Pour ajouter un bloc SCL dans le dossier des blocs de votre programme :

Pour ouvrir le menu des blocs :

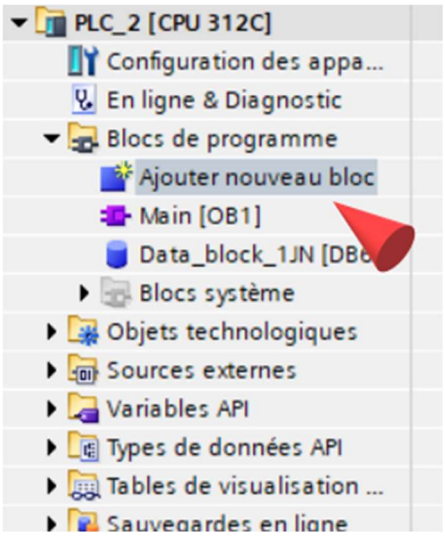


Figure 3. 1 Ajouter nouveau Bloc

Pour choisir le type de bloc spécifique :

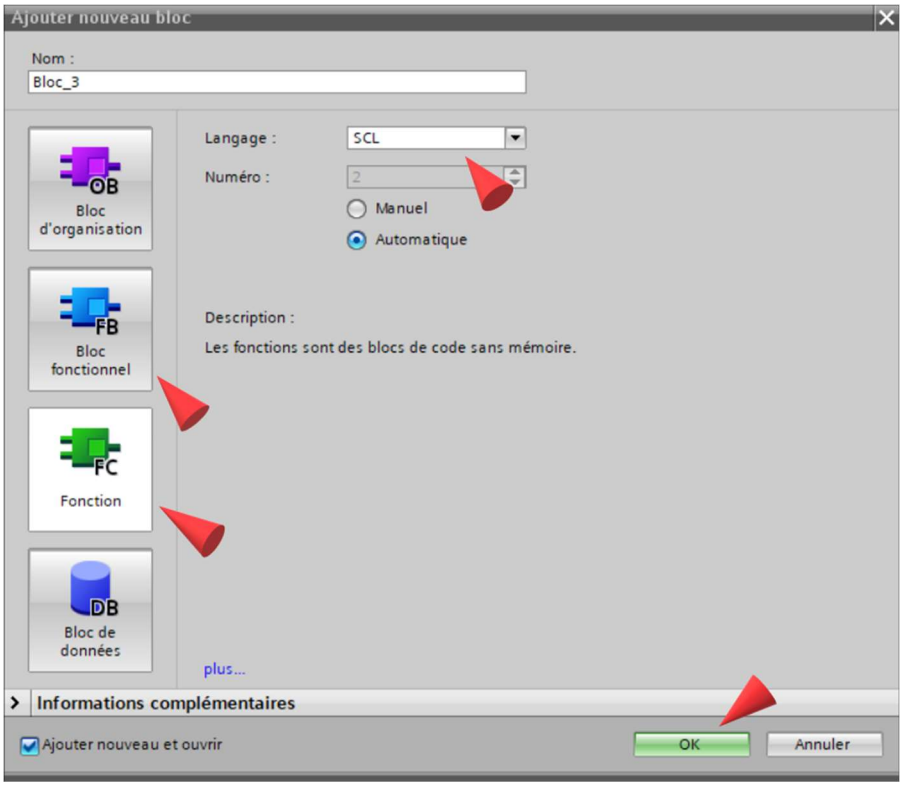


Figure 3. 2 Ajouter FC bloc

Les blocs SCL peuvent être une fonction ou un bloc de fonction.

Le lien ci-dessous est destiné à un aide-mémoire SCL très pratique.

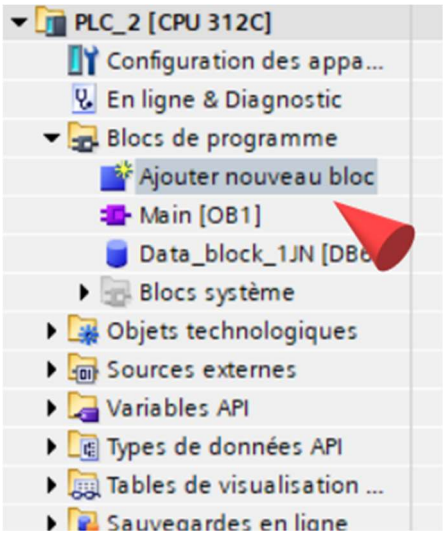
http://plc4good.org.ua/files/03_downloads/SCL_table/SCL-cheat-sheet.pdf

c. Langage de programmation GRAPH

Le langage de programmation S7-GRAPH s'ajoute à l'éventail des fonctions de STEP 7. Il permet de programmer graphiquement les commandes séquentielles. Avec S7-GRAPH, vous programmerez aisément et rapidement des commandes séquentielles que vous souhaitez piloter avec un système d'automatisation SIMATIC. Le processus est subdivisé en étapes au nombre de fonctions limité, la séquence est représentée graphiquement et peut être documentée par des vues et des textes. Les actions à exécuter sont associées aux étapes, tandis que des transitions règlent l'évolution entre deux étapes successives (réceptivités). Pour définir les réceptivités des transitions ainsi que les verrouillages ou les surveillances d'étape, vous aurez à utiliser un nombre restreint d'éléments du langage de programmation CONT (schéma à contacts) ou LOG (logigramme).

Pour ajouter un bloc GRAPH dans le dossier des blocs de votre programme :

Pour ouvrir le menu des blocs :



Pour choisir le type de bloc spécifique :

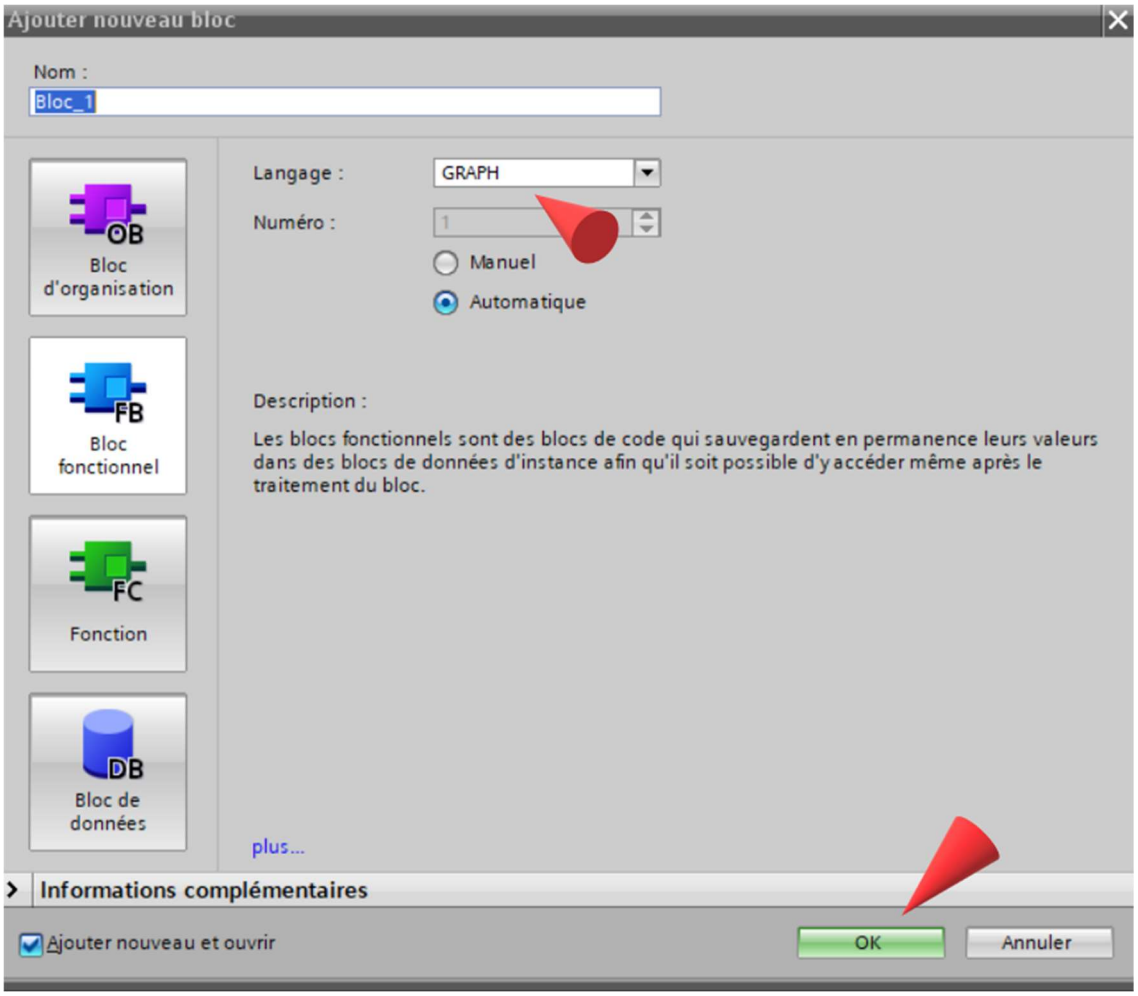


Figure 3. 3 Ajouter FB bloc

Lorsque vous ouvrez ce bloc, vous êtes accueilli par cet écran :

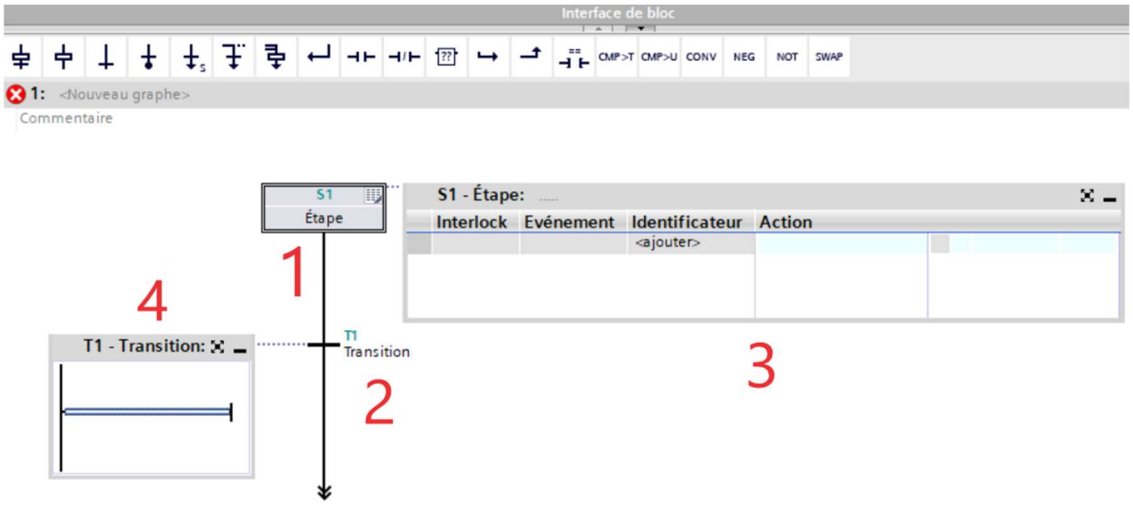


Figure 3. 4 GRAPH Interface

Semblable à un grafcet, (1) est une étape ; à l'ouverture comme à droite (3), vous pouvez effectuer des actions telles que l'appel de fonctions ou la modification d'états de bits. L'identificateur est important car il décide de la manière dont vous voulez exécuter l'action.

(2) est une transition, c'est un réseau Ladder (4). Quand il est fermé, la transition permet de passer à l'étape suivante, bien sûr, en ce qui concerne les règles GRAPH.

Cette section présente les composants à construire dans le bloc GRAPH.

Lorsqu'une transition est ouverte, « Favoris » étend sa sélection pour afficher les blocs et les instructions que vous pouvez ajouter à un réseau de transition.

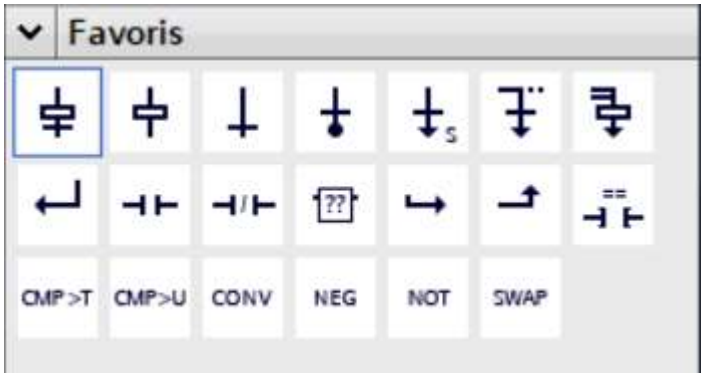


Figure 3. 5 GRAPH Favoris

3.2.4 Initiation sur quelque bloc utilisée

MOVE : Copier valeur

L'instruction "Copier valeur" vous permet de transférer le contenu de l'opérande à l'entrée IN dans l'opérande à la sortie OUT1. Le transfert s'effectue toujours dans le sens croissant des adresses. Ce bloc est très polyvalent et nous l'avons utilisé de nombreuses façons.

SCALE : Mise à l'échelle (FC 105)

Avec l'instruction "Mise à l'échelle", vous convertissez l'entier indiqué au paramètre IN en un nombre à virgule flottante qui est mis à l'échelle en unités physiques entre une valeur limite inférieure et une valeur limite supérieure. Vous définissez la valeur limite inférieure et supérieure de la plage de valeurs sur laquelle la valeur d'entrée est mise à l'échelle par le biais des paramètres LO_LIM et HI_LIM. Le résultat de l'instruction est fourni au paramètre OUT.

L'instruction "Mise à l'échelle" utilise l'équation suivante :

OUT = [((FLOAT (IN) – K1)/(K2–K1)) ∗ (HI_LIM–LO_LIM)] + LO_LIM

Les valeurs des constantes "K1" et "K2" sont déterminées par l'état logique du paramètre BIPOLAR. Le paramètre BIPOLAR peut prendre les états logiques suivants :

- Etat logique "1" : on suppose que la valeur du paramètre IN est bipolaire et se situe dans une plage de valeurs allant de -27648 à 27648. Dans ce cas, la constante "K1" a la valeur "-27648,0" et la constante "K2" la valeur "+27648,0".
- Etat logique "0" : on suppose que la valeur du paramètre IN est unipolaire et se situe dans une plage de valeurs allant de 0 à 27648. Dans ce cas, la constante "K1" a la valeur "0,0" et la constante "K2" la valeur "+27648,0".

Quand la valeur du paramètre IN est supérieure à la valeur de la constante "K2", le résultat de l'instruction prend la valeur de la limite supérieure (HI_LIM) et une erreur est renvoyée.

Quand la valeur du paramètre IN est inférieure à la valeur de la constante "K1", le résultat de l'instruction prend la valeur de la limite inférieure (LO_LIM) et une erreur est renvoyée.

Quand la limite inférieure indiquée est supérieure à la limite supérieure (LO_LIM > HI_LIM), le résultat est mis à l'échelle de manière inversement proportionnelle à la valeur d'entrée.

UNSCALE : Annuler mise à l'échelle (FC 106)

Avec l'instruction "Annuler mise à l'échelle", vous annulez la mise à l'échelle en unités physiques entre une valeur limite inférieure et une valeur limite supérieure du nombre à virgule flottante au paramètre IN et vous le convertissez en nombre entier. Vous définissez la valeur limite inférieure et supérieure de la plage de valeurs sur laquelle la mise à l'échelle de la valeur d'entrée est annulée par le biais des paramètres LO_LIM et HI_LIM. Le résultat de l'instruction est fourni au paramètre OUT.

L'instruction "Annuler mise à l'échelle" utilise l'équation suivante :

$$OUT = [((IN–LO\_LIM)/(HI\_LIM–LO\_LIM)) * (K2–K1)] + K1$$

Les valeurs des constantes "K1" et "K2" sont déterminées par l'état logique du paramètre BIPOLAR. Le paramètre BIPOLAR peut prendre les états logiques suivants :

- Etat logique "1" : on suppose que la valeur du paramètre IN est bipolaire et se situe dans une plage de valeurs allant de -27648 à 27648. Dans ce cas, la constante "K1" a la valeur "-27648,0" et la constante "K2" la valeur "+27648,0".
- Etat logique "0" : on suppose que la valeur du paramètre IN est unipolaire et se situe dans une plage de valeurs allant de 0 à 27648. Dans ce cas, la constante "K1" a la valeur "0,0" et la constante "K2" la valeur "+27648,0".

Lorsque la valeur dans le paramètre IN est supérieure à la valeur de la limite supérieure ("HI_LIM"), le résultat de l'instruction prend la valeur de la constante "K2" et une erreur est renvoyée.

Lorsque la valeur dans le paramètre IN est inférieure à la valeur de la constante de la limite inférieure (LO_LIM), le résultat de l'instruction prend la valeur de la constante "K1" et une erreur est émise.

COUNT_300C (SFB 47) et configuration de compteur rapide.

Étant donné qu'à une vitesse élevée, le codeur pourrait délivrer un signal plus souvent que le programme normal du processeur ne l'enregistrait, il est préférable d'utiliser le HSC du processeur.

Le compteur intégré des CPU S7 31xC est piloté par le bloc de fonction système SFB 47. En plus des entrées, pour piloter le compteur, il peut être paramétré par le SFB 47 (par ex. hystérésis, valeur de comparaison) ou interrogation des données paramétrées.

Le compteur sera paramétré par une tâche (numéro de tâche) et les valeurs correspondantes. Ensuite suivra l'activation de la tâche. Une nouvelle tâche ou un nouveau paramètre peut seulement être positionné ou interrogé lorsque la tâche précédente a été traitée. La fin d'une tâche est indiquée par le registre d'état.

Paramètres d'entrées	**Type**	**Description**
Modul_adress	WORD	Adresse d'E/S du module compteur en hexadécimal, comme paramétrée dans la configuration matérielle HW-Config (Valeur par défaut : W#16#300)
Channel_number	INT	Numéro du canal utilisé. Le nombre de canaux disponibles est dépendant de la CPU
SW_Gate	BOOL	Commande des Tors logiciel
Enable_output	BOOL	Validation de la sortie pour commande manuelle
Control_output	BOOL	Commande directe des sorties digitales
Paramètres de sorties	**Type**	**Description**
STS_Gate	BOOL	Affichage d'état des Tores internes
STS_Up	BOOL	Sens de comptage est en avant
STS_Down	BOOL	Sens de comptage est en arrière
STS_Latch	BOOL	Affichage d'état de l'entrée de verrouillage
STS_Hardwaregate	BOOL	Affichage d'état des Tors matériel
STS_Output	BOOL	Affichage d'état des sorties digitales
STS_Comperator	BOOL	Condition de comparaison remplie ou a été remplie (Faire attention au réglage dans la configuration matérielle de STEP 7)

STS_Overflow	BOOL	Dépassement haut
STS_Underflow	BOOL	Dépassement bas
STS_Zero_mark	BOOL	Passage à zéro apparu
Countervalue	DINT	Valeur actuelle du compteur
Latch_value	DINT	Valeur de verrouillage actuelle (dernière mesure)
Job_error	BOOL	Affiche une erreur lors d'un tâche de lecture ou d'écriture
Error_number	WORD	Numéro d'erreur de tache (lorsque "Erreur de tache = 1")
Paramètres d'entrées/sorties	**Type**	**Description**
Reset_status	BOOL	Reset du bit d'état de comparaison, de passage à zéro, dépassement haut et bas
WR_Count_value	BOOL	Lancement du tâche "Écriture valeur de comptage"
WR_Load_value	BOOL	Lancement de la tâche "Écriture valeur de chargement"
WR_Comperator_value	BOOL	Lancement de la tâche "Écriture valeur de comparaison"
WR_Hysteresis	BOOL	Lancement de la tâche "Écriture Hystérésis"
WR_Pulse_width	BOOL	Lancement de la tâche "Écriture durée d'impulsion"
WR_Job_value	DINT	Valeur pour chaque tâche d'écriture
RD_Load_value	BOOL	Lancement de la tâche "Lecture valeur de chargement actuelle"
RD_Comperator_value	BOOL	Lancement de la tâche "Lecture valeur de comparaison actuelle"
RD_Hystersis	BOOL	Lancement de la tâche "Lecture valeur d'hystérésis actuelle"
RD_Pulse_width	BOOL	Lancement de la tâche "Lecture valeur duré d'impulsion actuelle"
RD_Read_value	DINT	Valeur de retour de chaque tâche de lecture

Tableau 3. 1 Paramètres de bloc HSC

Raccordement	Nom /Adresse	Compte	Mesure de fréquence	Modulation de largeur d'impulsion
1	-	pas connecté		
2	DI + 0.0	Canal 0: Voie A / impulsion	Canal 0: Voie A / impulsion	-
3	DI + 0.1	Canal 0: Voie B / direction	Canal 0: Voie B / direction	0/ne pas utilizer
4	DI + 0.2	Canal 0: porte de matériel	Canal 0: porte de matériel	Canal 0: porte de matériel
5	DI + 0.3	Canal 1: Voie A / impulsion	Canal 1: Voie A / impulsion	-
6	DI + 0.4	Canal 1: Voie B / direction	Canal 1: Voie B / direction	0/ne pas utilizer
Raccordement	**Nom /Adresse**	**Compte**	**Mesure de fréquence**	**Modulation de largeur d'impulsion**
7	DI + 0.5	Canal 1: porte de matériel	Canal 1: porte de matériel	Canal 1: porte de matériel
8	DI + 0.6	Canal 0: Latch	-	-
9	DI + 0.7	Canal 1: Latch	-	-
10	DI + 1.0	-		
11	DI + 1.1	-		
12	2 M	La masse du chassis		
13	2 L+	Alimentation 24v pour les sorties		
14	DO + 0.0	Canal 0: Sortie	Canal 0: Sortie	Canal 0: Sortie
15	DO + 0.1	Canal 1: Sortie	Canal 1: Sortie	Canal 1: Sortie

Tableau 3. 2 tableau indiquant les connexions physiques du HSC du S7-312C

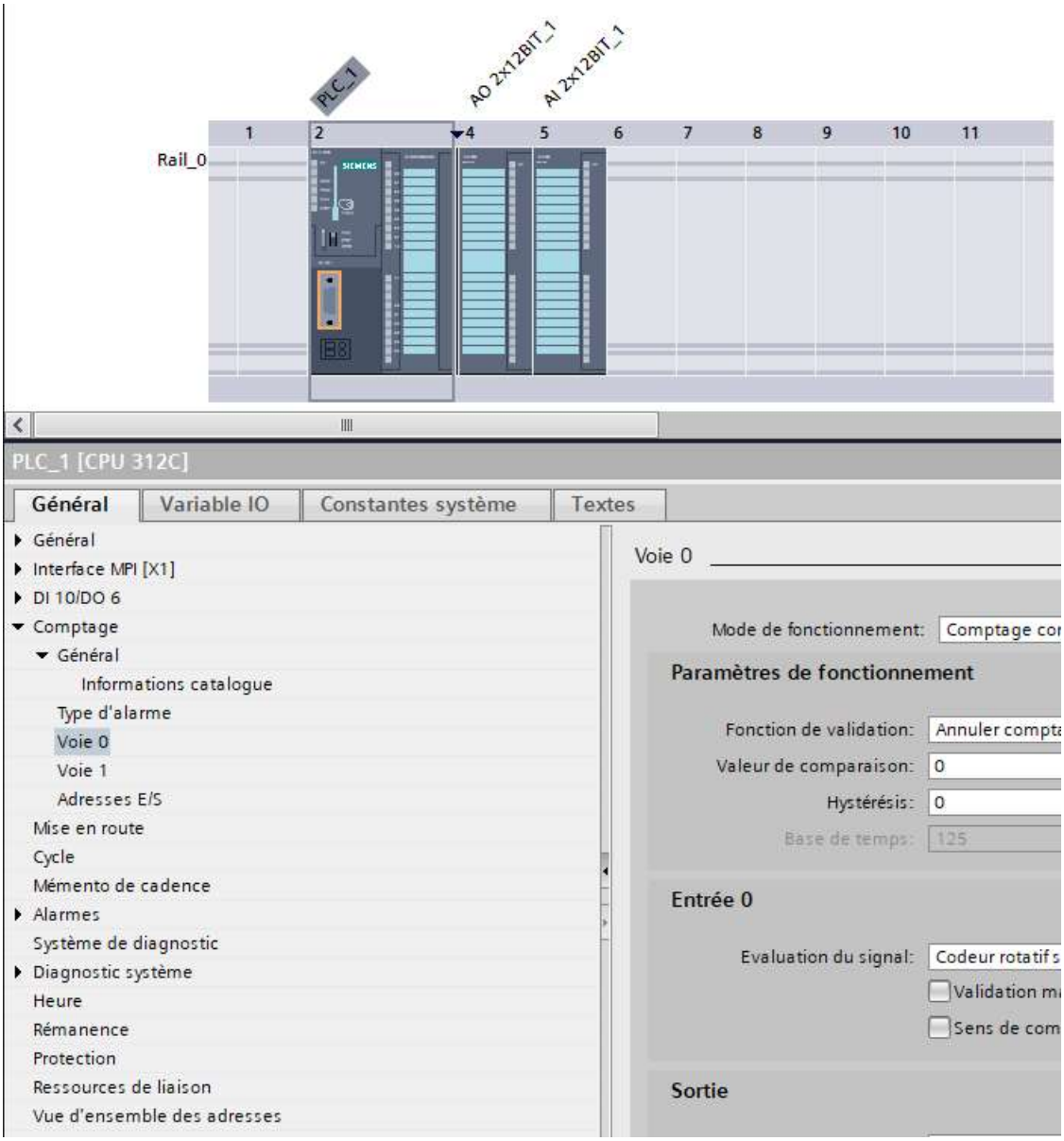


Figure 3. 6 L'accès à l'écran de configuration pour la première piste du HSC

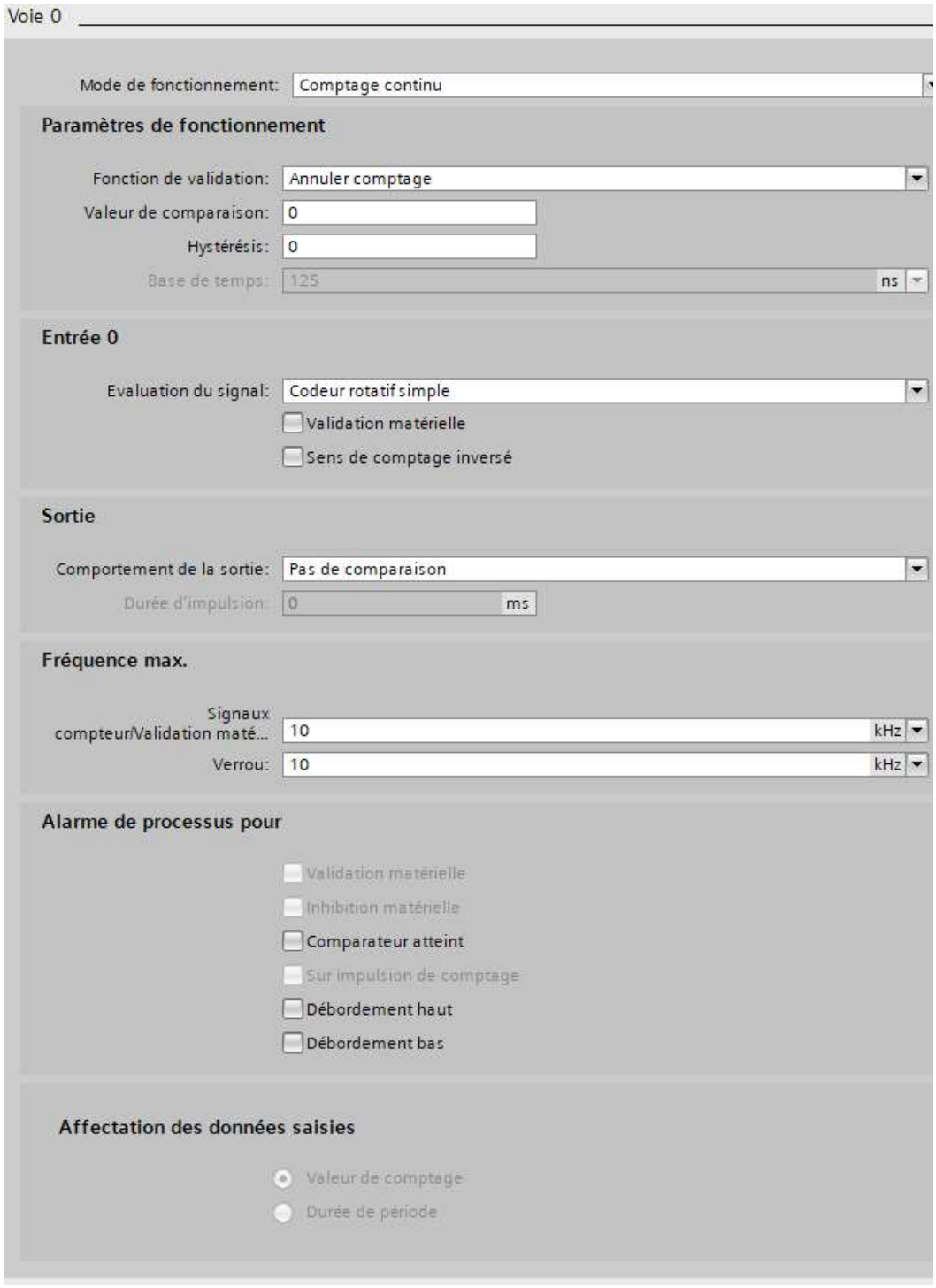


Figure 3.7 Voie 0 configuration

3.3 La programmation de chaque automate

3.3.1 API qui contrôle le chariot

Pour simplifier le diagnostic, le dépannage et la supervision, nous avons conçu le programme de manière à ce qu'il soit facile à comprendre et à superviser sans passer par l'interface IHM, à l'aide de la fonction ONLINE de TIA et d'un seul bloc GRAPH pour résoudre le flux des étapes.

La fonction ONLINE permet de surveiller l'automate

- Débogage des programmes utilisateur
- Affichage et modification du mode de fonctionnement de la CPU
- Affichage et réglage de l'heure et de la date de la CPU
- Affichage des informations sur le module
- Comparaison de blocs en ligne et hors ligne
- Diagnostic du matériel

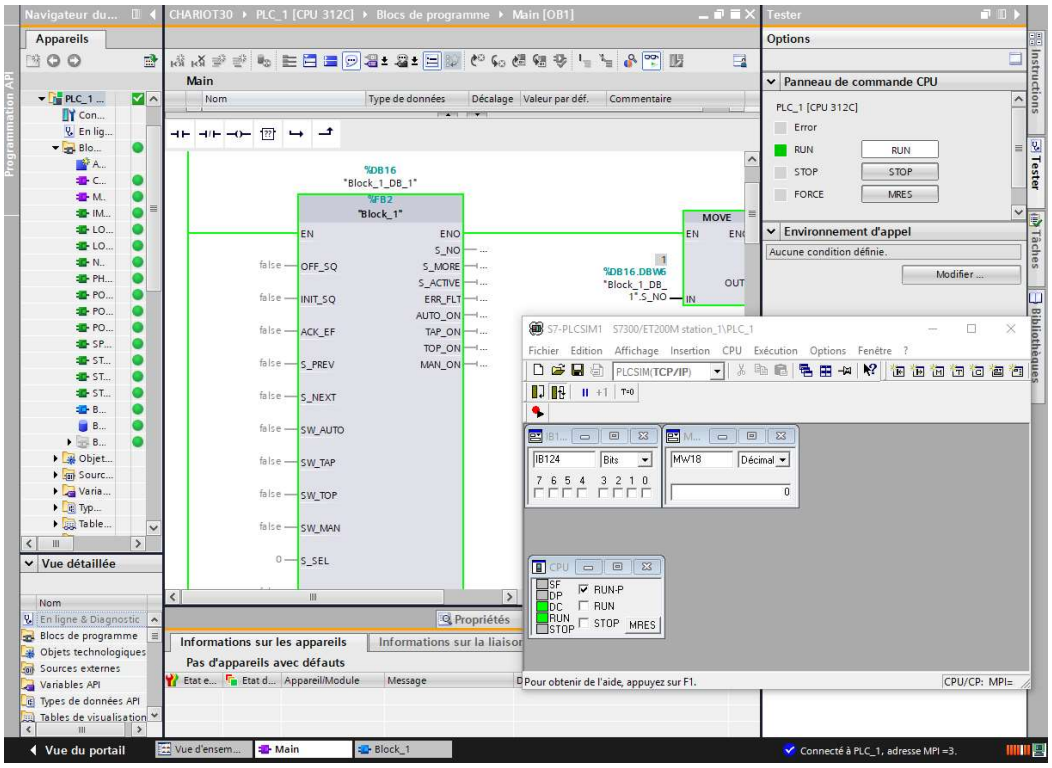


Figure 3. 8 Le programme ladder en mode en ligne

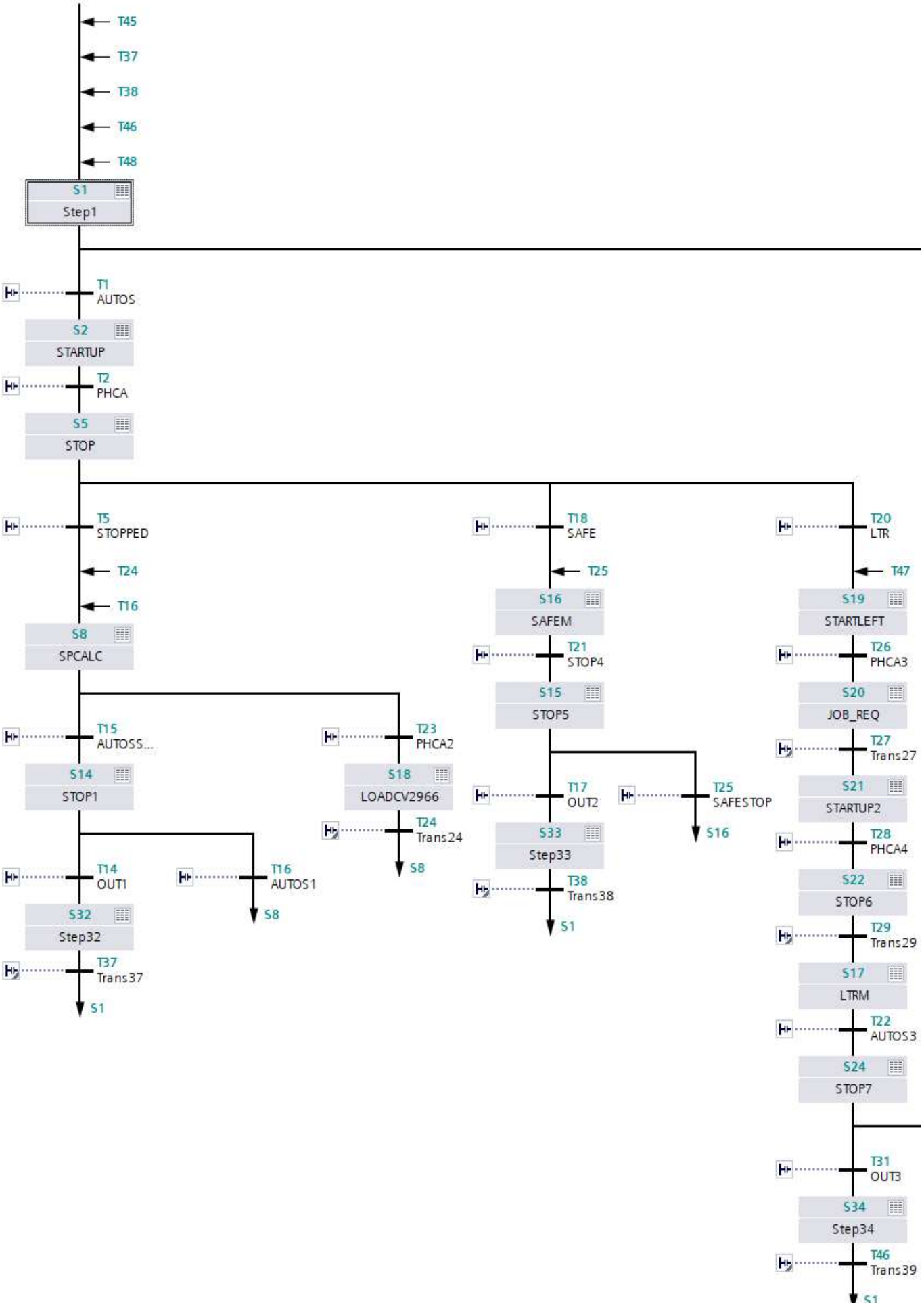
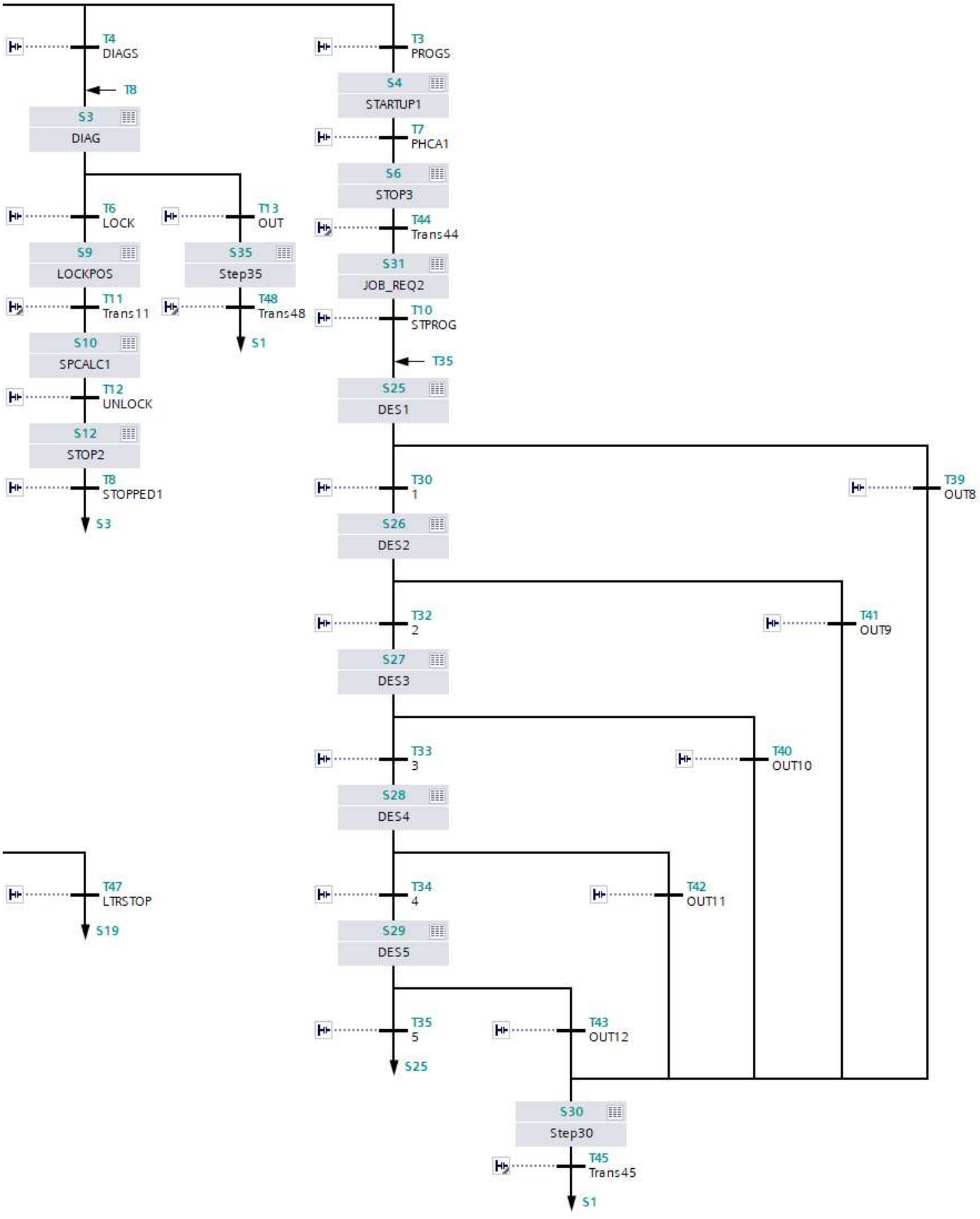


Figure 3. 9 GRAPH Bloc


T4
DIAGS
T8
S3
DIAG
T6
LOCK
T13
OUT
S9
LOCKPOS
S35
Step35
T11
Trans11
T48
Trans48
S1
S10
SPCALC1
T12
UNLOCK
S12
STOP2
T8
STOPPED1
S3
T3
PROGS
S4
STARTUP1
T7
PHCA1
S6
STOP3
T44
Trans44
S31
JOB_REQ2
T10
STPROG
T35
S25
DES1
T30
1
T39
OUT8
S26
DES2
T32
2
T41
OUT9
S27
DES3
T33
3
T40
OUT10
S28
DES4
T47
LTRSTOP
S19
T34
4
T42
OUT11
S29
DES5
T35
5
S25
T43
OUT12
S30
Step30
T45
Trans45
S1

Dans le programme principal, il y a les sorties et quelques blocs de fonctionnalités.

Chaque étape du bloc GRAPH contient un appel pour un ou plusieurs FC, des instructions simples ou les deux. En mode en ligne, lorsqu'une étape est active dans le bloc GRAPH, elle s'allume en vert, ce qui est très utile pour la supervision et le diagnostic.

3.3.2 Le API qui contrôle les feux rouges

Cet appareil contrôle des feux tricolores de circulation, dont le cahier des charges est résumé par le GRAFCET ci-dessous.

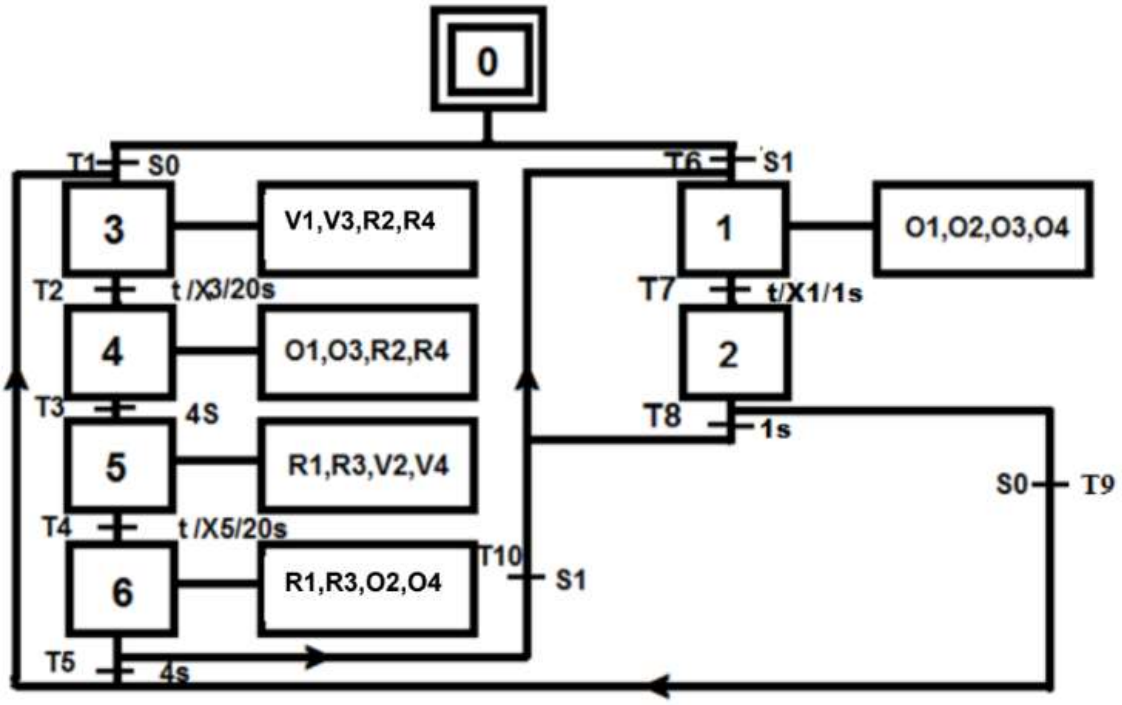


Figure 3. 10 Feux Rouge Grafcet

S0 bouton poussoir pour faire passer le contrôle au mode Jour.

S1 bouton poussoir pour faire passer le contrôle au mode Nuit.

3.4 Les blocs de communication

Il existe une sélection de blocs pour chaque support de communication entre des automates Siemens, les pages suivantes incluent une explication détaillée des blocs compatibles avec la communication que nous avons mise en œuvre.

X_SEND : Envoyer les données à un partenaire de communication hors de la propre station S7 locale

L'instruction "X_SEND" permet d'envoyer des données à un partenaire de communication situé hors de la propre station S7. Le partenaire de communication reçoit les données par le biais de l'instruction "X_RCV". L'envoi de données a lieu après appel de l'instruction avec REQ=1.

Veillez à ce que la zone d'émission définie par le paramètre SD (dans la CPU émettrice) soit inférieure ou égale à la zone de réception définie par le paramètre RD (dans le partenaire de communication). Si SD est de type de données BOOL, RD doit également être de type de données BOOL.

Paramètre	**Déclaration**	**Type de données**	**Zone de mémoire**	**Description**
REQ	Input	BOOL	I, Q, M, D, L	Paramètre de commande "request to activate"
CONT	Input	BOOL	I, Q, M, D, L	Paramètre de commande "continue"
DEST_ID	Input	WORD	I, Q, M, D, L ou constante	Paramètre d'adressage "destination ID". Il contient l'adresse MPI configurée du partenaire de communication.
REQ_ID	Input	DWORD	I, Q, M, D, L ou constante	Code de tâche pour l'identification des données dans le partenaire de communication
SD	Input	ANY	I, Q, M, D	Référence à la zone d'émission. Les types de données suivants sont autorisés : BOOL, BYTE, CHAR, WORD, INT, DWORD, DINT, REAL, DATE, TOD, TIME, S5TIME, DATE_AND_TIME, ainsi que les tableaux des types de

				données mentionnés à l'exception de BOOL. La taille maximale de la zone d'émission est de 76 octets.
RET_VAL	Return	INT	I, Q, M, D, L	Si une erreur s'est produite pendant l'exécution de l'instruction, la valeur en retour contient le code d'erreur correspondant.
BUSY	Output	BOOL	I, Q, M, D, L	BUSY=1 : L'émission n'est pas encore achevée. BUSY=0 : L'émission est achevée ou il n'y a pas d'émission active.

Tableau 3. 3 les paramètres de l'instruction "X_SEND"

X_RCV : Recevoir des données d'un partenaire de communication hors de la propre station S7 locale

L'instruction "X_RCV" permet de recevoir des données qu'un ou plusieurs partenaires de communication situés hors de la propre station S7 ont envoyées au moyen de l'instruction "X_SEND".

L'instruction "X_RCV" vous permet :

- de constater si des données envoyées sont actuellement disponibles ; elles ont éventuellement été rangées dans une file d'attente interne par le système d'exploitation ;
- de copier le bloc le plus ancien figurant dans la file d'attente dans une zone de réception que vous définissez.

Paramètre	**Déclaration**	**Type de données**	**Zone de mémoire**	**Description**
EN_DT	Input	BOOL	I, Q, M, D, L	Paramètre de commande "enable data transfer". Donnez-lui la valeur 0 pour vérifier s'il y a au moins un bloc en attente. La valeur 1 provoque la copie du bloc le plus ancien de la file d'attente dans la zone de la mémoire de travail que vous avez indiquée au moyen de RD.

RET_VAL	Return	INT	I, Q, M, D, L	Si une erreur s'est produite pendant l'exécution de l'instruction, la valeur en retour contient le code d'erreur correspondant. En l'absence d'erreur, RET_VAL contient : • Si EN_DT=0/1 et NDA=0 : W#16#7000. Dans ce cas, il n'y a pas de données dans la file d'attente. • Si EN_DT=0 et NDA=1 : la longueur en octets du bloc le plus ancien dans la file, sous forme de nombre positif Si EN_DT=1 et NDA=1 : la longueur en octets du bloc copié dans la zone de réception RD, sous forme de nombre positif
REQ_ID	Output	DWORD	I, Q, M, D, L	Code de tâche de l'instruction "X_SEND" dont les données émises figurent en première place dans la file d'attente, c'est-à-dire sont les plus anciennes. Si la file d'attente est vide, REQ_ID contient la valeur 0.
NDA	Output	BOOL	I, Q, M, D, L	Paramètre d'état "new data arrived". NDA=0: • Il n'y a pas de données dans la file d'attente. NDA=1: • La file d'attente contient au moins un bloc (appel de "X_RCV" avec EN_DT=0). Le bloc le plus ancien de la file d'attente a été copié dans le programme utilisateur (appel de "X_RCV" avec EN_DT=1).
RD	Output	ANY	I, Q, M, D	Référence à la zone de réception (receive data area). Les types de données suivants sont autorisés : BOOL, BYTE, CHAR, WORD, INT, DWORD, DINT, REAL, DATE, TOD, TIME, S5_TIME, DATE_AND_TIME, ainsi que les tableaux des types de données mentionnés à l'exception de BOOL. Si vous voulez rejeter le bloc le plus ancien figurant dans la file d'attente, donnez la valeur NIL au paramètre RD. La taille maximale de la zone de réception est de 76 octets.

Tableau 3. 4 les paramètres de l'instruction "X_RCV"

X_GET : Lire les données d'un partenaire de communication hors de la propre station S7 locale.

L'instruction "X_GET" permet de lire des données dans un partenaire de communication situé hors de la propre station S7. Il n'existe pas d'instruction correspondante dans le partenaire de communication.

La lecture est activée après appel de l'instruction avec REQ=1. Vous rappelez ensuite X_GET jusqu'à ce que BUSY=0 signale la réception des données. RET_VAL contient alors la longueur en octets du bloc reçu.

Veillez à ce que la zone de réception définie par le paramètre RD (dans la CPU réceptrice) soit au moins aussi longue que la zone de lecture définie par le paramètre VAR_ADDR (dans le partenaire de communication). Les types de données de RD et de VAR_ADDR doivent en outre concorder.

Paramètre	**Déclaration**	**Type de données**	**Zone de mémoire**	**Description**
REQ	Input	BOOL	I, Q, M, D, L	Paramètre de commande "request to activate"
CONT	Input	BOOL	I, Q, M, D, L	Paramètre de commande "continue"
DEST_ID	Input	WORD	I, Q, M, D, L ou constante	Paramètre d'adressage "destination ID". Il contient l'adresse MPI du partenaire de communication que vous avez configurée.
VAR_ADDR	Input	ANY	I, Q, M, D	Référence à la zone à lire dans la CPU partenaire. Vous devez choisir un type de données pris en charge par le partenaire de communication.
RET_VAL	Return	INT	I, Q, M, D, L	Si une erreur s'est produite pendant l'exécution de l'instruction, la valeur en retour contient le code d'erreur correspondant.

				En l'absence d'erreur, RET_VAL contient la longueur en octets du bloc copié dans la zone de réception RD, sous forme de nombre positif.
BUSY	Output	BOOL	I, Q, M, D, L	• BUSY=1 : La réception n'est pas encore achevée. BUSY=0 : La réception est achevée ou il n'y a pas de réception active.
RD	Output	ANY	I, Q, M, D	Référence à la zone de réception (receive data area). Les types de données suivants sont autorisés : BOOL, BYTE, CHAR, WORD, INT, DWORD, DINT, REAL, DATE, TOD, TIME, S5_TIME, DATE_AND_TIME, ainsi que les tableaux des types de données mentionnés à l'exception de BOOL. La zone de réception RD doit être au moins aussi longue que la zone de lecture VAR_ADDR dans le partenaire de communication. Les types de données de RD et de VAR_ADDR doivent en outre concorder. La taille maximale de la zone de réception est de 76 octets.

Tableau 3. 5 les paramètres de l'instruction "X_GET"

X_PUT : Ecrire les données sur un partenaire hors de la propre station S7.

L'instruction "X_PUT" permet d'écrire des données dans un partenaire de communication situé hors de la propre station S7. Il n'existe pas d'instruction correspondante dans le partenaire de communication.

L'écriture est activée après appel de l'instruction avec REQ=1. Vous rappelez ensuite X_PUT jusqu'à ce que BUSY=0 signale la réception de l'acquittement.

Veillez à ce que la zone d'émission définie par le paramètre SD (dans la CPU émettrice) ait la même longueur que la zone de réception définie par le paramètre VAR_ADDR (dans le partenaire de communication). Les types de données de SD et de VAR_ADDR doivent en outre concorder.

Paramètre	Déclaration	Type de données	Zone de mémoire	Description
REQ	Input	BOOL	I, Q, M, D, L	Paramètre de commande "request to activate"
CONT	Input	BOOL	I, Q, M, D, L	Paramètre de commande "continue"
DEST_ID	Input	WORD	I, Q, M, D, L ou constante	Paramètre d'adressage "destination ID". Il contient l'adresse MPI du partenaire de communication que vous avez configurée.
VAR_ADDR	Input	ANY	I, Q, M, D	Référence à la zone où écrire dans la CPU partenaire. Vous devez choisir un type de données pris en charge par le partenaire de communication.
SD	Input	ANY	I, Q, M, D	Référence à la zone de la propre CPU qui contient les données à envoyer. Les types de données suivants sont autorisés : BOOL, BYTE, CHAR, WORD, INT, DWORD, DINT, REAL, DATE, TOD, TIME,S5_TIME, DATE_AND_TIME, ainsi que les tableaux des types de données mentionnés à l'exception de BOOL. SD doit avoir la même longueur que le paramètre VAR_ADDR du partenaire de communication. Les types de données de SD et de VAR_ADDR doivent en outre concorder. La taille maximale de la zone d'émission est de 76 octets.
RET_VALUE	Return	INT	I, Q, M, D, L	Si une erreur s'est produite pendant l'exécution de l'instruction, la valeur en retour contient le code d'erreur correspondant.
BUSY	Output	BOOL	I, Q, M, D, L	• BUSY=1 : L'émission n'est pas encore achevée. BUSY=0 : L'émission est achevée ou il n'y a pas d'émission active.

Tableau 3. 6 les paramètres de l'instruction "X_PUT"

X_ABORT : Suspendre la liaison actuelle au partenaire de communication dans la propre station S7 locale

L'instruction "X_ABORT" vous permet d'interrompre la liaison à un partenaire de communication situé hors de la propre station S7, liaison qui avait été établie avec l'une des instructions "X_SEND", "X_GET" ou "X_PUT".

- Si la tâche associée à "X_SEND", "X_GET" ou "X_PUT" est achevée (BUSY = 0), les ressources de liaison utilisées pour cette tâche sont libérées des deux côtés après l'appel de l'instruction "X_ABORT".

- Si la tâche associée à "X_SEND", "X_GET" ou "X_PUT" n'est pas achevée (BUSY = 1), vous devez, une fois la liaison interrompue, appeler de nouveau l'instruction concernée avec REQ = 0 et CONT = 0 et attendre BUSY = 0. C'est seulement alors que toutes les ressources de liaison occupées seront libérées.

Vous ne pouvez appeler "X_ABORT" que du côté où "X_SEND", "X_PUT" ou "X_GET" s'exécute. L'interruption de la liaison est activée après appel de l'instruction avec REQ=1.

Paramètre	**Déclaration**	**Type de données**	**Zone de mémoire**	**Description**
REQ	Input	BOOL	I, Q, M, D, L	Paramètre de commande "request to activate"
DEST_ID	Input	WORD	I, Q, M, D, L ou constante	Paramètre d'adressage "destination ID". Il contient l'adresse MPI du partenaire de communication que vous avez configurée.
RET_VAL	Return	INT	I, Q, M, D, L	Si une erreur s'est produite pendant l'exécution de l'instruction, la valeur en retour contient le code d'erreur correspondant.
BUSYT	Output	BOOL	I, Q, M, D, L	• BUSY=1 : L'interruption de la liaison n'est pas encore achevée. BUSY = 0 : L'interruption de la liaison est achevée.

Tableau 3. 7 les paramètres de l'instruction "X_ABORT"

3.5 L’interface sur le logiciel WinCC

3.5.1 Règles IHM

Dans l’automatisation des processus, l’interface opérateur est généralement une unité d’interface homme-machine (HMI). L'IHM joue un rôle important dans la création d'un environnement visuel convivial entre l'utilisateur et la technologie.

Il existe divers aspects importants dans la conception d'écrans IHM pour répondre aux critères de qualité, ces aspects comprennent la représentation des couleurs, les problèmes de disposition de l'écran, de graphiques et d'images, valeurs de texte et de données, alarmes, navigation, contrôle, etc.

3.5.2 Les sous-écrans IHM et leurs utilisations

a. L'IHM qui contrôle le chariot

Cet écran (figure 3.5) permet de choisir manuellement la direction et la vitesse du moteur, il offre un affichage de la valeur actuelle du codeur et un écran d'alarme.

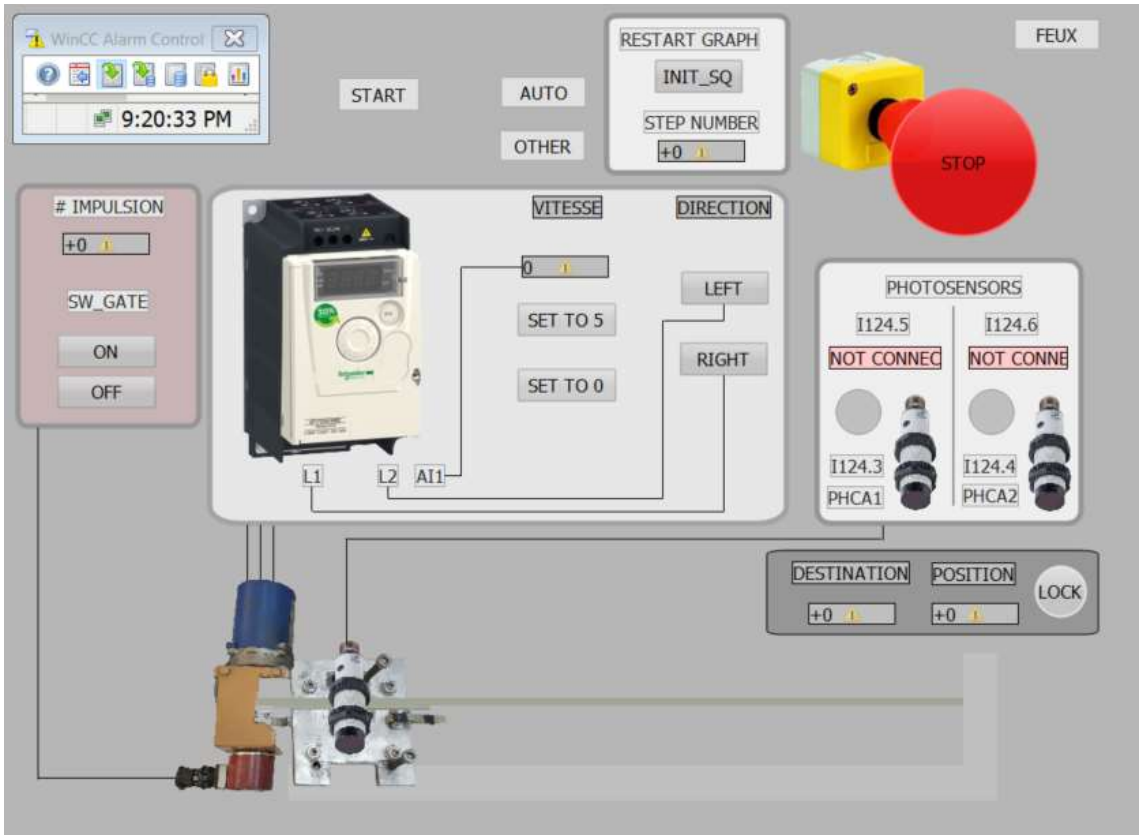


Figure 3. 11 Écran 1 (DIAG)

Cet écran (figure 3.6) permet de choisir une position pour le chariot. avec modes d'arrêt du chariot lors de la détection d'objet et mesure de la distance entre deux objets. Aussi, cet écran permet d'insérer plusieurs positions pour que le chariot se déplace l'une après l'autre.

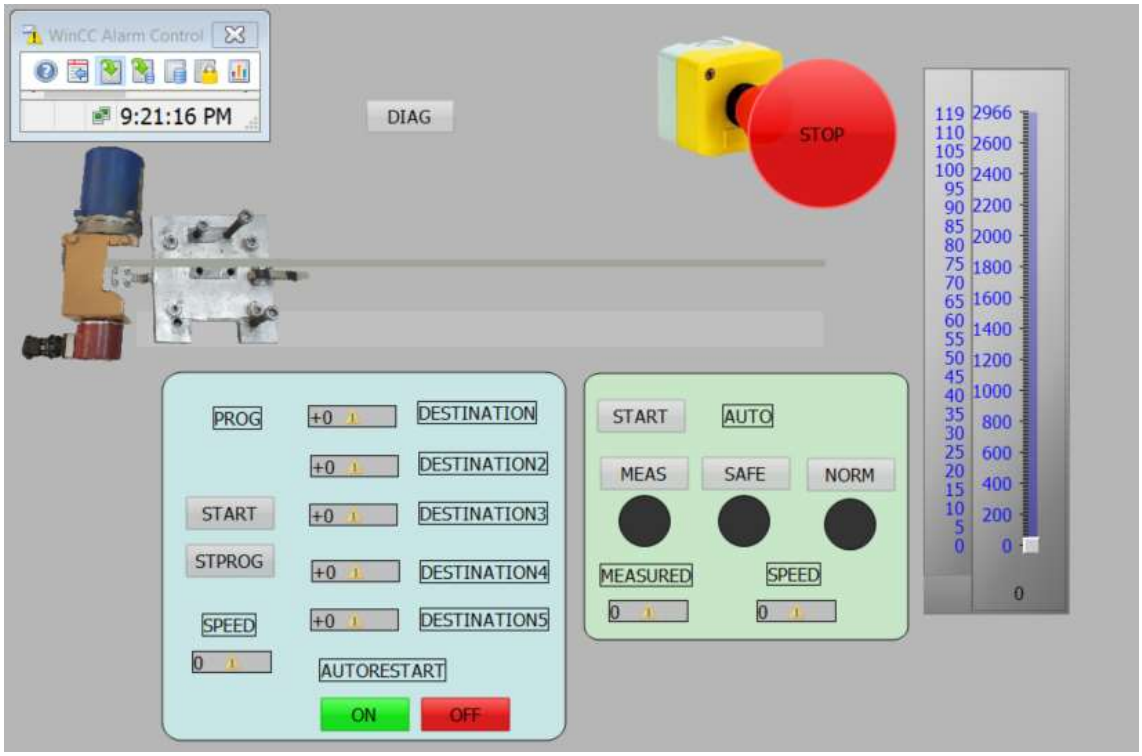


Figure 3. 12 Écran 2 (AUTO)

b. Le IHM qui contrôle les feux rouges

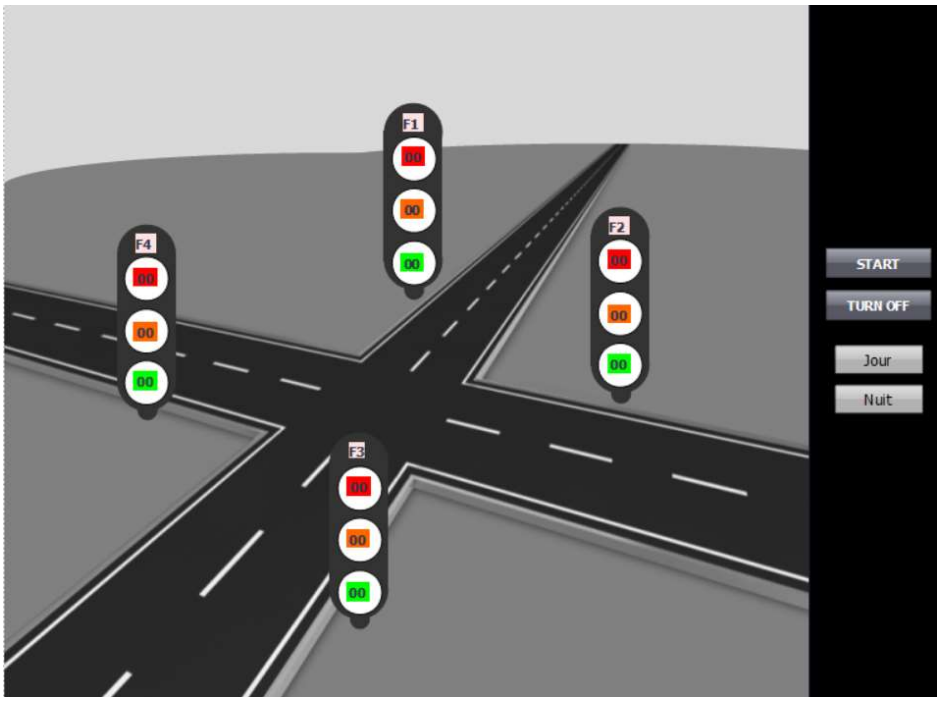


Figure 3. 13 L'écran IHM du logiciel WinCC

Le bouton START permet à l'IHM de fonctionner.

Le bouton TURN OFF désactive la fonctionnalité IHM.

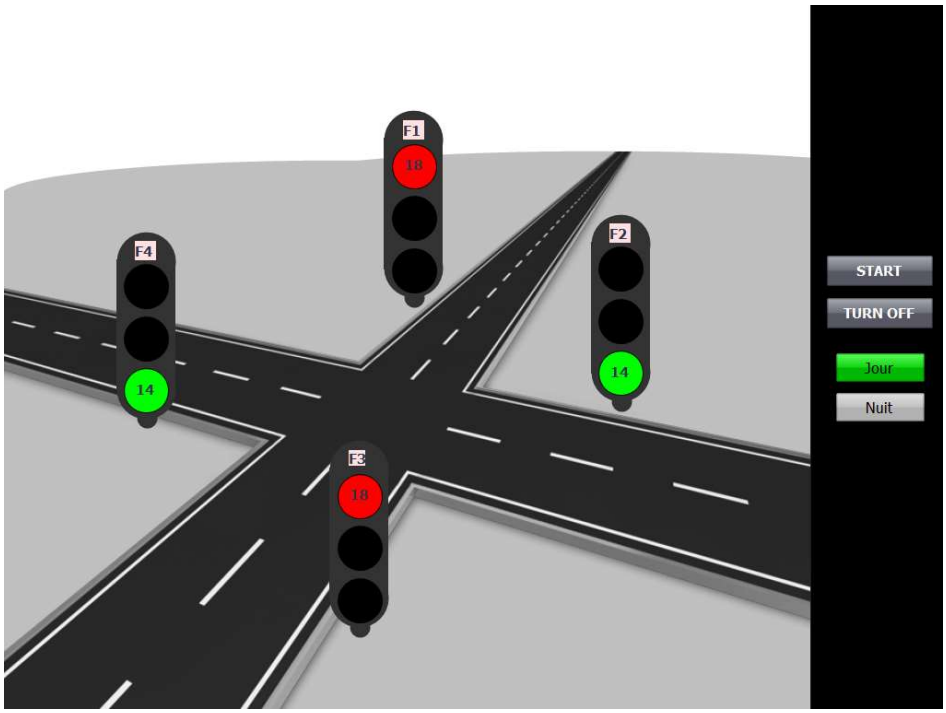


Figure 3. 14 L'écran IHM en mode jour

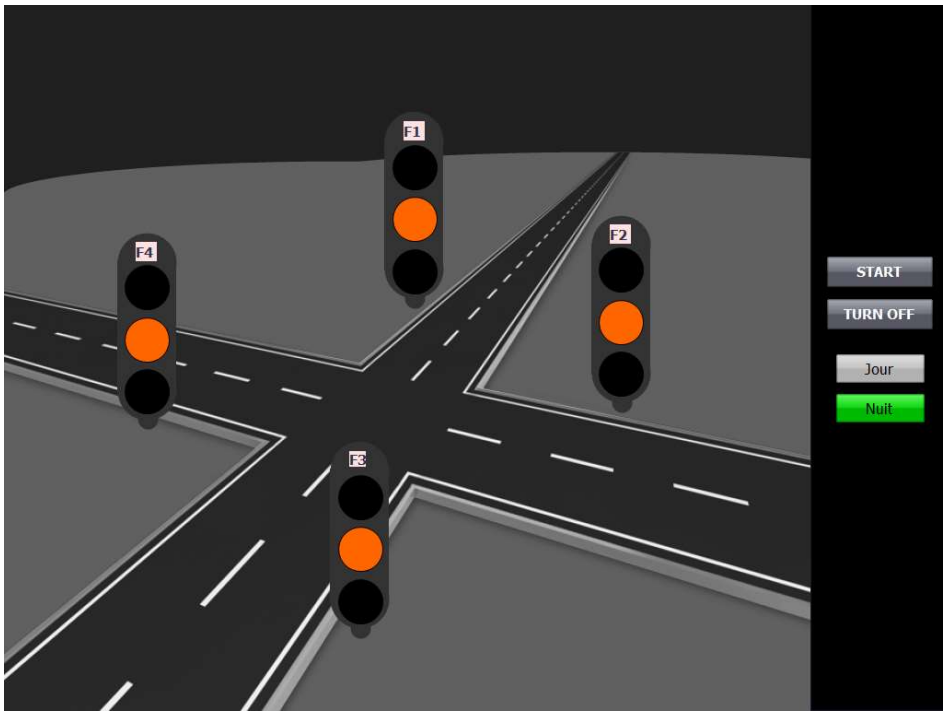


Figure 3. 15 L'écran IHM en mode nuit

3.6 Conclusion

La programmation dans le logiciel STEP7 est facile et flexible et offre plusieurs façons de programmer et d'optimiser votre travail, et l'intégration avec le kit WinCC dans le logiciel TIA offre un espace très confortable pour le travail.

3.7 Les résultats finaux

La communication que nous avons établie entre les automates est stable et cohérente. Et parce que les données que nous avons communiquées sont simples et limitées (et aussi parce que le réseau lui-même est petit), nous n'avons pas rencontré de problèmes liés à la vitesse relativement lente du support utilisé.

Les outils proposés pour programmer un transfert de données dans le logiciel STEP7 sont simples, mais il est difficile, voire impossible, de simuler le type de communication utilisé.

Notez que cela n'est pas un problème dans la nouvelle technologie de communication PROFINET.

Les fonctions que nous avons programmées dans chaque automate ont bien fonctionné. Une fois que nous avons utilisé le HSC dans le API du chariot, cela a fonctionné avec une grande précision et rapidité.

Conclusion générale

La technologie d'automatisation offre des solutions à une grande variété de difficultés industrielles et chaque fabricant a développé son propre écosystème pour son matériel et son logiciel. Pourtant, la communauté de l'ingénierie de l'automatisation a convenu d'un bon nombre de normes en matière de communication, d'électricité, etc.

La figure centrale de l'espace Siemens est le logiciel TIA qui permet de développer, de tester et enfin de superviser un projet industriel complet grâce aux nombreuses fonctionnalités qu'il réunit.

Nous avons rencontré dans la période de notre travail de nombreux obstacles, tels que l'installation de certains matériels, les différentes broches ou lignes de connexion (puissance ou signal) entre différentes générations d'un même matériel, des fonctionnalités incompatibles avec la simulation.et la courbe d'apprentissage du domaine de l'automatisation industrielle en général.

À la fin de notre parcours, nous sommes satisfaits de ce que nous avons appris, même si ce que nous avons montré lors de la réalisation est loin de ce que nous avons accumulé de l'expérience et des informations.

Nous espérons qu'à l'avenir, les étudiants pourront utiliser les informations détaillées simplifiées que nous avons présentées dans ce document.

Référence bibliographique

[1] : Duysinx P,2008. <<présentation des langages automate programmable >>. Revue technique. SIMENS API série S7.

[2] : Mémoire master, Etude d'un système de supervision et de contrôle,SCADA de la région de transport est RTE, Skikda année 2014.

[3] : bacem jrad,2011/2012 : <<support de coures : système automatisée >>.

[4] : Philippe LE BRUN, Automates programmables industriels.

[5] : Alain GONZAGA, les automates programmables industriels.

[6] : Alain GONZAGA,les automates programmables industriels.

[7] : site : http://www.elmoudjahid.com Développement de l'infrastructure urbaine : Alger, prochaine étape de la tournée des villes durable.la date de création 18-06-2012/0.00.

[8]: Formation step7 Totally Integrated Automation (T.I.A.), 2001.

[9] : M. KARIM, A. YASSIN << Etude d'adaptation d'un automate S7-300 sur une aléseuse GSP Ebauchée implémentation d'une interface homme/machine >>PFE faculté des sciences de l'ingénieure 2015.

[10] : Dr. Ir. H. Lecocq « Cours les réseaux locaux industriels » 2004.

[11] : Bertrand PETIT ; architectures des réseaux ,2006.

[12] : G. pugolle,les réseaux, EYROLLES ;1997.

[13] : Guide des automatisme, Copyright 2002-2007 Thierry Schanen & POS Industry.

[14] : le site : http://www4.ac-nancy-metz.fr/cpge-pmf-epinal/Cours_TD_SII/Elec/Machine%20asynchrone.pdf .coure de génie électrique. Machine asynchrone.

[15] : Document de formation pour une solution complète d'automatisation Totally Integrated Automation (T I A). Annexe IV Notions de base sur les bus de terrain avec SIMATIC S7-300.

[16] : Site : https://www.automation-sense.com/blog/automatisme/les-ob-de-demarrage-sur-step7-ob100-ob101-ob102.html 7/13/2019

[17] : L'aide du logiciel TIA PORTAL V13.

[18] : S7-300 CPU 31xC et CPU 31x : Caractéristiques techniques PDF de siemens.

[19]: SIMATIC S7-300 Module data pdf de siemens.

[20] : site **:** http://www.automotsys.com.au/encodersmc.html. 7/13/2019

[21] : Schneider-electric, guide d'exploitation, Altivar 12 Variateurs de vitesse pour moteurs asynchrone, 2010 PDF.

[22] : site **:** https://fr.wikipedia.org/wiki/Profibus. 7/13/2019

[23]: site**:** https://fr.wikipedia.org/wiki/Système_automatisé. 7/13/2019

[24]: site**:** http://www.iuma.ulpgc.es/~nunez/clases-sed-mai-68ppc/automatas-programables-plc/static-scribd-com-plc-operation.pdf.

[25]: site**:** https://realpars.com/plc-manufacturers. 7/13/2019

[26]: site**:** https://w3.siemens.com/mcms/automation-software/en/tia-portal-software/wincc-tia-portal/wincc-tia-portal-runtime/wincc-rt-professional/Pages/default.aspx. 7/13/2019

[27] : site **:** tvaira.free.fr/lpsari/CoursRLI-partie1.pdf.

[28] : site **:** https://cache.industry.siemens.com/dl/files/686/1254686/att_46478/v1/S7komm_e.pdf.

[29]: https://support.industry.siemens.com/cs/attachments/59193579/profibus_step7_v13_function_manual_en-US_en-US.pdf.

[30]: site**:** http://www.siemens.fi/pool/products/industry/iadt_is/tuotteet/automaatiotekniikka/teollinen_tiedonsiirto/PROFINET/man_pnsystem_description.pdf.

[31] : site**:** https://www.plcacademy.com/profibus-cable-connector-termination/ . 7/13/2019

[32] : https://automationforum.in/t/what-are-the-advantages-disadvantages-of-plc/2430 7/13/2019

[33] : https://fr.wikipedia.org/wiki/Chien_de_garde_(informatique) 7/13/2019