

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



Université Mohamed Khider Biskra
Faculté des Sciences et de la Technologie
Département de Génie Electrique
Filière : Automatique et informatique industrielle
Option : automatique avancés

Réf:.....

**Mémoire de Fin d'Etudes
En vue de l'obtention du diplôme:**

MASTER

Thème

**Commande d'un système pneumatique
didactique par automate programmable
SIMATIC S7-1200**

Présenté par :
Bensizerara Saliha
Soutenu le : ... Juin 2018

Devant le jury composé de :

Mr. ARIF ALI

MCA

President

Mr. SAADOUNE Achour

MCA

Encadreur

Mr. GUATTAF Abderrazak

MCA

Examineur

Année universitaire : 2017 / 2018

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



Université Mohamed Khider Biskra
Faculté des Sciences et de la Technologie
Département de Génie Electrique
Filière : Automatique et informatique industrielle

Mémoire de Fin d'Etudes
En vue de l'obtention du diplôme:

MASTER

Thème

**Commande d'un système pneumatique didactique par
automate programmable SIMATIC S7-1200**

Présenté par :

Bensizerara Saliha

Avis favorable de l'encadreur :

Mr. SAADOUNE Achour

Avis favorable du Président du Jury

Mr. ARIF ALI

Cachet et signature

Dédicace

*Je dédie ce mémoire à ma mère, mon père, qui m'a encouragé à
aller de l'avant*

Et qui m'ont donnée tous leurs soins pour continuer mes études.

A mes frères et mes sœurs

Et toutes mes amies.

*Et spécialement :hanene,wahida,amel
Amar ,Badreddine ,Habib ,Hamoudi,Abdelmalek*

A tous mes camarades de promotion 2017/2018

saliha

Remerciement

Je tiens premièrement à me prosterner devant Allah Le Tout Puissant, le remerciant de m'avoir donné le courage et la patience pour terminer ce travail.

Je remercie infiniment mon encadreur Dr. Saadoune Achour pour le suivi de mon travail, pour ses lectures, pour sa grande patience et pour l'aide qu'il m'a apportée durant la préparation de ce mémoire.

Mes remerciements vont également aux membres de jury pour m'avoir honoré par leur participation à l'évaluation de ce modeste travail.

Je dois remercier également Dr. Ghodhbane Hatem le chef du département informatique et aux enseignants de notre université.

Finalement, nous remercions ceux qui nous ont aidés de près ou de loin à réaliser ce projet.

Finalement, je remercie ma famille et tous mes amis qui m'a encouragé pour terminer ce travail.

Résumé

Dans ce projet on a essayé de faire la commande et la programmation d'un système pneumatique didactique (PTP ROBOT PNEUMATIQUE TRAINER) par un automate programmable siemens (SIMATIC S7-1200) à l'aide d'un logiciel de programmation TIA PORTAL V14 disponible au niveau de notre laboratoire d'automatique. Pour faire la simulation de ce système on a utilisé le simulateur PLCSIM.

Ce travail nous a permis de comprendre les différentes étapes suivies pour l'automatisation des systèmes, maîtriser le logiciel de programmation TIA PORTAL des automates de la gamme S7-1200 et par ailleurs assure une bonne formation pour les étudiants de filière d'automatique.

ملخص

في هذا المشروع حاولنا التحكم وبرمجة نظام هوائي تعليمي PTP ROBOT PNEUMATIQUE TRAINER بواسطة جهاز التحكم المنطقي القابل للبرمجة سيمنس (SIMATIC S7-1200) وباستخدام نظام البرمجة TIA PORTAL V14 المتوفر على مستوى مخبر الآلية. لإجراء المحاكاة لهذا النظام استخدمنا برنامج المحاكاة .PLCSIM

هذا المشروع سمح لنا بمعرفة مختلف المراحل المتبعة من أجل جعل الأنظمة الية، التمكن من الاستخدام الجيد لبرنامج TIA PORTAL الخاص بأجهزة التحكم المنطقي سيمنس S7-1200 ومن جهة أخرى ضمان تكوين جيد لطلبة شعبة الآلية.

Abstract

In this project we attempt to make the order and programming of educational systems PTP ROBOT TRAINER by PLC programmable Siemens SIMATIC S7-1200 using a programming software TIA PORTAL V14 available in laboratory of automatic.

to make the simulation of program we have used a PLCSIM programming software which contains a simulator to automate Siemens .this project allowed us to have good information on the different type of automate Siemens and also training ensure good training for student in the industry of automatic.

Sommaire

Introduction générale

Résumé

Liste des figures

Liste du tableau

Chapitre 1 : STRUCTURE GENERALE D'UNE INSTALLATION AUTOMATISEE

1.1. Introduction	2
1.2. Définition d'un système automatisé	2
1.2.1. Les objectifs	2
1.3. Structure générale d'une installation automatisée	3
1.3.1. Partie opérative	4
1.3.2. Partie commande	4
1.3.3. Constituants d'automatisme	5
1.3.3.1. Acquisition des données	5
1.3.3.2. Commande de puissance	5
1.3.3.3. Dialogue entre l'homme et la machine	5
1.3.3.4. Traitement des données	5
1.4. Automates programmables industriels	5
1.4.1. Définition	5
1.4.2. Structure d'un automate programmable	6
1.4.3. Fonctionnement d'un automate programmable	9
1.4.4. Alimentation de l'automate programmable industriel	9
1.5. Les langages de programmation	9
1.5.1. Logiciel de programmation STEP7-Micro/Win	10
1.5.2. Le langage booléenne (le langage liste d'instruction IL)	10
1.5.3. Le langage littéral structuré (ST)	10
1.5.4. Le langage Grafset	10
1.5.5. FDB (basé sur des blocs fonctions ou logigramme (LOG))	10
1.6. Quelques types des automates programmables industriels	11
1.7. conclusion	11

Chapitre 2 : DESCRIPTION D'AUTOMATE S7-1200

2. Automate SIEMENS S7-1200-V14	12
2.1. Présentation de l'automate SIEMENS S7-1200-V14	12
2.2. Possibilités d'extension de la CPU	13
2.3. Besoins du système	14
2.4. Logiciel de programmation TIA PORTAL	14
2.4.1. Présentation de l'adressage	15
2.4.2. Editeur CONT (schéma à contacts)	15
2.5. Présentation du logiciel	17
2.5.1. Introduction	17
2.5.2. Création d'un projet	17
2.5.3. Création d'une configuration matérielle	18
2.6. Présentation des blocs de programmation	21
2.6.1. Blocs d'organisation (OB)	22
2.6.2. Fonctions (FC)	22
2.6.3. Blocs fonctionnels (FB)	22
2.6.4. Blocs de données (DB)	23

Chapitre 3 : COMMANDE ET PROGRAMMATION D'UN SYSTEME DIDACTIQUE

3.1. Introduction	27
3.2. Les composants utilisés	28
3.3. Cahier des charges	29
3.4. Description des entrées/sorties	29
3.5. Description du cahier des charges par Grafcet	31
3. 6. Organigramme	34
3.7. Les adresses des entrées/sorties	35
3.8. Programmation en langage CONTACT (LADDER)	37
3.9. conclusion	47
Conclusion générale	48
Bibliographie	49

Liste des tableaux

Tableau 2-1 Besoins du système	14
Tableau 3.1 : Les éléments utilisés dans ce projet	27
Tableau 3.1 : Les éléments utilisés dans le PTP	28
Tableau 3.2 : Les adresses des entrées	34
Tableau 3.3 : Les adresses des sorties	35

Introduction générale

La compétitivité des entreprises impose un recours à la fois fréquent et intensif à des technologies de production avancée. La productique et la complexité des opérations à exécuter conduisent à la mise en œuvre de dispositifs et systèmes pour l'automatisation des ateliers de fabrication ou de production.

L'automate programmable industriel A.P.I est aujourd'hui le constituant le plus répandu pour réaliser des automatismes. On le trouve pratiquement dans tous les secteurs de l'industrie car il répond à des besoins d'adaptation et de flexibilité pour un grand nombre d'opérations. Cette émergence est due en grande partie à la puissance de son environnement de développement et aux larges possibilités d'interconnexions.

Dans ce contexte, on s'est intéressé à la commande et la programmation d'un système didactique (PTP ROBOT PNEUMATIQUE TRAINER) par un automate programmable siemens (SIMATIC S7-1200) à l'aide d'un logiciel de programmation TIA PORTAL V14.

Notre projet est présenté en trois chapitres :

- Le premier chapitre contient la description de la structure d'une installation automatisée.
- Le deuxième chapitre présente la description de l'automate Siemens S7-1200 et le logiciel de programmation TIA PORTAL.
- Dans le troisième chapitre on présente les étapes de commande et de programmation du système didactique (PTP ROBOT PNEUMATIQUE TRAINER).

Nous terminons ce travail par une conclusion générale.

Chapitre 1 :

STRUCTURE GENERALE D'UNE INSTALLATION AUTOMATISEE

1.1 Introduction

Un système technique est un ensemble d'éléments fonctionnels en interaction organisés en fonction d'une finalité ou d'un but. L'automatisation d'un processus industriel concerne tous les aspects de l'activité industrielle : production, assemblage, montage, contrôle, stockage, manutention, ...etc. L'objectif de l'automatisme est d'étudier le comportement du système (ce qu'il fait) en fonction de son évolution (consignes, informations), de l'environnement extérieur et du temps.

Avec l'automatisme, on pense souvent au monde de l'industrie, ceci est légitime, c'est là que l'on trouve principalement les API (Automate Programmable Industriel), mais de nos jours, l'automatisme est présent dans tous les systèmes simples : aussi bien le tapis roulant de la caisse de grande surface que les portes automatiques des magasins

1.2 Définition d'un système automatisé

Un automatisme est un appareillage assurant le fonctionnement automatique d'une machine ou d'une installation.

Un système physique créé par les hommes dans un but précis : c'est un système technologique qui est en relation avec leur environnement.

Une machine à un système dit automatisé, lorsque le processus qui permet de passer d'une situation initiale à une situation finale se fait sans intervention humaine et que ce comportement est répétitif chaque fois que les états qui caractérisent la situation initiale sont remplies. [1]

1.2.1 Les objectifs : Les objectifs de l'automatisation d'un système sont nombreux, on cite :

*Amélioration de la productivité de l'entreprise en réduisant les coûts de production (mains d'œuvre, matière, énergie) et en améliorant la qualité de produit.

*Amélioration des conditions de travail en supprimant les travaux pénibles et en améliorant la sécurité.

*Augmentation de la production.

*Augmentation de la disponibilité des moyens de production en améliorant les possibilités de gérer le système.

Un automatisme peut être divisé en deux parties :

1.3.1 Partie opérative

Appelée parfois partie puissance, la partie opérative d'un automatisme assure la transformation de la matière d'œuvre.

- **La partie mécanique** : chariots, glissières, engrenages, poulies, broches...
- **Les actionneurs** convertissent l'énergie d'entrée disponible sous une certaine forme (électrique, Pneumatique, hydraulique) en une énergie utilisable sous une autre forme, par exemple :
 - Energie thermique destinée à chauffer un four (l'actionneur étant alors une résistance électrique).
 - Energie mécanique destinée à provoquer une translation de chariot (l'actionneur pouvant être un vérin hydraulique ou pneumatique).
 - Energie mécanique destinée à provoquer une rotation de broche (l'actionneur pouvant être alors un moteur électrique).
- **Les préactionneurs** reçoivent les signaux de commande et réalisent la commutation de puissance avec les actionneurs. Les préactionneurs des moteurs électriques sont appelés contacteurs. **Les préactionneurs** des vérins et des moteurs hydrauliques et pneumatiques sont appelés distributeurs (à commande électrique ou pneumatique).
- **Les capteurs**, qui communiquent à la partie commande des informations sur la position d'un mobile, une vitesse, la présence d'une pièce, une pression...
 - **Les capteurs T.O.R.** (tout ou rien), qui délivrent un signal de sortie logique, c'est à dire 0 ou 1. Exemple : détecteur de fin de course.
 - **Les capteurs numériques**, ou « incrémentaux », qui associés à un compteur, délivrent des signaux de sortie numérique. Exemple : capteur ou codeur incrémental utilisé pour la mesure des déplacements des chariots de machine à commande numérique.
 - **Les capteurs analogiques**, ou proportionnels » qui permettent de prendre en compte la valeur réelle d'une grandeur physique. Exemple : Sonde de température.
- **Les appareils de ligne.** Ceux-ci représentent l'ensemble des composants indispensables à la mise en œuvre et à la bonne marche de l'automatisme

1.3.2 Partie commande

Appelée également « partie traitement des informations », elle regroupe tous les composants de traitement des informations nécessaires à la bonne marche de la **partie opérative**.

- La partie commande communique avec l'opérateur par l'intermédiaire d'un **pupitre**.

- Les informations entre la partie commande et la partie opérative passent souvent par l'intermédiaire **d'interfaces**.

Trois technologies sont actuellement utilisées

- _ Électromécanique,
- _ Pneumatique,
- _ Électronique.

1.3.3 Constituants d'automatisme

Ces quatre fonctions essentielles sont animées par des constituants d'automatisme dans les technologies : électrique, électronique, électropneumatique, pneumatique.

1.3.3.1 Acquisition des données

Réalisée par des détecteurs électroniques, des cellules photo-électriques, des manostats, ...etc. Ces appareils collectent, décodent et transforment en un signal électrique des grandeurs physiques de nature différentes (longueur, vitesse, masse, pression, nombre...).

1.3.3.2 Commande de puissance

Assurée par des contacteurs électromagnétiques, des variateurs de vitesse électronique qui contrôlent le courant à destination des moteurs électriques, ou en technologie pneumatique par des distributeurs.

1.3.3.3 Dialogue entre l'homme et la machine

Transmis grâce à des claviers, des boutons poussoirs, des écrans de contrôle des voyants lumineux. Par ces moyens, l'homme reçoit des informations renvoie ses ordres et surveille le processus.

1.3.3.4 Traitement des données

Fonction remplie par un automate programmable dans un automatisme industrielle selon le programme mis en mémoire. Il réagit aux signaux électriques qu'il reçoit et envoie les ordres d'actions aux organes de commande de puissance. [2]

1.4 Automates programmables industriels

1.4.1 Définition

L'automate programmable est un produit de technologie électronique dont le fonctionnement est défini par un programme ; c'est un organe de traitement de l'information comme toutes les familles de constituants pneumatiques à relais ou électronique. [3]

1.4.2 Structure d'un automate programmable

Un automate programmable industriel (A.P.I.) est un système muni d'un microprocesseur. Le processeur dialogue avec les interfaces, c'est à dire les entrées/sorties (E/S), avec la mémoire, avec l'outil de programmation et éventuellement avec le réseau de communication.

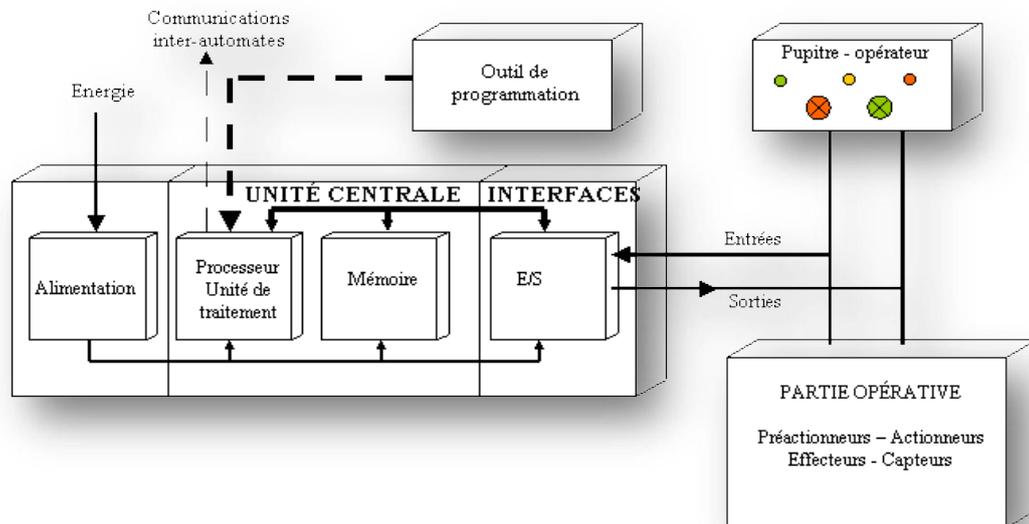


Figure 1.2 : Schéma d'une structure d'automate programmable. [4]

Mémoire contient, sous forme binaire, les informations indispensables au bon Le processeur, la mémoire et les interfaces sont interconnectés grâce à des bus. Le fonctionnement de l'API (Programme et données). Sur une mémoire, on peut lire, écrire et effacer.

L'automate programmable reçoit les informations relatives à l'état du système et puis commande les pré-actionneurs suivant le programme inscrit dans sa mémoire.

Un API se compose donc de trois grandes parties :

- Le microprocesseur.
- La zone mémoire.
- Les interfaces Entrées/Sorties. [4]

➤ Le microprocesseur

Le microprocesseur réalise toutes les fonctions logiques ET, OU, les fonctions de temporisation, de comptage, de calcul... à partir d'un programme contenu dans sa mémoire. Il est connecté aux autres éléments (mémoire et interface E/S) par des liaisons parallèles appelées 'BUS' qui véhiculent les informations sous forme binaire. [4]

➤ **La zone mémoires**

a. La Zone mémoire va permettre

- De recevoir les informations issues des capteurs d'entrées.
- De recevoir les informations générées par le processeur et destinées à la commande des sorties (valeur des compteurs, des temporisations, ...).
- De recevoir et conserver le programme du processus. [4]

b. Action possible sur une mémoire

- ECRIRE pour modifier le contenu d'un programme.
- EFFACER pour faire disparaître les informations qui ne sont plus nécessaires.
- LIRE pour en lire le contenu d'un programme sans le modifier. [4]

c. Technologie des mémoires

- RAM (Random Access Memory) : mémoire vive dans laquelle on peut lire, écrire et effacer (contient le programme).
- ROM (Read Only Memory) : mémoire morte accessible uniquement en lecture.
- EPROM mémoires mortes reprogrammables effacement aux rayons ultra-violets.
- EEPROM mémoires mortes reprogrammables effacement électrique. [4]

➤ **Les interfaces d'entrées/sorties**

Les entrées reçoivent des informations en provenance des éléments de détection (capteurs) et du pupitre opérateur (BP). Les sorties transmettent des informations aux pré-actionneurs (relais, électrovannes ...) et aux éléments de signalisation (voyants) du pupitre. [4]

a. Interfaces d'entrées

Elles sont destinées à :

- Recevoir l'information en provenance des capteurs.
- Traiter le signal en le mettant en forme, en éliminant les parasites et en isolant électriquement l'unité de commande de la partie opérative. [4]

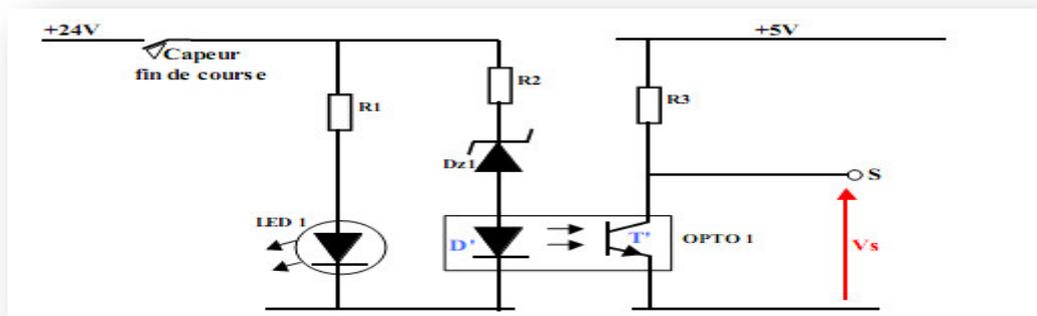


Figure 1.3 : Schéma d'interface d'entrée. [5]

❖ Fonctionnement de l'interface d'entrée

Lors de la fermeture du capteur :

- LED1 signale que l'entrée automate est actionnée.
- LED D' d'optocoupleur s'éclaire.
- La photo transistor T' de l'optocoupleur devient passant.
- La tension $V_s=0V$.

Donc lors de l'activation d'une entrée automate, l'interface d'entrée envoie une 0 logique à l'unité de traitement et une 1 logique lors de l'ouverture du contact du capteur (entrée non actionnée). [6]

b. Interfaces de sorties

Elles sont destinées à :

- Commander les pré-actionneurs et éléments des signalisations du système.
- Adapter les niveaux de tensions de l'unité de commande à celle de la partie opérative du système en garantissant une isolation galvanique entre ces dernières

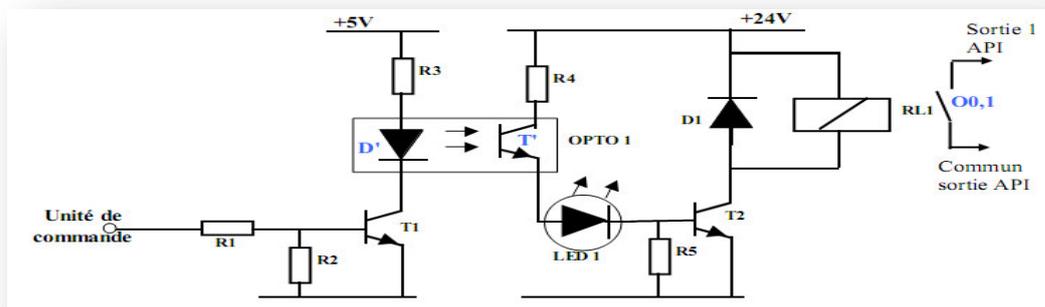


Figure 1.4 : Schéma d'interface de sortie. [5]

❖ Fonctionnement de l'interface de sortie

Lors de la commande d'une sortie automate :

- L'unité de commande envoie une 1 logique (5V).
- T1 devient passant, donc D' s'éclaire.
- La photo transistor T' de l'optocoupleur devient passant.
- LED 1 s'éclaire et nous informe de la commande de la sortie O0, 1.
- T2 devient passant.
- La bobine RL1 devient sous tension et commande la fermeture du contact de la sortie O0, 1.

Donc pour commander une sortie automate l'unité de commande doit envoyer :

- Une 1 logique pour actionner une sortie API.
- Une 0 logique pour stopper la commande d'une sortie API. [5]

1.4.3 Fonctionnement d'un automate programmable

Lorsque le programme est introduit dans la mémoire de l'automate en utilisant soit une console de programmation, soit un PC, la phase d'exécution est alors possible. Elle est généralement obtenue après la mise en RUN de l'automate (cette commande peut être logicielle ou matérielle). [6]

1.4.4 Alimentation de l'automate programmable industriel

L'alimentation intégrée dans l'API, fournit à partir des tensions usuelles des réseaux (230V, 24V) les tensions continues nécessaires au fonctionnement des circuits électroniques.[5]

1.5. Les langages de programmation

Le langage de programmation permet l'établissement du dialogue entre l'utilisateur et l'automate programmable. La norme internationale CEI 1131-3 propose 5 langages de programmation que l'on retrouvera chez les principaux constructeurs d'automate programmable : [6]

1.5.1. Le langage graphique à contacts (LD : LADDER) ou diagramme

C'est une transcription des schémas à relais, le principe en est de donner au programme une apparence de schéma électrique développé. La saisie du programme se fait en dessinant à l'écran du terminal de programmation, les réseaux LADDER, à partir des symboles graphiques proposés : contacts, bobines et blocs. [6]

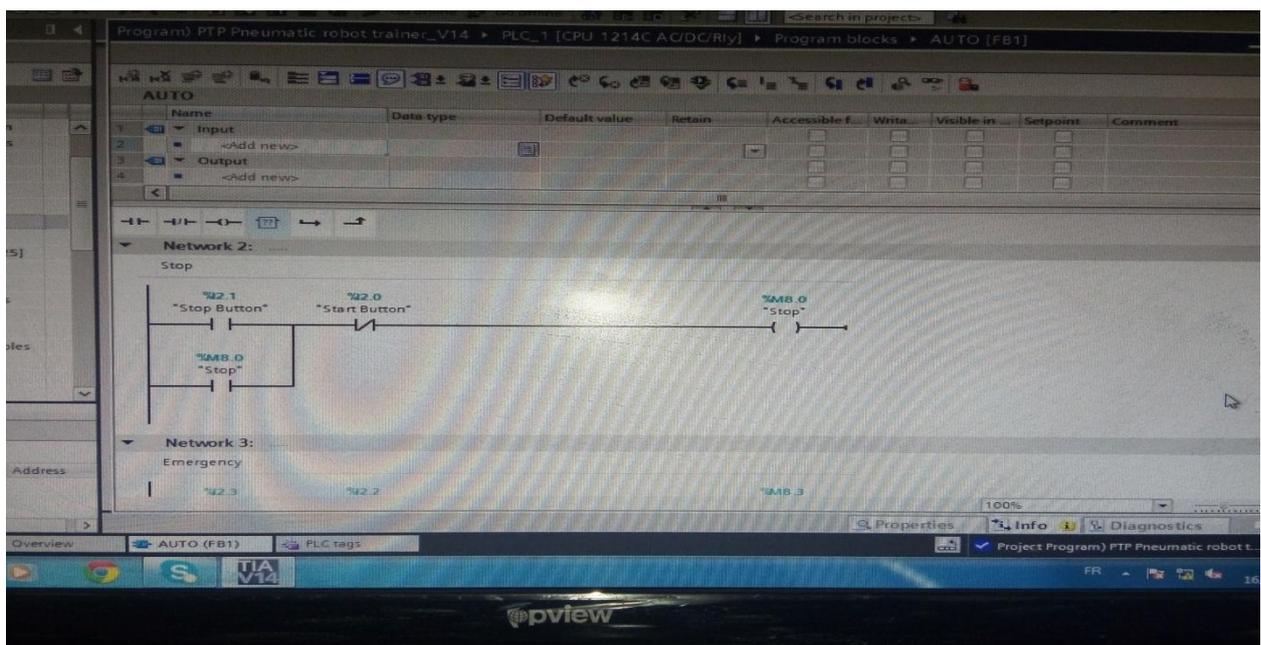


Figure 1.6 : Exemple de représentation langage CONT. [6]

1.5.2 Le langage booléenne(le langage liste d'instruction IL)

C'est un langage «machine» qui permet l'écriture de traitement logique et numérique : il permet de décrire des équations booléennes utilisant des opérateurs simples (ET, OU, NON ...) ou des fonctions déjà préexistantes (temporisateurs, compteurs, séquenceurs ...). [6]

1.5.3 Le langage littéral structuré(ST)

Il est directement issu des langages informatiques. Il permet essentiellement la description de calculs logiques et arithmétiques, en utilisant des phrases structurées (du type IF, THEN, ELSE par exemple). [6]

1.5.4 Le langage Grafcet

Ce langage permet de représenter graphiquement et de façon structurée le fonctionnement d'un automatisme séquentiel. La saisie du programme se fait en «dessinant» les grafkets à partir des symboles proposés. Les réceptivités et les actions sont décrites sous forme de réseau LADDER. Lors de l'exécution du programme, le processeur ne lira que les parties du logiciel concernant les étapes actives. [6]

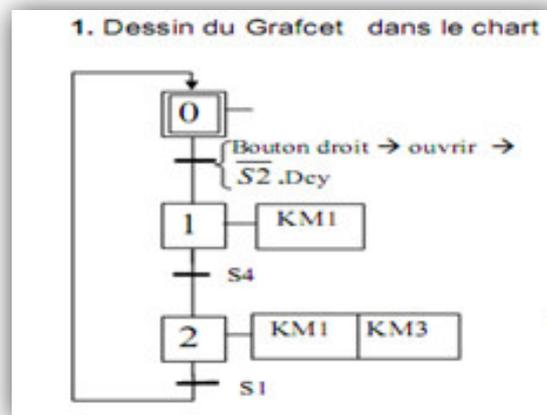


Figure 1.7 : Exemple de représentation langage Grafcet [6]

1.5.5 FDB (basé sur des blocs fonctions ou logigramme (LOG))

Il permet de visualiser les opérations sous forme de boîtes logiques ressemblant à des schémas de porte logique courants. IL n' Ya pas de contacts ni de bobines comme dans l'éditeur CONT, mais des opérations équivalentes existent sous forme de boîtes d'opérations. La logique du programme se dérive des connexions entre ces boîtes d'opérations : on peut utiliser la sortie d'une opération (par exemple une boîte ET) pour valider une autre opération (par **exemple** une temporisation) afin de créer la logique de commande nécessaire. [7]

1.6 Quelques types des automates programmables industriels

- ◆ Schneider série TSX 17, 37 (Micro), 47 avec les afficheurs XBT Mage lis ou Pro face.
 - Logiciel : Pl7 Micro ou Pl7 Pro.
- ◆ Siemens série S7-1200.
 - Logiciel : Step7.
- ◆ Mitsubishi FXON, Série A1.
 - Médoc.

Dans ce projet on utilise un automate programmable Siemens série S7-1200(SIEMENS S7-1200).

Chapitre 2 :

2.1 Automate SIEMENS S7-1200-V14

2.1.1 Présentation de l'automate SIEMENS S7-1200-V14

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

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

La CPU fournit un port PROFINET permettant de communiquer par le biais d'un réseau PROFINET. Des modules supplémentaires sont disponibles pour communiquer via les Réseaux PROFIBUS, GPRS, RS485 ou RS232 [10].

Pour ce projet, le S7-1200 sera programmé en langage LADDER à l'aide du logiciel TIA PORTAL sous Windows.

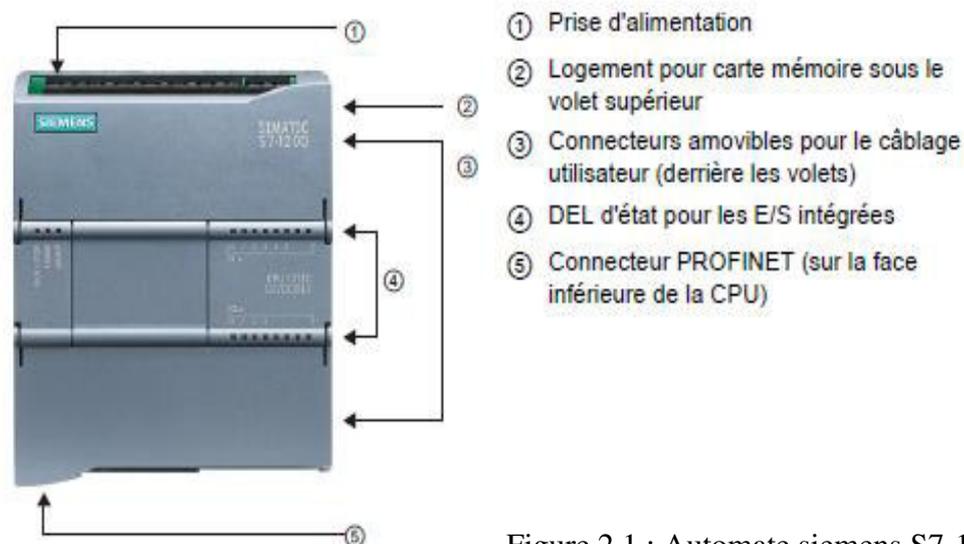


Figure 2.1 : Automate siemens S7-1200[10].

2.2. Possibilités d'extension de la CPU :

La gamme S7-1200 offre divers modules et cartes enfichables pour accroître les capacités de la CPU avec des E/S supplémentaires ou d'autres protocoles de communication.

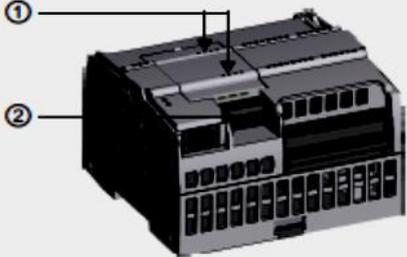
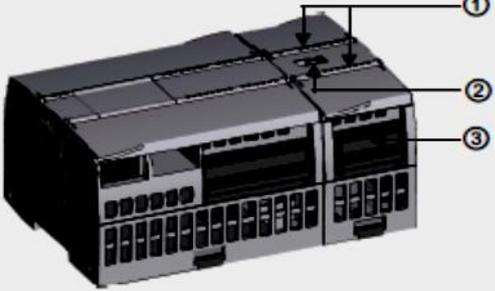
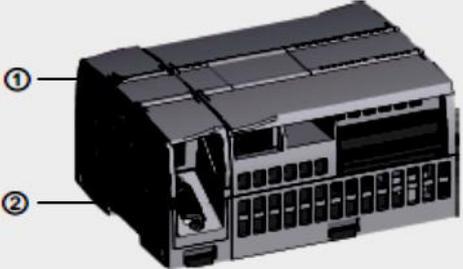
Type de module	Description		
<p>La CPU prend en charge une carte d'extension enfichable :</p> <ul style="list-style-type: none"> • Un Signal Board (SB) fournit des E/S supplémentaires pour votre CPU. Le SB se raccorde à l'avant de la CPU. • Un Communication Board (CB) vous permet d'ajouter un autre port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

Figure 2.2 : Possibilités d'extension de la CPU [10].

2.3. Besoins du système

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

Tableau 2-1 Besoins du système

Matériel/logiciel	Condition de requise
Type de processeur	Pentium M, 1,6 GHz ou similaire
Mémoire vive	1 Go
Espace disponible sur le disque dur	2 Go sur le lecteur système C:\
Systèmes d'exploitation	* Windows XP Professional SP3 * Windows 2003 Server R2 StdE SP2 * Windows 7 Home Premium (STEP 7 Basic uniquement, non pris en charge pour STEP 7 Professional) * Windows 7 (Professional, Enterprise, Ultimate) * Windows 2008 Server StdE R2
Carte graphique	32 Mo RAM
Profondeur de couleur	24 bits
Résolution de l'écran	1024 x 768
Réseau	Ethernet 20Mbits/s ou plus Lecteur optique
Lecteur optique	DVD-ROM

2.4 Logiciel de programmation TIA PORTAL

Dans TIA PORTAL, le projet vous permet d'associer votre programme à toutes les informations nécessaires pour que vous puissiez communiquer avec un API et charger le programme dans cet API. [7]

2.4.1 Présentation de l'adressage

Vous pouvez identifier les paramètres des éléments de commande dans votre programme de manière absolue ou symbolique. Une référence absolue(ou directe) utilise la zone de mémoire et l'adresse de bit ou d'octet pour identifier l'opérande. Une référence symbolique (ou indirect) utilise une combinaison de caractères alphanumérique pour identifier l'opérande. [8]

Les bits d'entrées de l'automate sont notés :

Ii.n	avec n=0, 1,2,..., 7 le numéro du bit, I= entrée	Exemple : I0.0 : entrée 0 bit 0 (Interrupteur)
-------------	---	---

Les bits de sortie de l'automate sont notés :

Qi.n	avec n=0, 1,2,..., 7 le numéro du bit, Q= sortie	Exemple : Q0 .0 sortie0 bit 0 (Lampe)
-------------	---	--

Les bits internes de l'automate sont notés :

Mi.n	avec n=0, 1,2,..., 7 le numéro du bit M=Mémento=mémoire	Exemple : M10 : mémoire 10 (CF0)
-------------	--	---

Les CPU SIMATIC S7-1200 mettent à votre disposition plusieurs langages vous permettant de résoudre une large gamme de tâches d'automatisation. Les deux jeux d'opérations de base disponibles dans une CPU S7-1200 sont le jeu d'opérations SIMATIC et le jeu d'opérations CEI 1131-3. Concernant ce dernier, les jeux d'instruction normalisés CEI proposés sont :

➤ Le langage LD (CONT)

Vous devez, lors de chaque création de programme, choisir :

- Le jeu d'opérations convenant le mieux à votre application (SIMATIC ou CEI 1131-3).
- Le type d'éditeur convenant le mieux à vos besoins de programmation (liste d'instructions, schéma à contacts ou logigramme). [8]

2.4.2 Editeur CONT (schéma à contacts)

L'éditeur schéma à contacts (CONT) de SETP7-Micro/WIN 32 permet de créer des programmes qui ressemblent à un schéma de câblage électrique. La programmation en CONT est la méthode choisie par de nombreux programmeurs d'automates programmables et par le personnel de maintenance ; c'est un langage qui convient également très bien aux programmeurs débutants, fondamentalement les programmes CONT permettent à la CPU d'émuler le trajet de courant

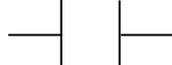
électrique partant d'une source de tension, à travers une série de conditions d'entrée logiques validant, à leur tour, des conditions de sortie logiques. Généralement on subdivise le code en sections de petite taille et faciles à comprendre, souvent appelées "réseaux". L'exécution du programme se fait réseau par réseau, de gauche à droite et de haut en bas, comme prescrit par le programme. Lorsque la CPU a atteint la fin du programme, elle recommence au début du programme.

Les différentes opérations sont représentées par des symboles graphiques de trois types fondamentaux : [8]

▪ **Contacts**

▪ Les contacts représentent des commutateurs à travers lesquels le courant peut circuler. Le courant circule à travers un contact à fermeture uniquement si ce contact est fermé (1 logique) ; le courant circule à travers un contact à ouverture ou inversé (NOT) uniquement si ce contact est ouvert (0 logique).

Symbole du contact :

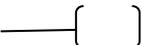
Le contact normalement ouvert (NO) (en NO : normally open) : 

Le contact normalement fermé (NF) (en NC : normally closed) : 

- * Le contact à fermeture est fermé (activé) lorsque la valeur du bit affecté est égale à 1.
- * Le contact à ouverture est fermé (activé) lorsque la valeur du bit affecté est égale à 0.
- * Des contacts connectés en série créent des réseaux de logique ET.
- * Des contacts connectés en parallèle créent des réseaux de logique OU.

***Bobine**

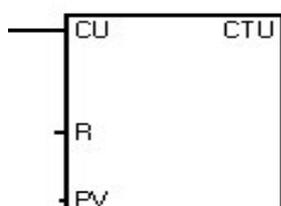
Les bobines représentent généralement des résultats de sortie logiques, tels que lampes, démarreurs de moteur, relais intermédiaires, conditions de sortie internes, etc.

Symbole de la bobine : [8] 

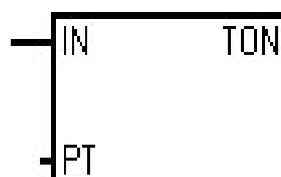
▪ **Boîte**

Les boîtes représentent généralement une fonction (par exemple une temporisation, un compteur ou une opération arithmétique) qui est exécutée lorsque le courant atteint cette boîte. [8]

Symbole de la boîte :



Compteur



Temporisateur

2.5 Présentation du logiciel

2.5.1 Introduction

Le logiciel « Totally Integrated Automation Portal » (TIA) est un logiciel de programmation des automates de la gamme S7-1200 et des pupitres opérateur de la gamme KTP. Celui-ci est composé de STEP7 Basic et Win CC Basic. Il reprend la même philosophie de programmation que le logiciel STEP7 Pro avec une interface simplifiée et l'intégration de la programmation de pupitre opérateur.

Le souhait de SIEMENS est d'intégrer toutes leurs gammes de produits pour un seul logiciel.

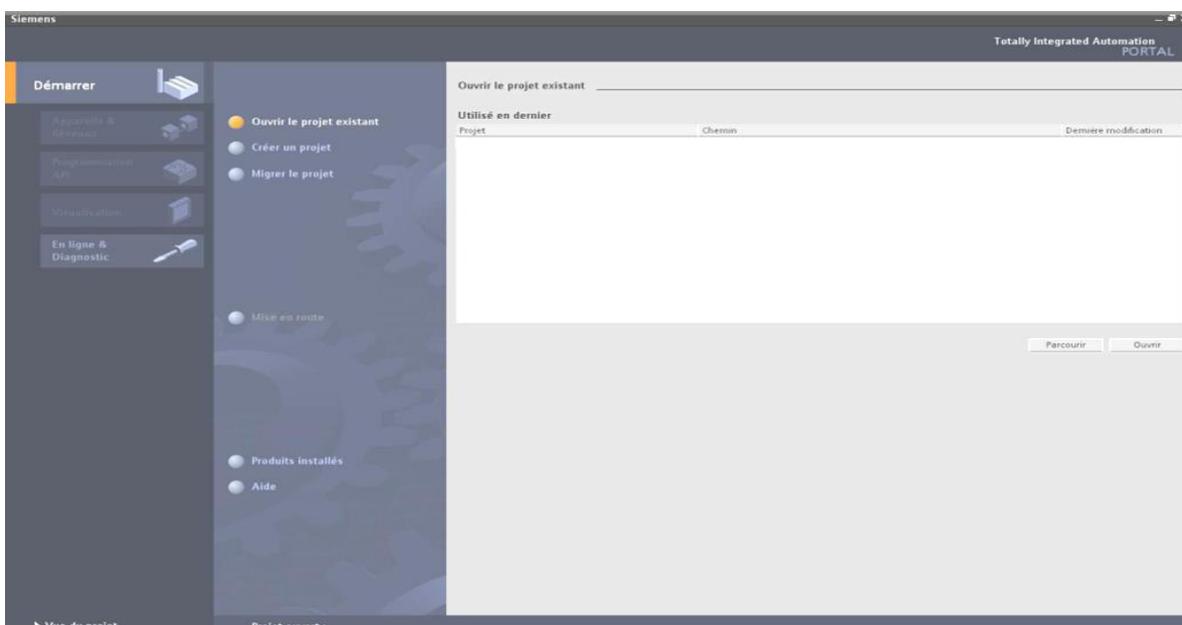
Lors de l'installation vous vous apercevrez que les licences sont intégrées au CD d'installation et se placent sur votre disque dur automatique, par contre celles-ci sont verrouillées, vous ne pouvez donc ni les supprimer, ni les déplacer [10].

2.5.2 Création d'un projet

Une fois le logiciel entièrement installé, cliquez sur l'icône pour le démarrer



Voici la page de démarrage que vous verrez :

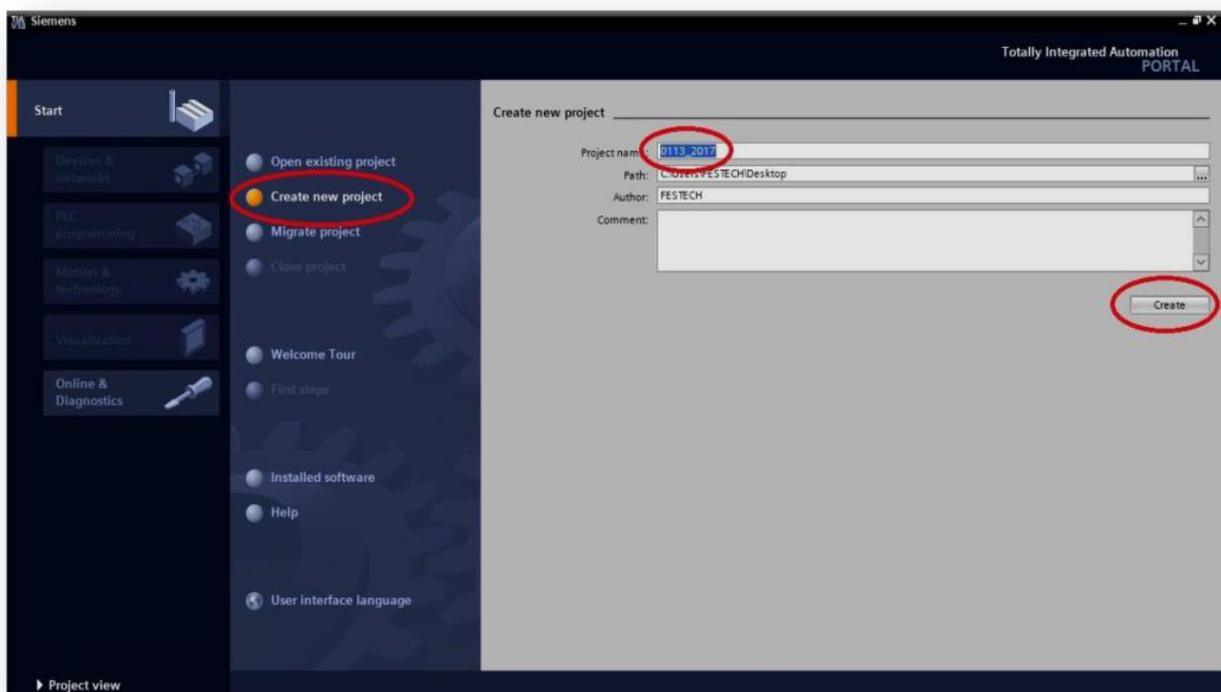


Vous pouvez sur cette page :

- Visualiser les projets déjà créés et les ouvrir.
- Créer un nouveau projet.
- Migrer un projet existant.
- Se mettre en ligne avec un automate S7-1200 ou un pupitre KTP et avoir accès au diagnostic.

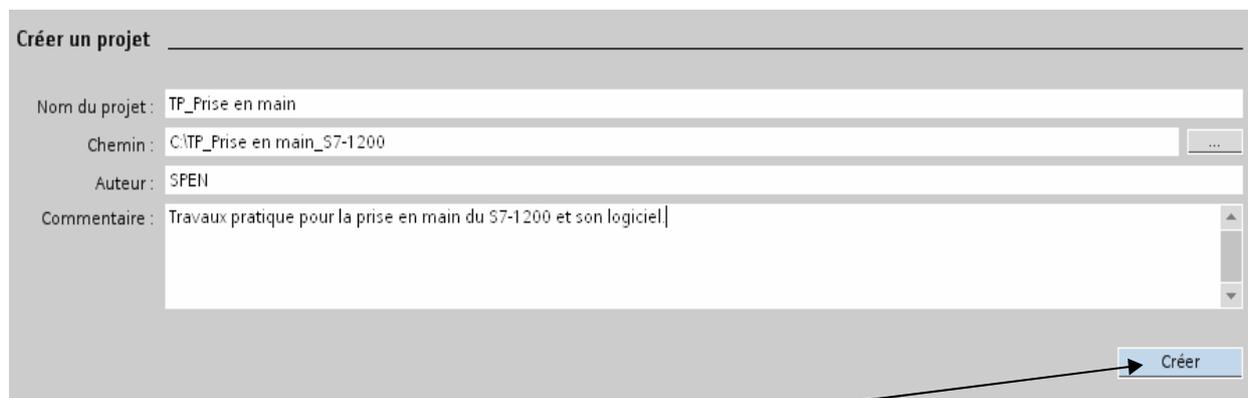
En bas de cette page vous voyez le projet ouvert et vous avez l'accès à ce projet en cliquant sur « Vue du projet ».

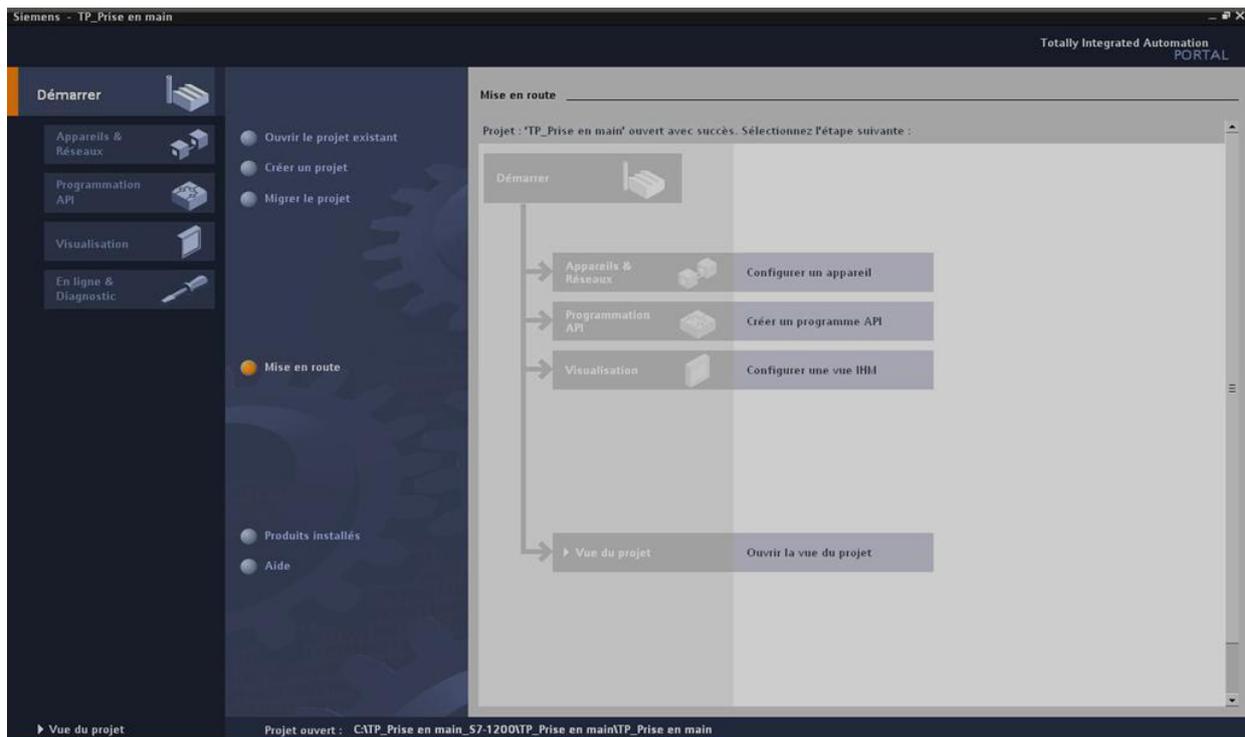
Nous allons dans un premier temps créer un nouveau projet, cliquez pour cela sur l'onglet « Créer un projet » :



Le projet étant créé, nous allons maintenant pouvoir saisir la configuration matérielle

2.5.3 Création d'une configuration matérielle

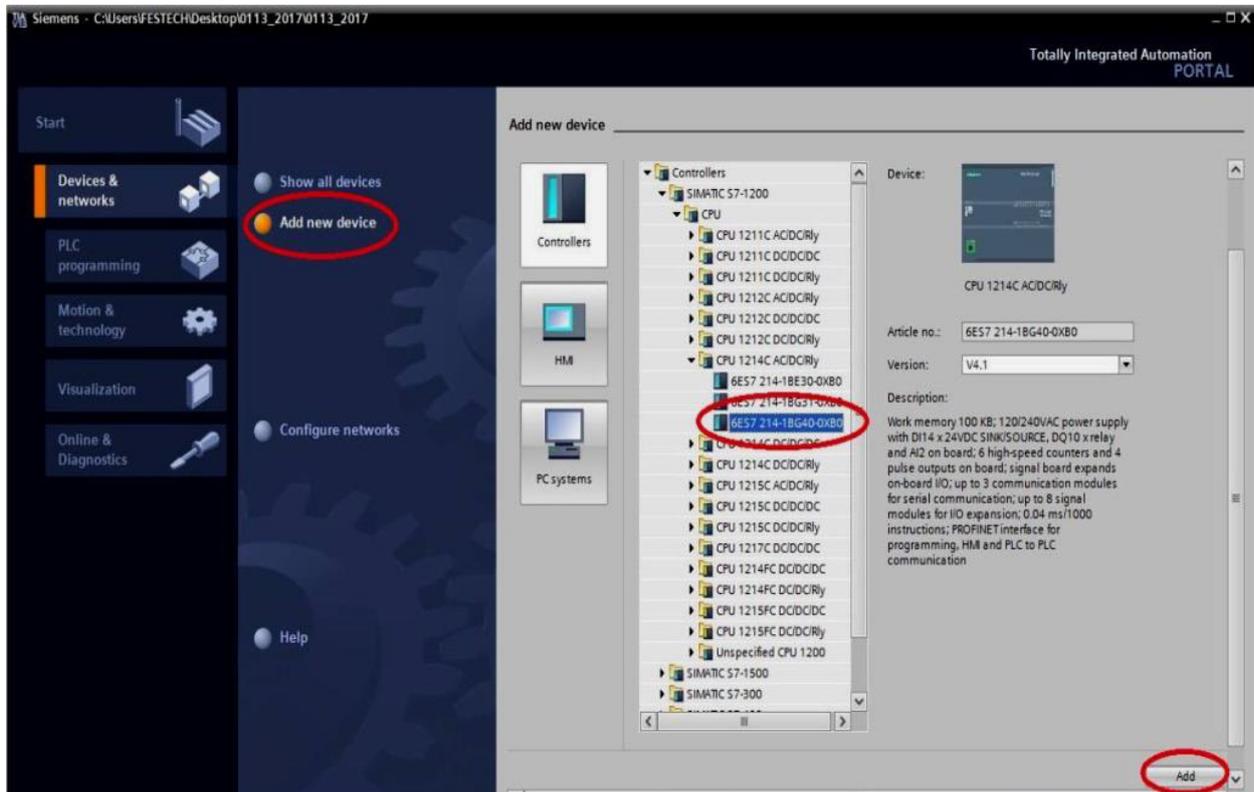




Cliquez sur « Configurer un appareil », puis sur « Ajouter un appareil ».

Nous allons tout d'abord ajouter la CPU, si vous ne connaissez pas la référence vous pouvez cliquer sur « CPU 1214 non spécifiée », vous pourrez alors la saisir plus tard ou la configuration matérielle se mettra à jour lors de la connexion avec la dite CPU.

Nous allons ajouter une CPU 1214C AC/DC/Rly > 6ES7 214-1BG40-0XB0 [11].

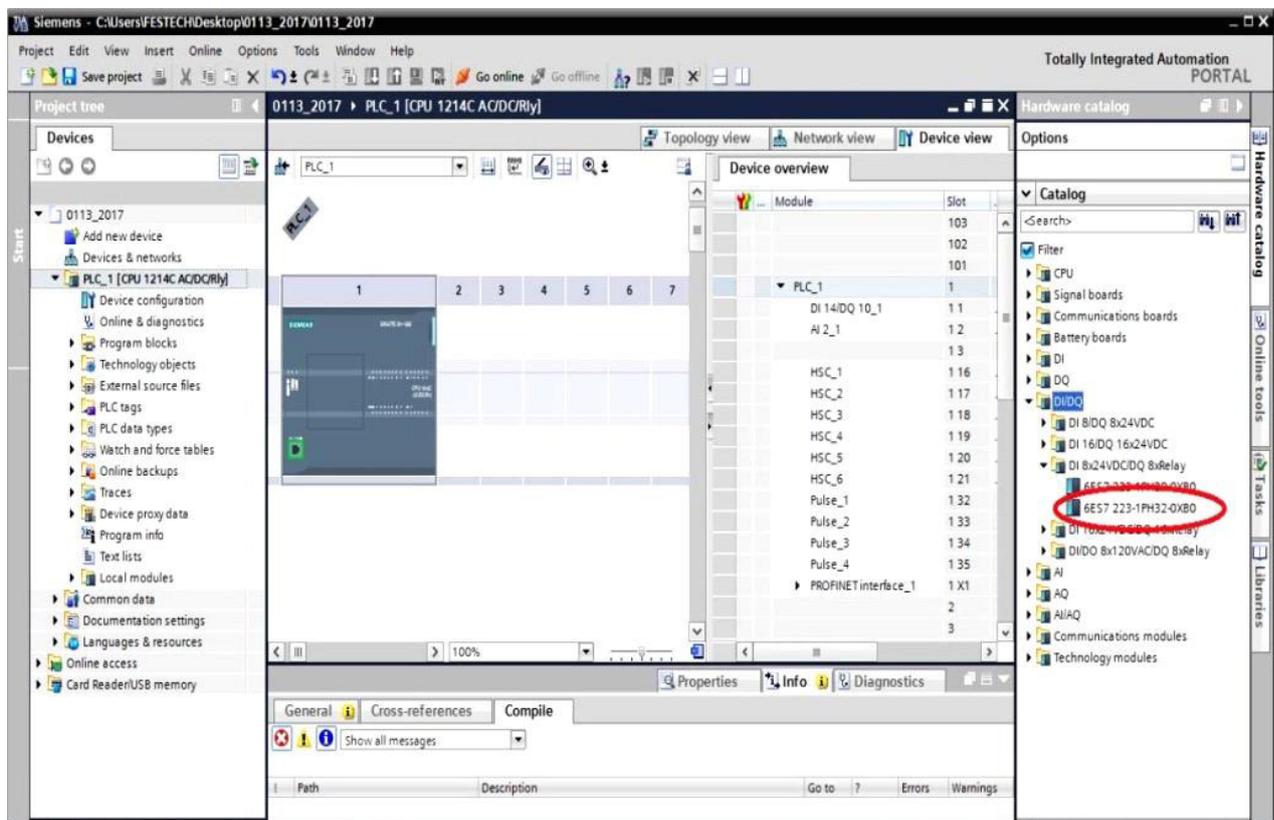


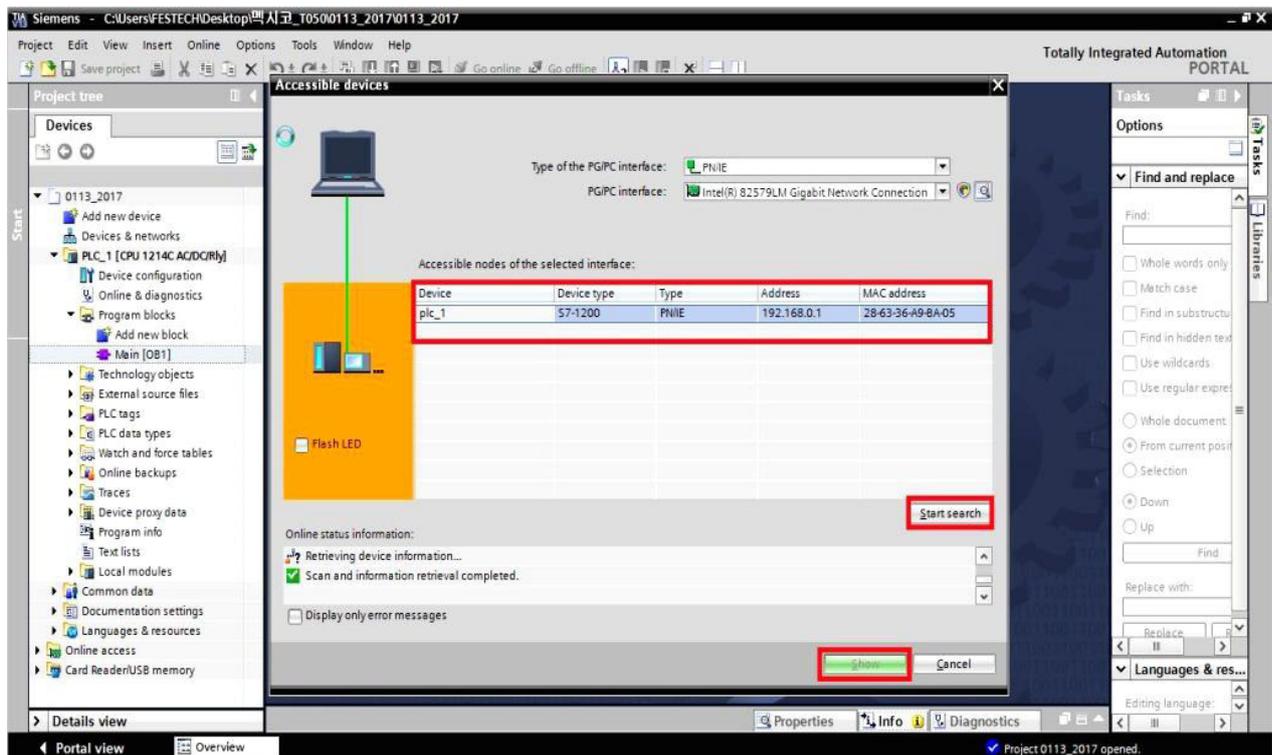
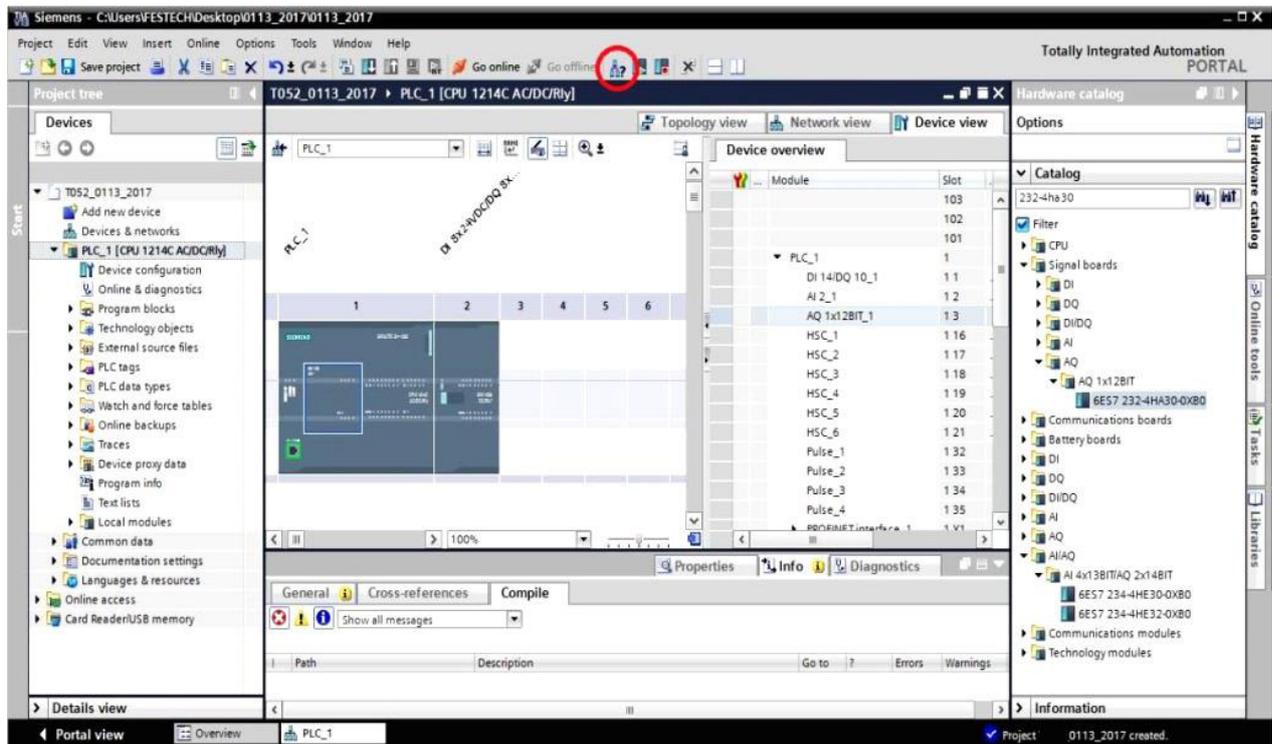
En cliquant sur votre CPU « API » vous verrez d'autres onglets dans la fenêtre des propriétés. Sur l'onglet « Interface PROFINET » saisissez l'adresse : 192.168.0.1 et laissez le masque sur : 255.255.255.0.

Naviguez sur les onglets pour voir les paramètres de la CPU, mais laissez toutes les valeurs par défaut.

Vous pouvez aussi faire un double-clic sur la CPU pour obtenir le mode avancé de la configuration de votre appareil.

Nous allons ajouter un « signal board » (modules en face avant de la CPU). Pour cela, cliquez sur le module « DI/DQ » (Réf : SM1223 DC / RLY (6ES7 223-1PH32-0XB0) et glissez le sur la CPU [11].





2.6 Présentation des blocs de programmation

Dans la programmation d'automate SIEMENS, la structure du programme est séparée en 4 types de blocs différents :

- Bloc d'organisation
- Fonction

- Bloc fonctionnel
- Bloc de données

Nous allons voir les différences et l'utilité de ces blocs.

2.6.1. Blocs d'organisation (OB)

Il existe 7 blocs d'organisation différents :

- OB cyclique (Program cycle), il s'agit de blocs traités de manière cyclique. Ce sont des blocs de code de niveau supérieur dans le programme, dans lesquels vous pouvez programmer des instructions ou appeler d'autres blocs. Le bloc cyclique OB1 est déjà créé à la création du projet.
- OB de démarrage (Startup), le traitement de ces OB est réalisé qu'une fois, lorsque la CPU passe de STOP en RUN. Le traitement de l'OB de démarrage est suivi de celui de l'OB cyclique.
- OB d'alarme temporisée (Time delay interrupt), ils interrompent le traitement cyclique du programme après écoulement d'un temps défini. Vous indiquez le temps de retard dans le paramètre d'entrée de l'instruction étendue "SRT_DINT".
- OB d'alarme cyclique (Cyclic interrupt), ils interrompent le traitement cyclique du programme à intervalles de temps définis. Vous pouvez spécifier les intervalles de temps dans cette boîte de dialogue ou dans les propriétés de l'OB.
- OB d'alarme du processus (Hardware interrupt), ils interrompent le traitement cyclique du programme en réponse à un événement matériel. Vous définissez l'événement matériel dans les propriétés du matériel.
- OB d'erreur de temps (Time error interrupt), ils interrompent le traitement cyclique du programme lorsque le temps de cycle maximum est dépassé. Vous définissez le temps de cycle maximum dans les propriétés de la CPU.
- OB d'alarme de diagnostic (Diagnostic error interrupt), ils interrompent le traitement cyclique du programme lorsque le module pour lequel l'alarme de diagnostic a été activée détecte une erreur.

Vous retrouverez ces informations en ajoutant un OB à votre programme.

2.6.2 Fonctions (FC)

Les fonctions sont des blocs de code sans mémoire.

2.6.3. Blocs fonctionnels (FB)

Les blocs fonctionnels sont des blocs de code qui sauvegardent en permanence leurs valeurs dans des blocs de données d'instance afin qu'il soit possible d'y accéder même après le traitement du bloc.

2.6.4 Blocs de données (DB)

Les blocs de données sont des zones de données dans le programme utilisateur qui contiennent des données utilisateur.

Vous pouvez sélectionner 2 types de bloc :

- un bloc de données global, qui est indépendant de tout autre bloc. (Par exemple nous programmons un DB Global pour toutes les données d'échange entre API et HMI).
- un bloc de données d'instance, qui dépend d'un bloc fonctionnel, il s'agit de la mémoire des valeurs du bloc dont il dépend [10].

Une fois le projet ouvert, cliquez sur « Vue du projet » (en bas, à gauche de la page) ou sur le 4^{ème} onglet « Ouvrir la vue du projet ».

Vous voyez maintenant votre projet complet. Nous allons dans un premier temps ajouter les variables à notre projet.

Cliquez sur l'onglet « API » puis sur l'onglet « Variable API » et double-cliquez sur variable API. Entrez ces valeurs :

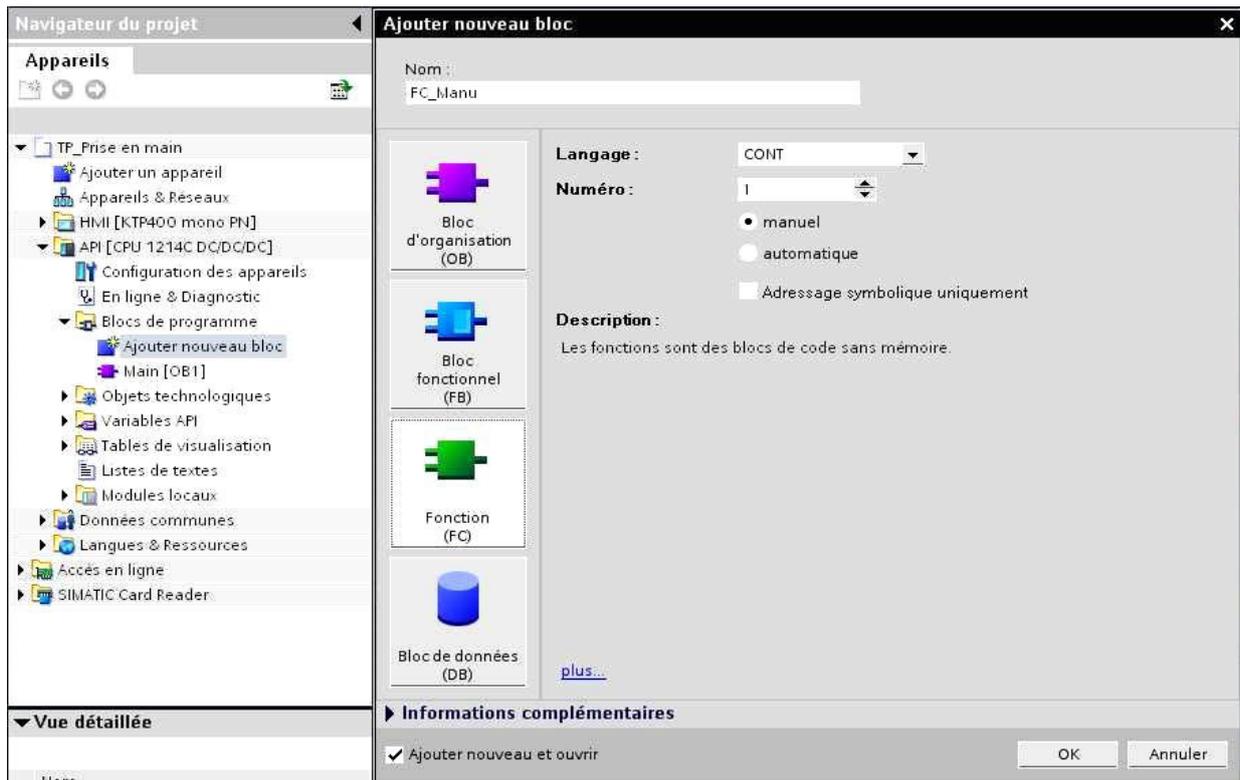
Nom	Table des variables	Type de données	Adresse	Réma...	Access...	Boole...	Modu...
START	Default tag table	Bool	%I0.0				
STOP	Default tag table	Bool	%I0.1				
RESET	Default tag table	Bool	%I0.2				
STEP	Default tag table	Bool	%I0.3				
AU/MA	Default tag table	Bool	%I0.4				
S201	Default tag table	Bool	%I8.0				
S202	Default tag table	Bool	%I8.1				
S203	Default tag table	Bool	%I8.2				
S204	Default tag table	Bool	%I8.3				
S205	Default tag table	Bool	%I8.4				
S206	Default tag table	Bool	%I8.5				
S207(FUNCTION)	Default tag table	Bool	%I8.6				
L10.L12	Default tag table	Bool	%Q0.0				
L11.L13	Default tag table	Bool	%Q0.1				
L20.L22	Default tag table	Bool	%Q0.2				
L21.L23	Default tag table	Bool	%Q0.3				
L30.L32	Default tag table	Bool	%Q0.4				
L31.L33	Default tag table	Bool	%Q0.5				
L40.L42	Default tag table	Bool	%Q0.6				
L41.L43	Default tag table	Bool	%Q0.7				
L14	Default tag table	Bool	%Q8.0				
L15	Default tag table	Bool	%Q8.1				

Nous pouvons alors créer notre fonction de gestion du mode manuel.

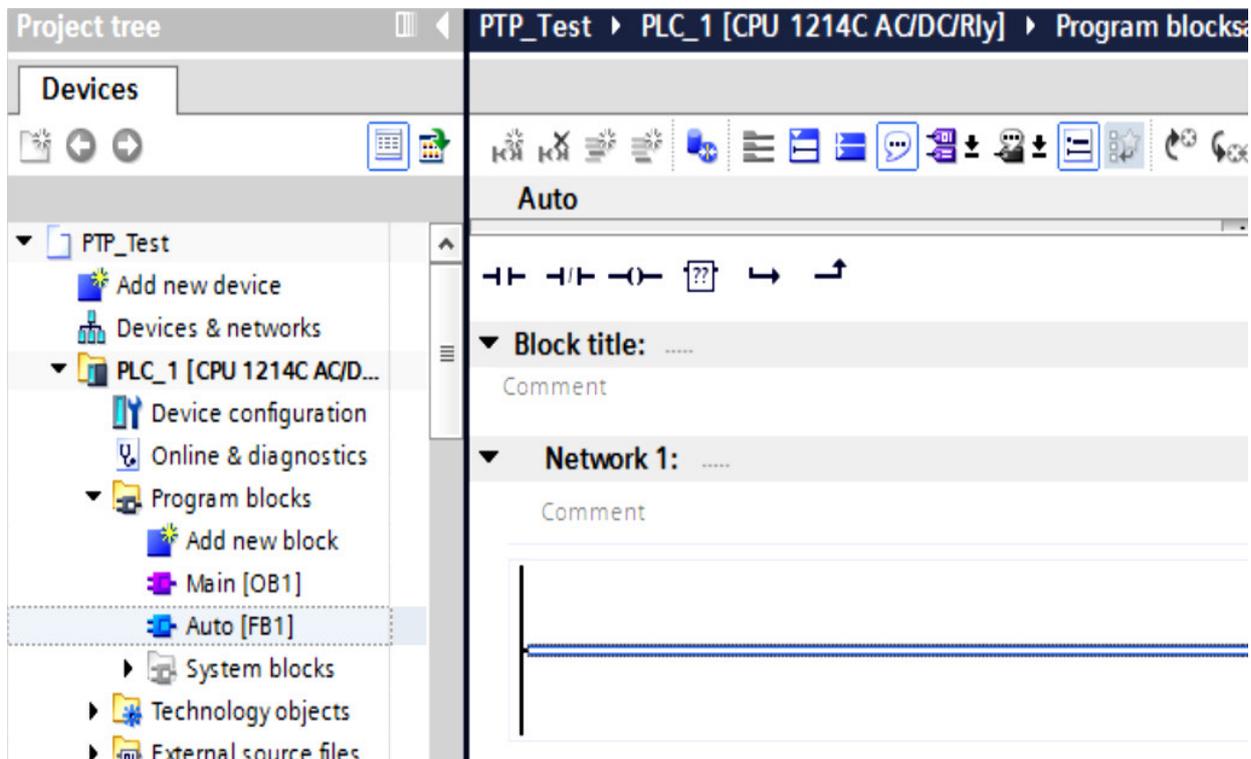
Cliquez sur l'onglet « API » puis sur l'onglet « Blocs de programme » et enfin double-cliquez sur « Ajouter nouveau bloc ».

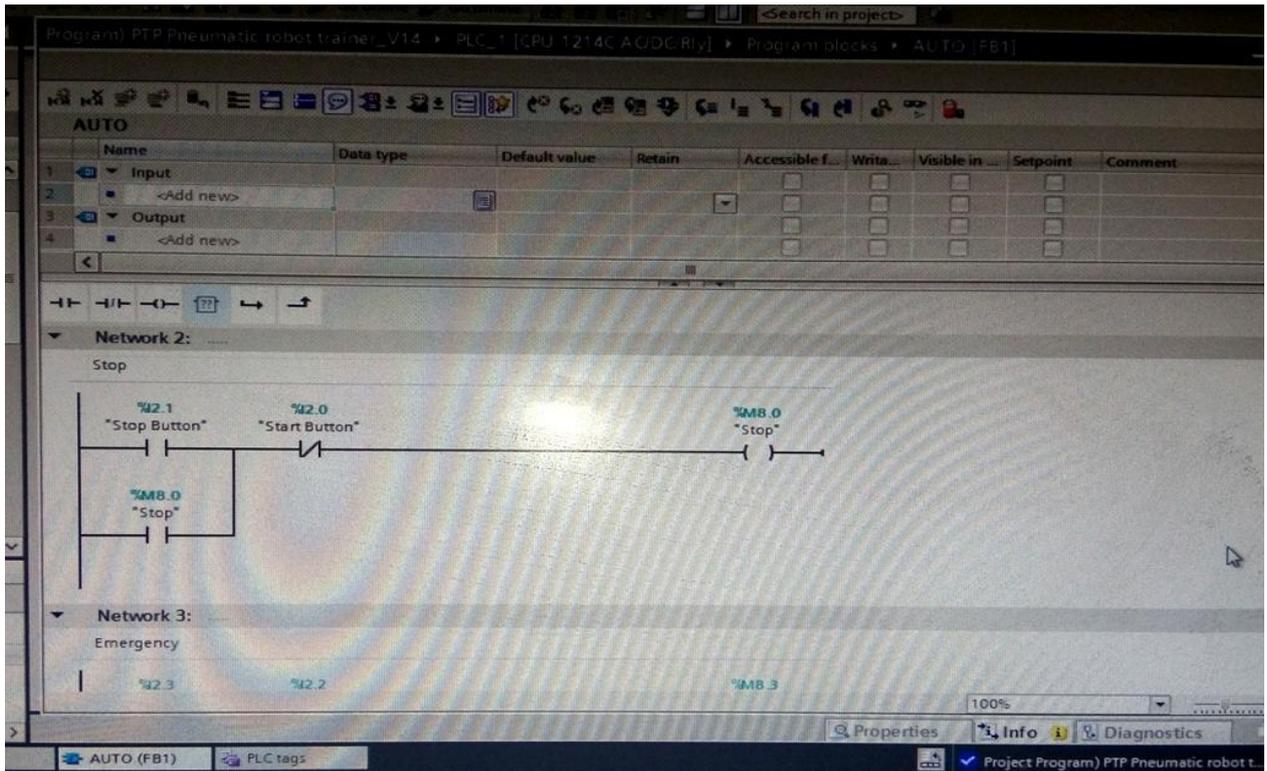
Saisissez le nom, le numéro (cochez manuel) et décochez « Adressage symbolique uniquement »

Puis cliquez sur « OK »

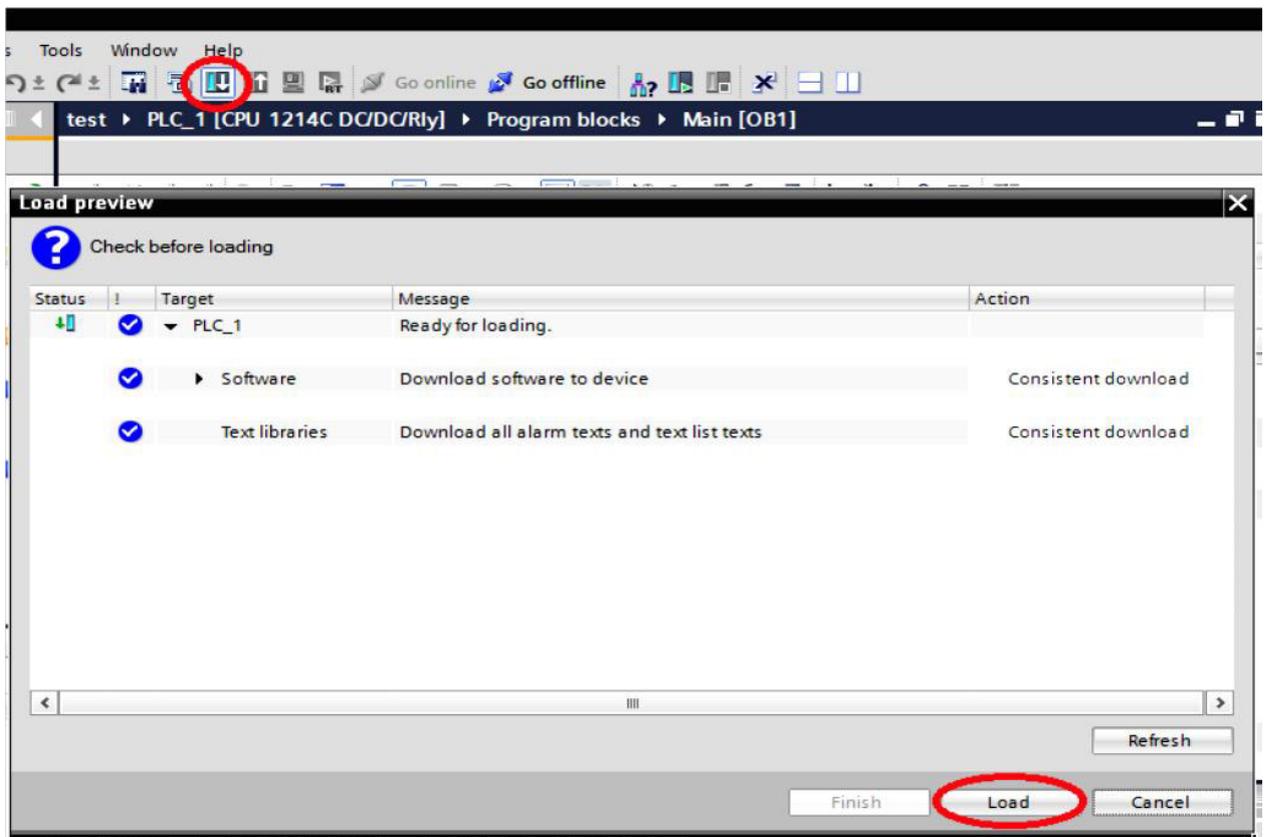


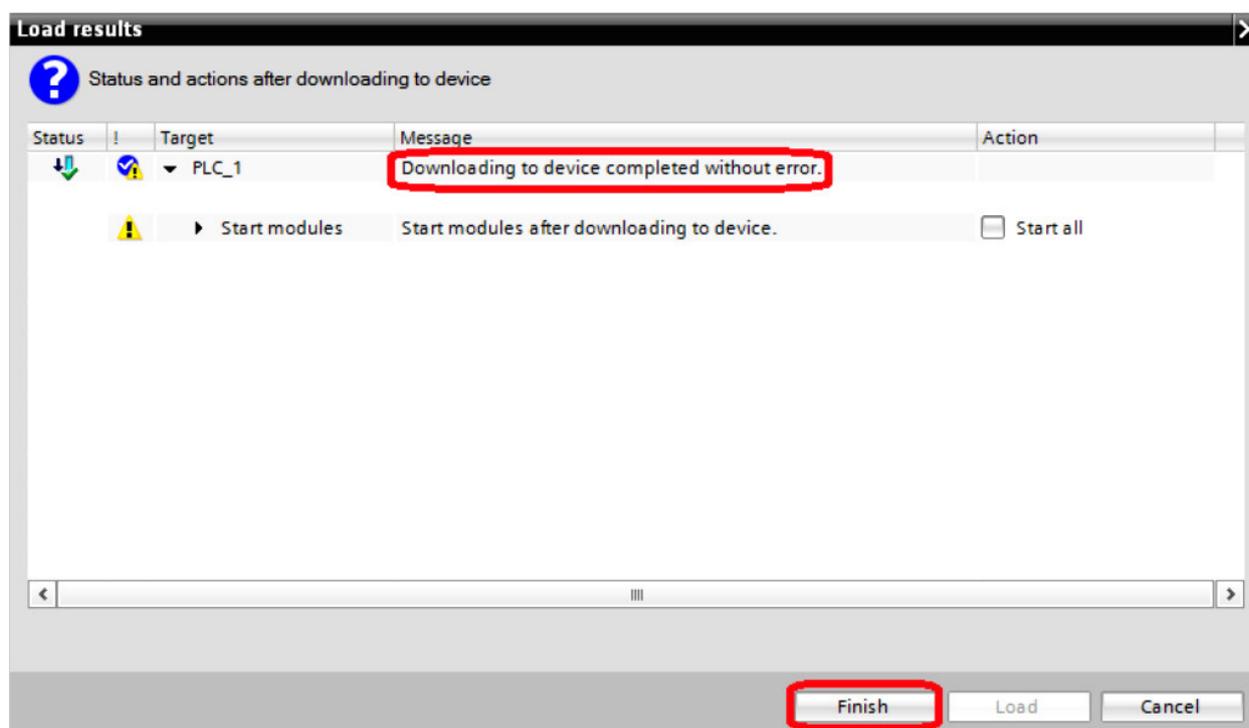
Le bloc s'ouvre automatiquement, il nous reste plus qu'à programmer le mode manuel[10].





Puis charger le programme dans CPU [11].





Chapitre 2 :

2.1 Automate SIEMENS S7-1200-V14

2.1.1 Présentation de l'automate SIEMENS S7-1200-V14

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

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

La CPU fournit un port PROFINET permettant de communiquer par le biais d'un réseau PROFINET. Des modules supplémentaires sont disponibles pour communiquer via les Réseaux PROFIBUS, GPRS, RS485 ou RS232 [10].

Pour ce projet, le S7-1200 sera programmé en langage LADDER à l'aide du logiciel TIA PORTAL sous Windows.

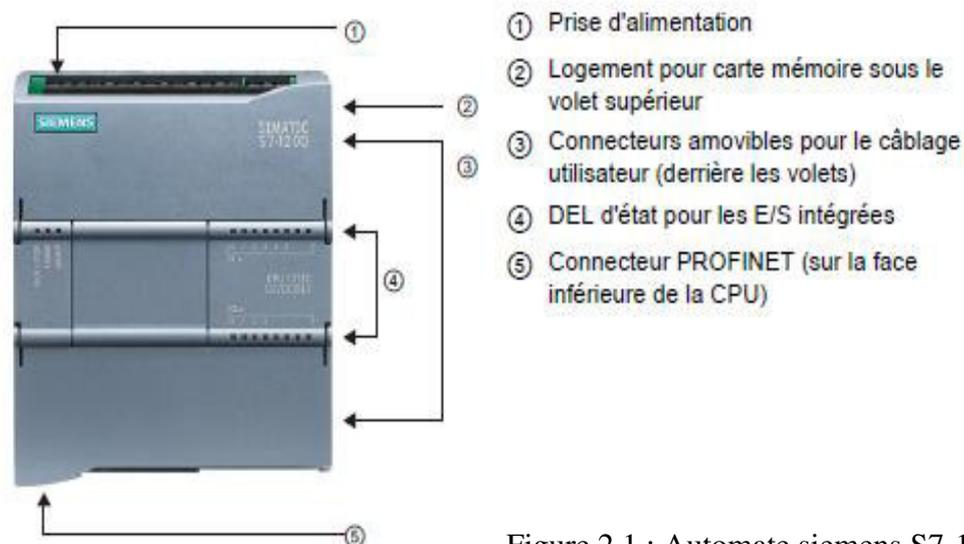


Figure 2.1 : Automate siemens S7-1200[10].

2.2. Possibilités d'extension de la CPU :

La gamme S7-1200 offre divers modules et cartes enfichables pour accroître les capacités de la CPU avec des E/S supplémentaires ou d'autres protocoles de communication.

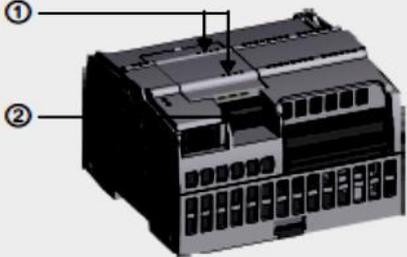
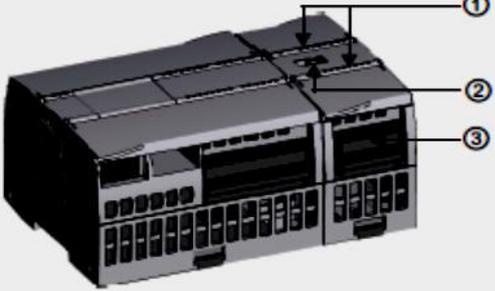
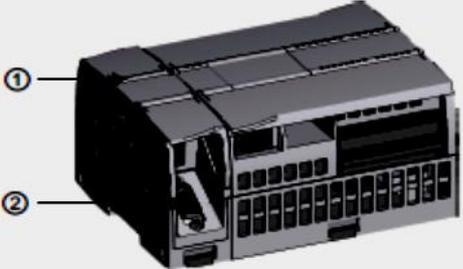
Type de module	Description		
<p>La CPU prend en charge une carte d'extension enfichable :</p> <ul style="list-style-type: none"> • Un Signal Board (SB) fournit des E/S supplémentaires pour votre CPU. Le SB se raccorde à l'avant de la CPU. • Un Communication Board (CB) vous permet d'ajouter un autre port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

Figure 2.2 : Possibilités d'extension de la CPU [10].

2.3. Besoins du système

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

Tableau 2-1 Besoins du système

Matériel/logiciel	Condition de requise
Type de processeur	Pentium M, 1,6 GHz ou similaire
Mémoire vive	1 Go
Espace disponible sur le disque dur	2 Go sur le lecteur système C:\
Systèmes d'exploitation	* Windows XP Professional SP3 * Windows 2003 Server R2 StdE SP2 * Windows 7 Home Premium (STEP 7 Basic uniquement, non pris en charge pour STEP 7 Professional) * Windows 7 (Professional, Enterprise, Ultimate) * Windows 2008 Server StdE R2
Carte graphique	32 Mo RAM
Profondeur de couleur	24 bits
Résolution de l'écran	1024 x 768
Réseau	Ethernet 20Mbits/s ou plus Lecteur optique
Lecteur optique	DVD-ROM

2.4 Logiciel de programmation TIA PORTAL

Dans TIA PORTAL, le projet vous permet d'associer votre programme à toutes les informations nécessaires pour que vous puissiez communiquer avec un API et charger le programme dans cet API. [7]

2.4.1 Présentation de l'adressage

Vous pouvez identifier les paramètres des éléments de commande dans votre programme de manière absolue ou symbolique. Une référence absolue(ou directe) utilise la zone de mémoire et l'adresse de bit ou d'octet pour identifier l'opérande. Une référence symbolique (ou indirect) utilise une combinaison de caractères alphanumérique pour identifier l'opérande. [8]

Les bits d'entrées de l'automate sont notés :

Ii.n	avec n=0, 1,2,..., 7 le numéro du bit, I= entrée	Exemple : I0.0 : entrée 0 bit 0 (Interrupteur)
-------------	---	---

Les bits de sortie de l'automate sont notés :

Qi.n	avec n=0, 1,2,..., 7 le numéro du bit, Q= sortie	Exemple : Q0 .0 sortie0 bit 0 (Lampe)
-------------	---	--

Les bits internes de l'automate sont notés :

Mi.n	avec n=0, 1,2,..., 7 le numéro du bit M=Mémento=mémoire	Exemple : M10 : mémoire 10 (CF0)
-------------	--	---

Les CPU SIMATIC S7-1200 mettent à votre disposition plusieurs langages vous permettant de résoudre une large gamme de tâches d'automatisation. Les deux jeux d'opérations de base disponibles dans une CPU S7-1200 sont le jeu d'opérations SIMATIC et le jeu d'opérations CEI 1131-3. Concernant ce dernier, les jeux d'instruction normalisés CEI proposés sont :

➤ Le langage LD (CONT)

Vous devez, lors de chaque création de programme, choisir :

- Le jeu d'opérations convenant le mieux à votre application (SIMATIC ou CEI 1131-3).
- Le type d'éditeur convenant le mieux à vos besoins de programmation (liste d'instructions, schéma à contacts ou logigramme). [8]

2.4.2 Editeur CONT (schéma à contacts)

L'éditeur schéma à contacts (CONT) de SETP7-Micro/WIN 32 permet de créer des programmes qui ressemblent à un schéma de câblage électrique. La programmation en CONT est la méthode choisie par de nombreux programmeurs d'automates programmables et par le personnel de maintenance ; c'est un langage qui convient également très bien aux programmeurs débutants, fondamentalement les programmes CONT permettent à la CPU d'émuler le trajet de courant

électrique partant d'une source de tension, à travers une série de conditions d'entrée logiques validant, à leur tour, des conditions de sortie logiques. Généralement on subdivise le code en sections de petite taille et faciles à comprendre, souvent appelées "réseaux". L'exécution du programme se fait réseau par réseau, de gauche à droite et de haut en bas, comme prescrit par le programme. Lorsque la CPU a atteint la fin du programme, elle recommence au début du programme.

Les différentes opérations sont représentées par des symboles graphiques de trois types fondamentaux : [8]

▪ Contacts

▪ Les contacts représentent des commutateurs à travers lesquels le courant peut circuler. Le courant circule à travers un contact à fermeture uniquement si ce contact est fermé (1 logique) ; le courant circule à travers un contact à ouverture ou inversé (NOT) uniquement si ce contact est ouvert (0 logique).

Symbole du contact :

Le contact normalement ouvert (NO) (en NO : normally open) : 

Le contact normalement fermé (NF) (en NC : normally closed) : 

* Le contact à fermeture est fermé (activé) lorsque la valeur du bit affecté est égale à 1.

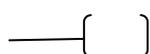
* Le contact à ouverture est fermé (activé) lorsque la valeur du bit affecté est égale à 0.

* Des contacts connectés en série créent des réseaux de logique ET.

* Des contacts connectés en parallèle créent des réseaux de logique OU.

*Bobine

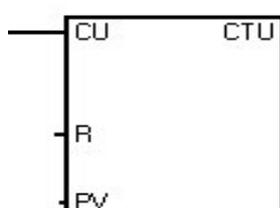
Les bobines représentent généralement des résultats de sortie logiques, tels que lampes, démarreurs de moteur, relais intermédiaires, conditions de sortie internes, etc.

Symbole de la bobine : [8] 

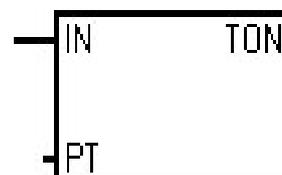
▪ Boîte

Les boîtes représentent généralement une fonction (par exemple une temporisation, un compteur ou une opération arithmétique) qui est exécutée lorsque le courant atteint cette boîte. [8]

Symbole de la boîte :



Compteur



Temporisateur

2.5 Présentation du logiciel

2.5.1 Introduction

Le logiciel « Totally Integrated Automation Portal » (TIA) est un logiciel de programmation des automates de la gamme S7-1200 et des pupitres opérateur de la gamme KTP. Celui-ci est composé de STEP7 Basic et Win CC Basic. Il reprend la même philosophie de programmation que le logiciel STEP7 Pro avec une interface simplifiée et l'intégration de la programmation de pupitre opérateur.

Le souhait de SIEMENS est d'intégrer toutes leurs gammes de produits pour un seul logiciel.

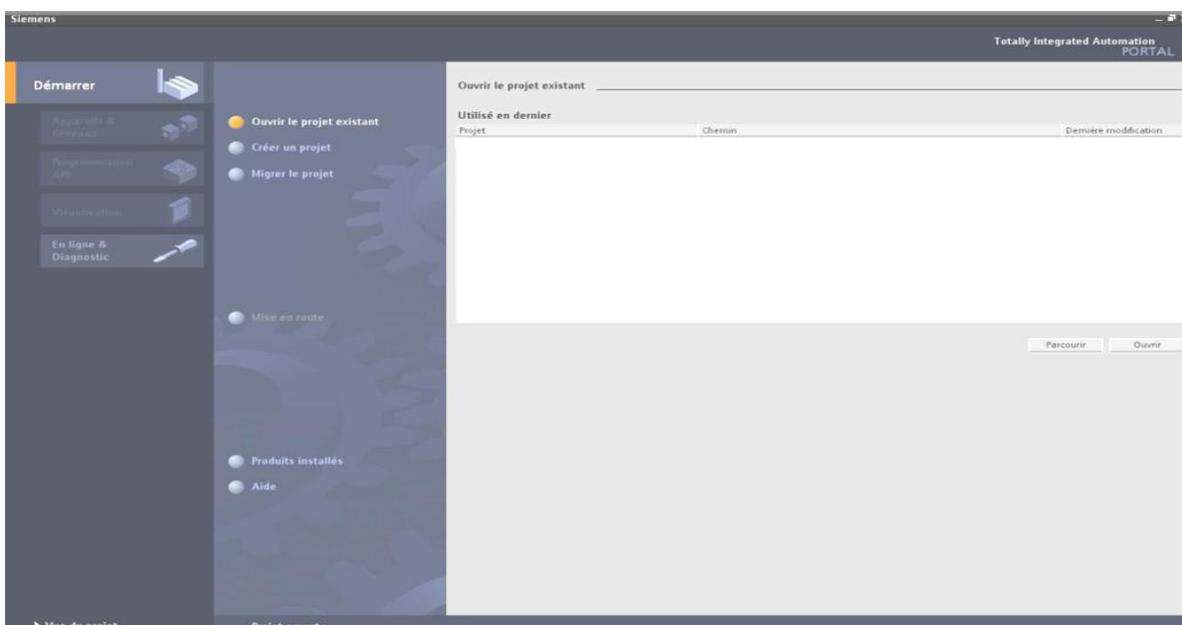
Lors de l'installation vous vous apercevrez que les licences sont intégrées au CD d'installation et se placent sur votre disque dur automatique, par contre celles-ci sont verrouillées, vous ne pouvez donc ni les supprimer, ni les déplacer [10].

2.5.2 Création d'un projet

Une fois le logiciel entièrement installé, cliquez sur l'icône pour le démarrer



Voici la page de démarrage que vous verrez :

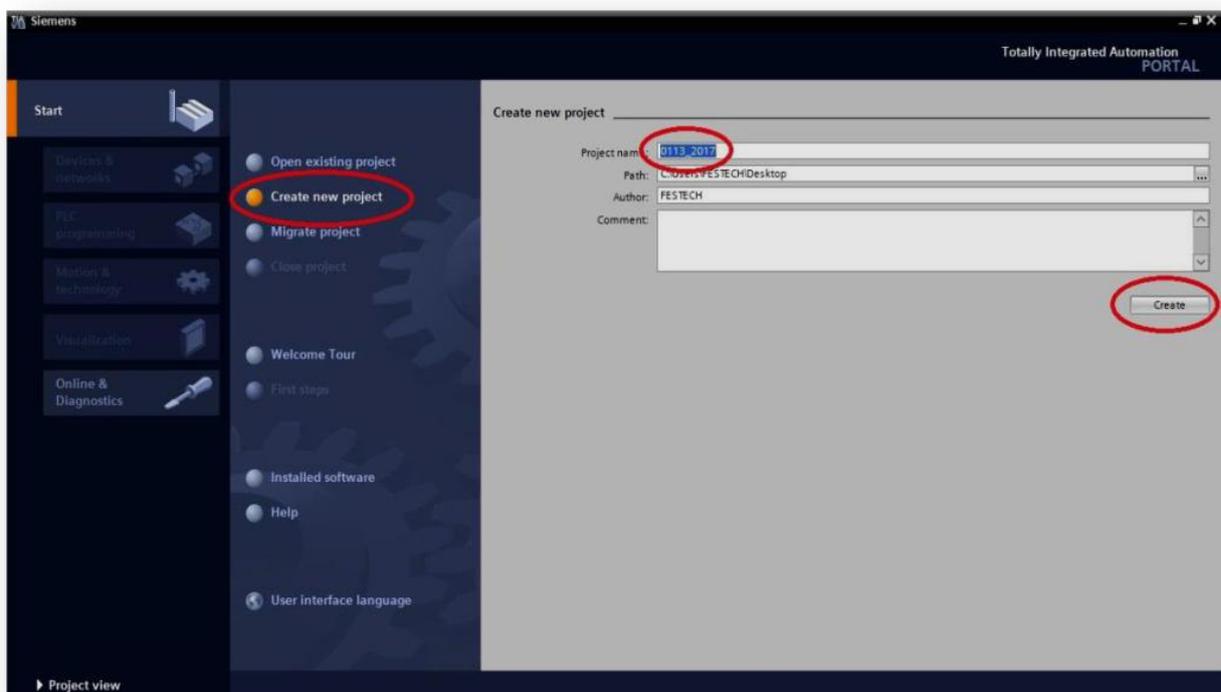


Vous pouvez sur cette page :

- Visualiser les projets déjà créés et les ouvrir.
- Créer un nouveau projet.
- Migrer un projet existant.
- Se mettre en ligne avec un automate S7-1200 ou un pupitre KTP et avoir accès au diagnostic.

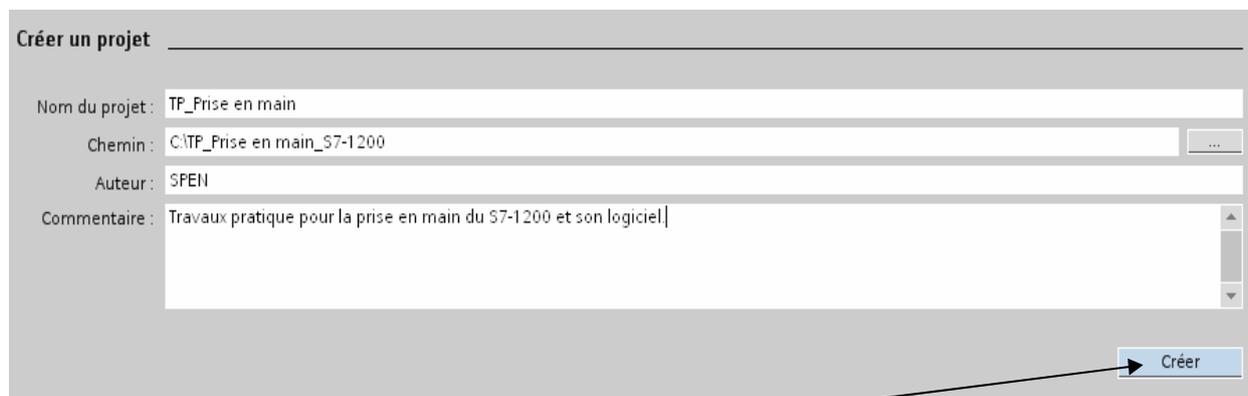
En bas de cette page vous voyez le projet ouvert et vous avez l'accès à ce projet en cliquant sur « Vue du projet ».

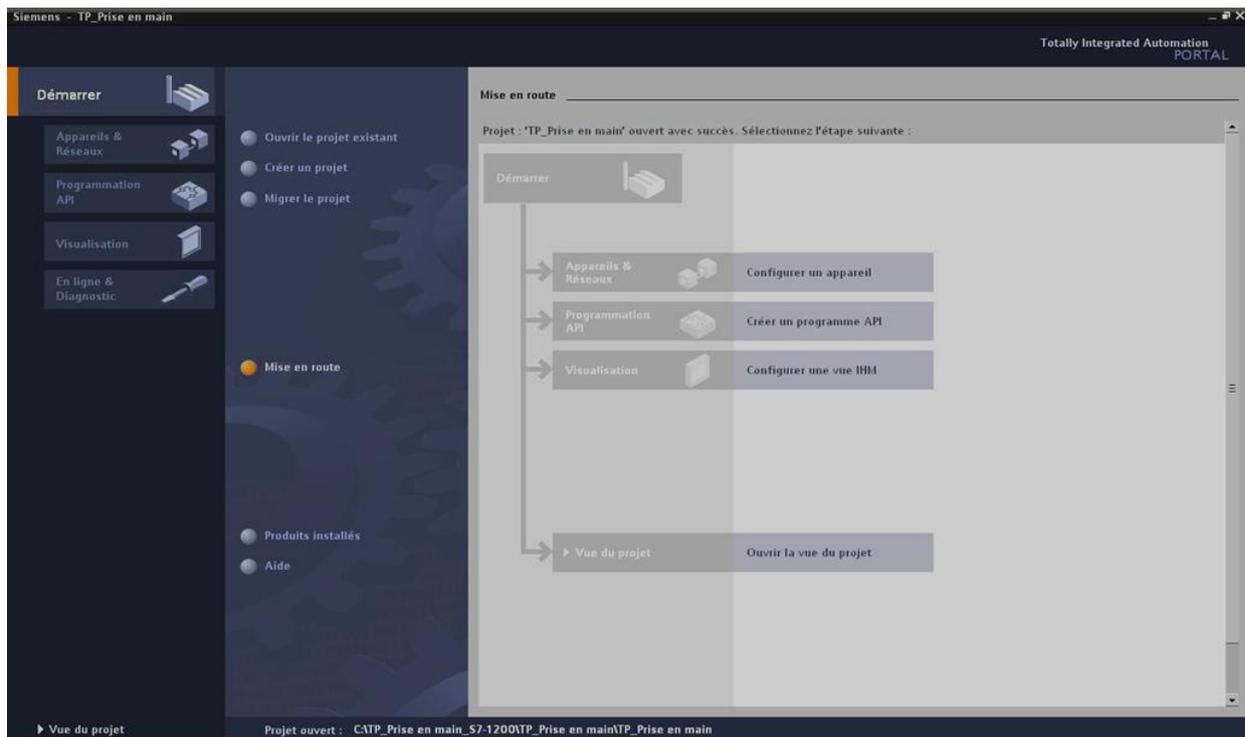
Nous allons dans un premier temps créer un nouveau projet, cliquez pour cela sur l'onglet « Créer un projet » :



Le projet étant créé, nous allons maintenant pouvoir saisir la configuration matérielle

2.5.3 Création d'une configuration matérielle

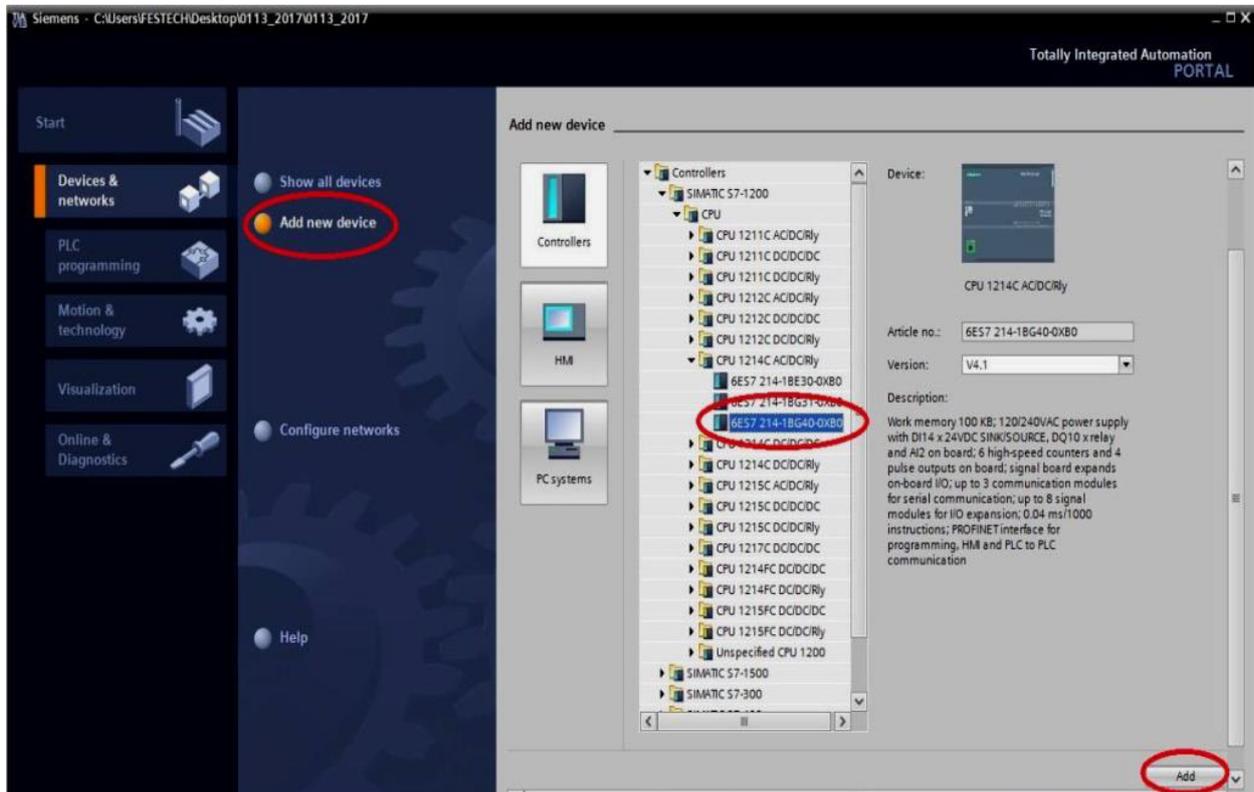




Cliquez sur « Configurer un appareil », puis sur « Ajouter un appareil ».

Nous allons tout d'abord ajouter la CPU, si vous ne connaissez pas la référence vous pouvez cliquer sur « CPU 1214 non spécifiée », vous pourrez alors la saisir plus tard ou la configuration matérielle se mettra à jour lors de la connexion avec la dite CPU.

Nous allons ajouter une CPU 1214C AC/DC/Rly > 6ES7 214-1BG40-0XB0) [11].

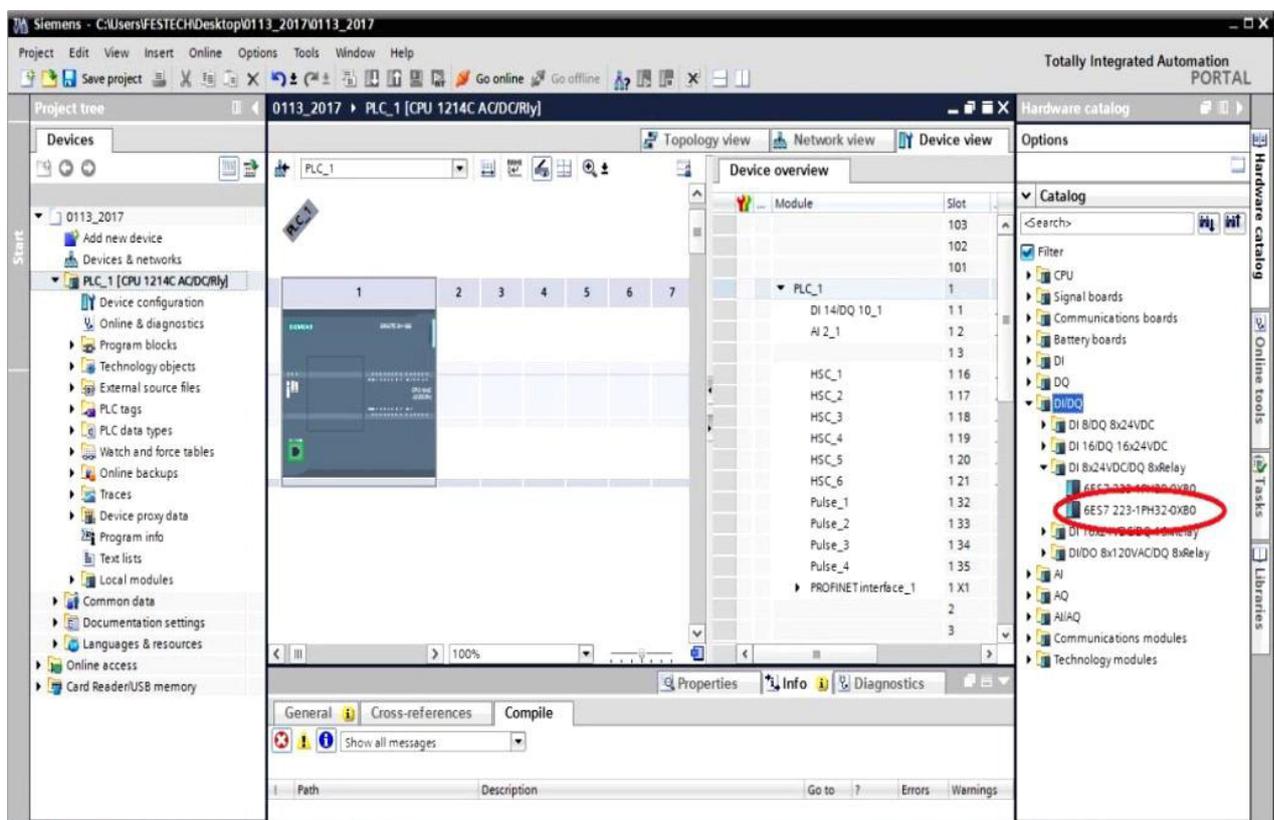


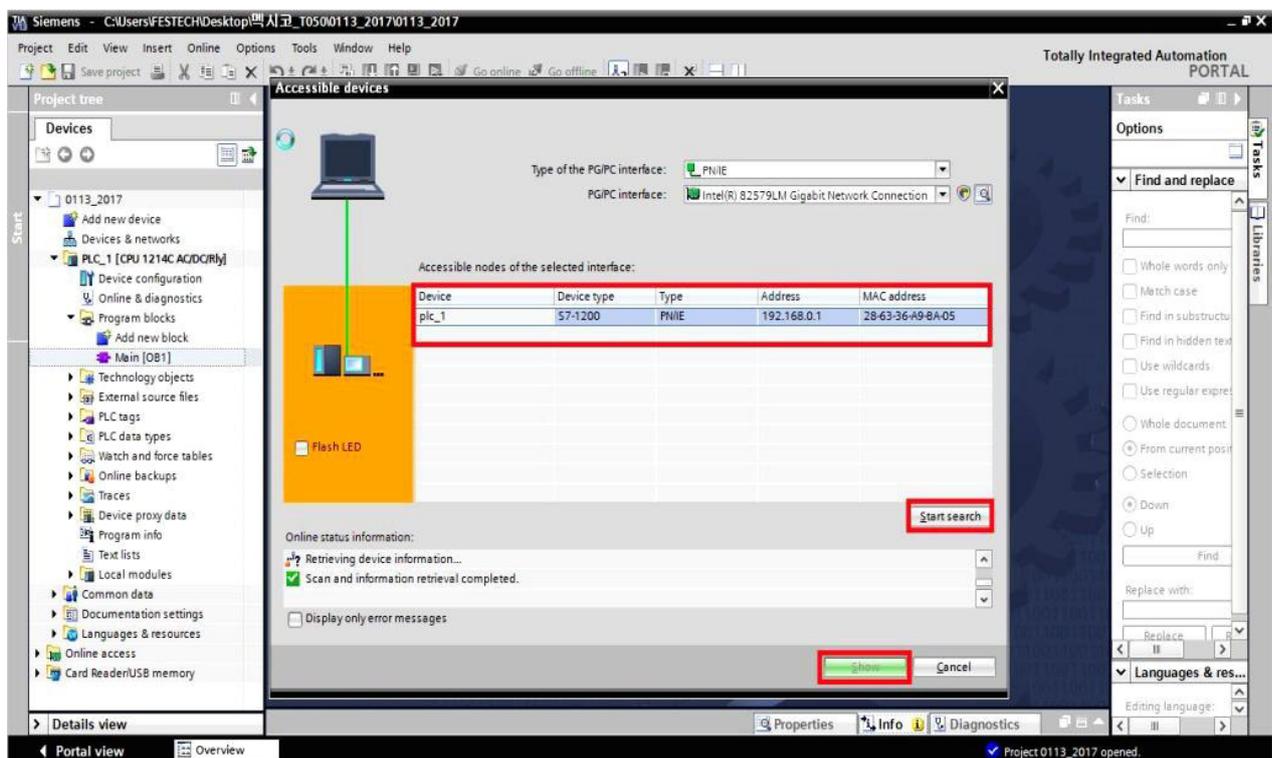
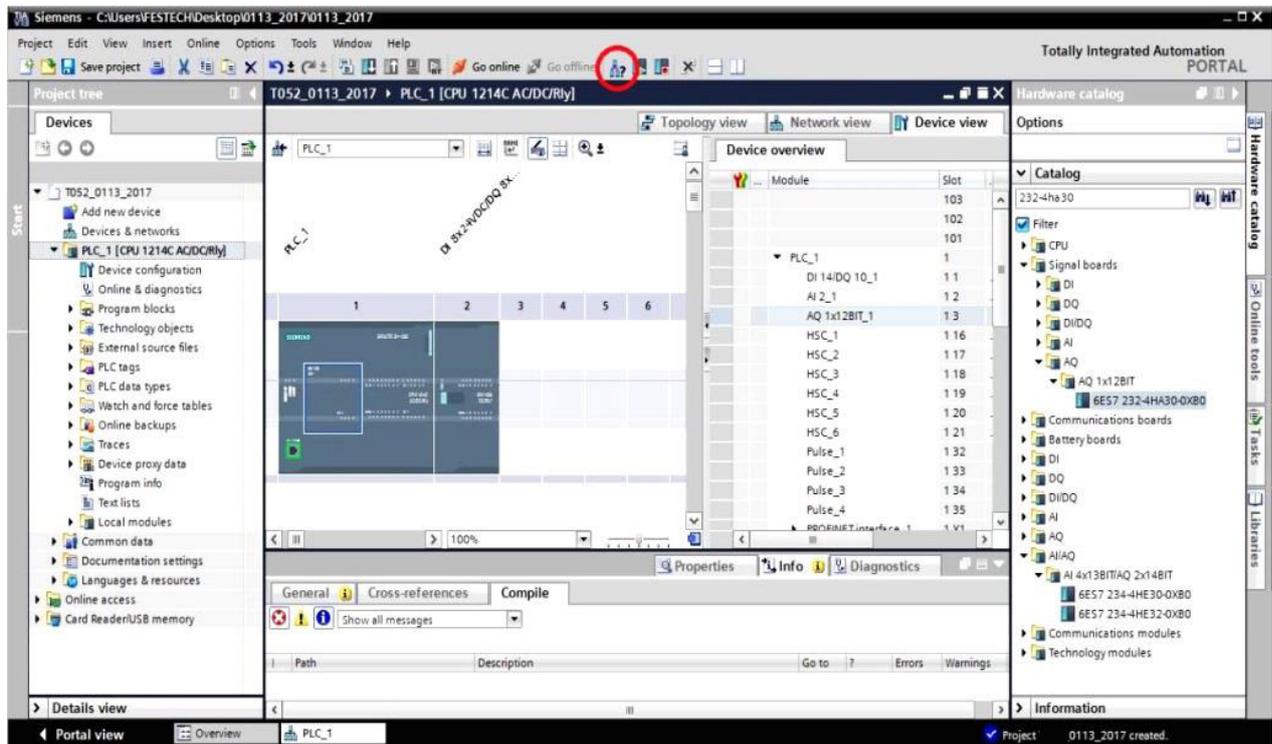
En cliquant sur votre CPU « API » vous verrez d'autres onglets dans la fenêtre des propriétés. Sur l'onglet « Interface PROFINET » saisissez l'adresse : 192.168.0.1 et laissez le masque sur : 255.255.255.0.

Naviguez sur les onglets pour voir les paramètres de la CPU, mais laissez toutes les valeurs par défaut.

Vous pouvez aussi faire un double-clic sur la CPU pour obtenir le mode avancé de la configuration de votre appareil.

Nous allons ajouter un « signal board » (modules en face avant de la CPU). Pour cela, cliquez sur le module « DI/DQ » (Réf : SM1223 DC / RLY (6ES7 223-1PH32-0XB0) et glissez le sur la CPU [11].





2.6 Présentation des blocs de programmation

Dans la programmation d'automate SIEMENS, la structure du programme est séparée en 4 types de blocs différents :

- Bloc d'organisation
- Fonction

- Bloc fonctionnel
- Bloc de données

Nous allons voir les différences et l'utilité de ces blocs.

2.6.1. Blocs d'organisation (OB)

Il existe 7 blocs d'organisation différents :

- OB cyclique (Program cycle), il s'agit de blocs traités de manière cyclique. Ce sont des blocs de code de niveau supérieur dans le programme, dans lesquels vous pouvez programmer des instructions ou appeler d'autres blocs. Le bloc cyclique OB1 est déjà créé à la création du projet.
- OB de démarrage (Startup), le traitement de ces OB est réalisé qu'une fois, lorsque la CPU passe de STOP en RUN. Le traitement de l'OB de démarrage est suivi de celui de l'OB cyclique.
- OB d'alarme temporisée (Time delay interrupt), ils interrompent le traitement cyclique du programme après écoulement d'un temps défini. Vous indiquez le temps de retard dans le paramètre d'entrée de l'instruction étendue "SRT_DINT".
- OB d'alarme cyclique (Cyclic interrupt), ils interrompent le traitement cyclique du programme à intervalles de temps définis. Vous pouvez spécifier les intervalles de temps dans cette boîte de dialogue ou dans les propriétés de l'OB.
- OB d'alarme du processus (Hardware interrupt), ils interrompent le traitement cyclique du programme en réponse à un événement matériel. Vous définissez l'événement matériel dans les propriétés du matériel.
- OB d'erreur de temps (Time error interrupt), ils interrompent le traitement cyclique du programme lorsque le temps de cycle maximum est dépassé. Vous définissez le temps de cycle maximum dans les propriétés de la CPU.
- OB d'alarme de diagnostic (Diagnostic error interrupt), ils interrompent le traitement cyclique du programme lorsque le module pour lequel l'alarme de diagnostic a été activée détecte une erreur.

Vous retrouverez ces informations en ajoutant un OB à votre programme.

2.6.2 Fonctions (FC)

Les fonctions sont des blocs de code sans mémoire.

2.6.3. Blocs fonctionnels (FB)

Les blocs fonctionnels sont des blocs de code qui sauvegardent en permanence leurs valeurs dans des blocs de données d'instance afin qu'il soit possible d'y accéder même après le traitement du bloc.

2.6.4 Blocs de données (DB)

Les blocs de données sont des zones de données dans le programme utilisateur qui contiennent des données utilisateur.

Vous pouvez sélectionner 2 types de bloc :

- un bloc de données global, qui est indépendant de tout autre bloc. (Par exemple nous programmons un DB Global pour toutes les données d'échange entre API et HMI).
- un bloc de données d'instance, qui dépend d'un bloc fonctionnel, il s'agit de la mémoire des valeurs du bloc dont il dépend [10].

Une fois le projet ouvert, cliquez sur « Vue du projet » (en bas, à gauche de la page) ou sur le 4^{ème} onglet « Ouvrir la vue du projet ».

Vous voyez maintenant votre projet complet. Nous allons dans un premier temps ajouter les variables à notre projet.

Cliquez sur l'onglet « API » puis sur l'onglet « Variable API » et double-cliquez sur variable API. Entrez ces valeurs :

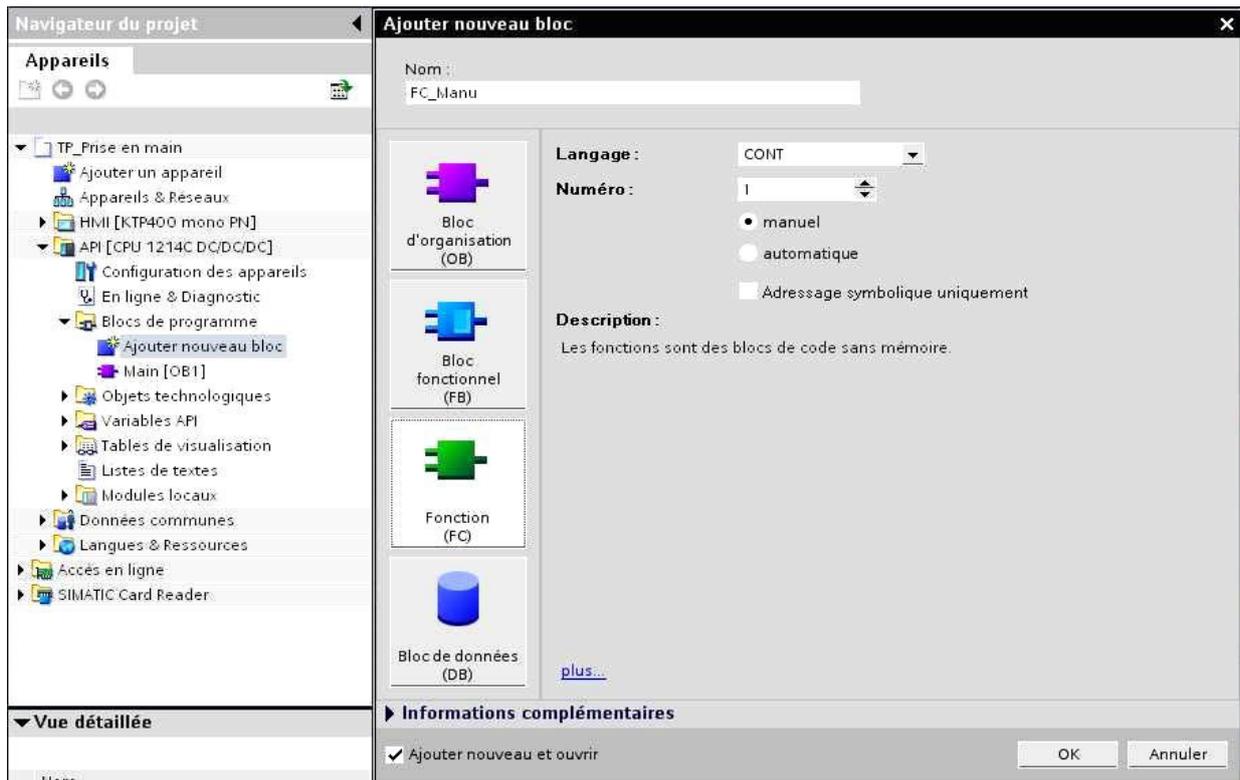
Nom	Table des variables	Type de données	Adresse	Réma...	Access
START	Default tag table	Bool	%I0.0				
STOP	Default tag table	Bool	%I0.1				
RESET	Default tag table	Bool	%I0.2				
STEP	Default tag table	Bool	%I0.3				
AU/MA	Default tag table	Bool	%I0.4				
S201	Default tag table	Bool	%I8.0				
S202	Default tag table	Bool	%I8.1				
S203	Default tag table	Bool	%I8.2				
S204	Default tag table	Bool	%I8.3				
S205	Default tag table	Bool	%I8.4				
S206	Default tag table	Bool	%I8.5				
S207(FUNCTION)	Default tag table	Bool	%I8.6				
L10.L12	Default tag table	Bool	%Q0.0				
L11.L13	Default tag table	Bool	%Q0.1				
L20.L22	Default tag table	Bool	%Q0.2				
L21.L23	Default tag table	Bool	%Q0.3				
L30.L32	Default tag table	Bool	%Q0.4				
L31.L33	Default tag table	Bool	%Q0.5				
L40.L42	Default tag table	Bool	%Q0.6				
L41.L43	Default tag table	Bool	%Q0.7				
L14	Default tag table	Bool	%Q8.0				
I.15	Default tag table	Bool	%Q8.1				

Nous pouvons alors créer notre fonction de gestion du mode manuel.

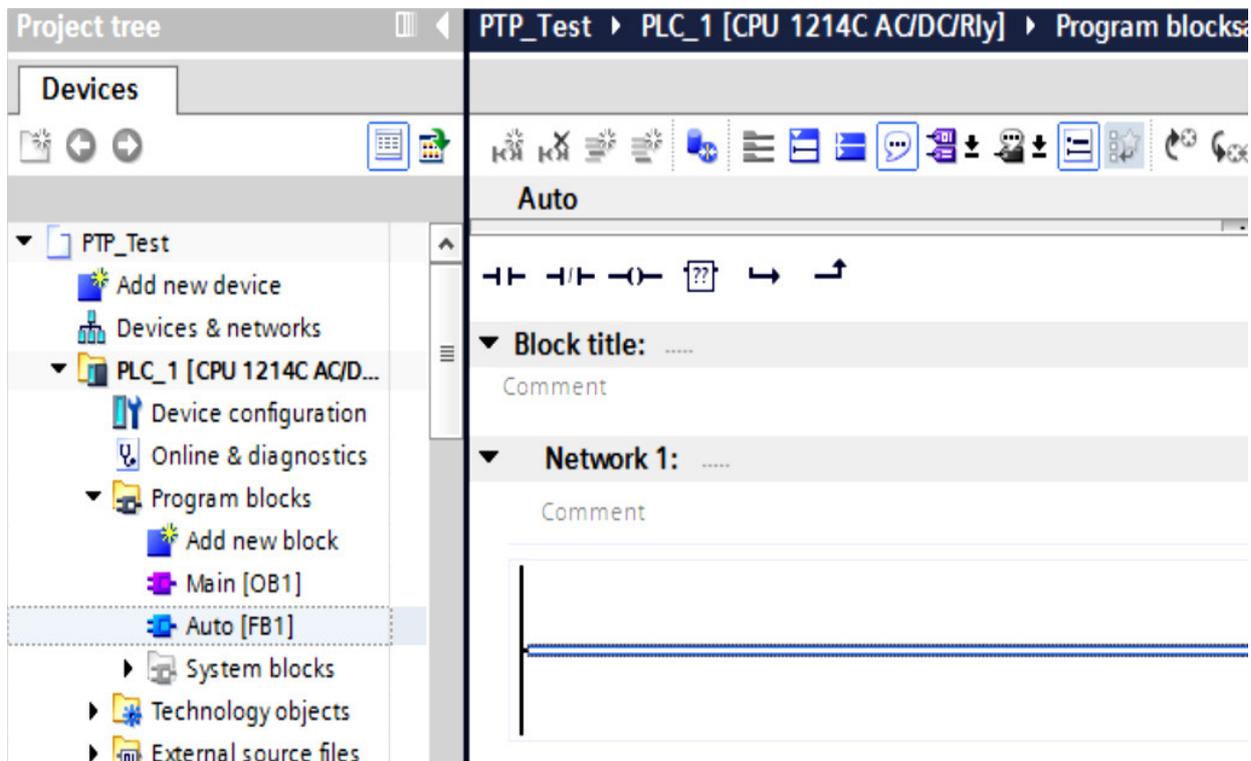
Cliquez sur l'onglet « API » puis sur l'onglet « Blocs de programme » et enfin double-cliquez sur « Ajouter nouveau bloc ».

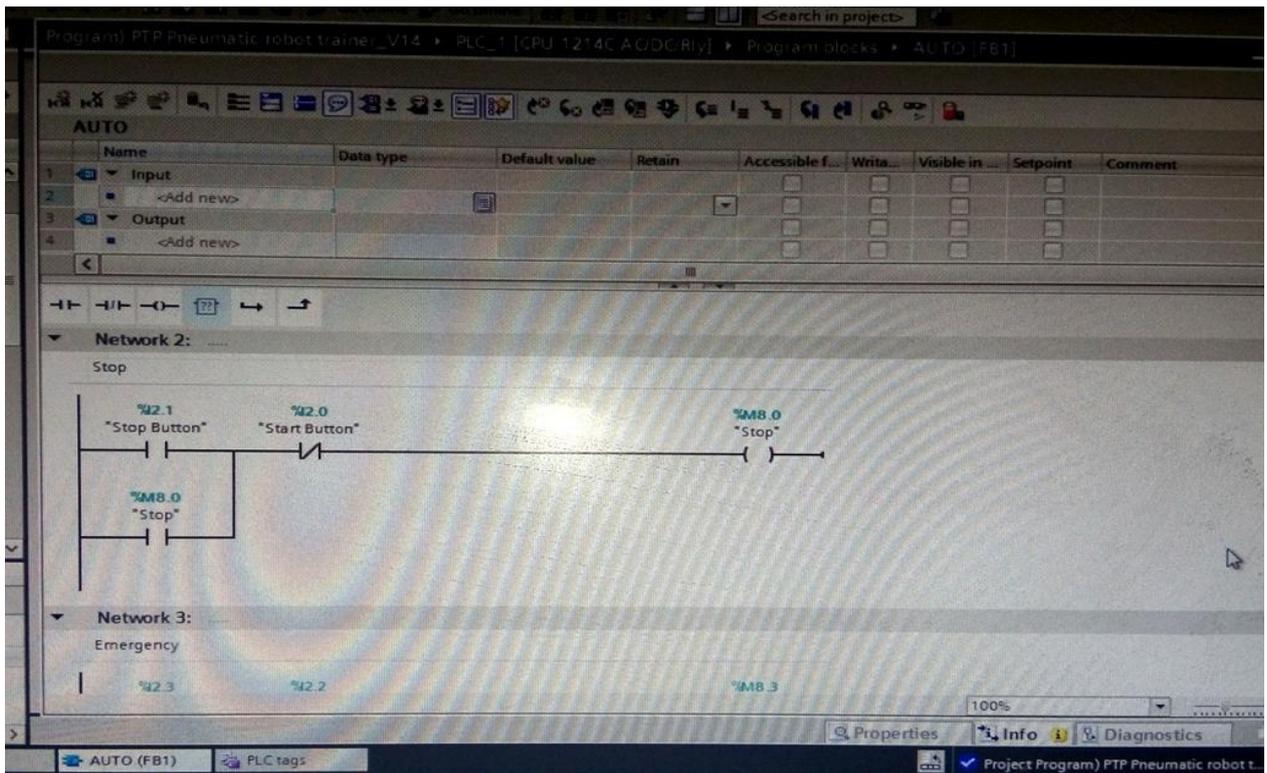
Saisissez le nom, le numéro (cochez manuel) et décochez « Adressage symbolique uniquement »

Puis cliquez sur « OK »

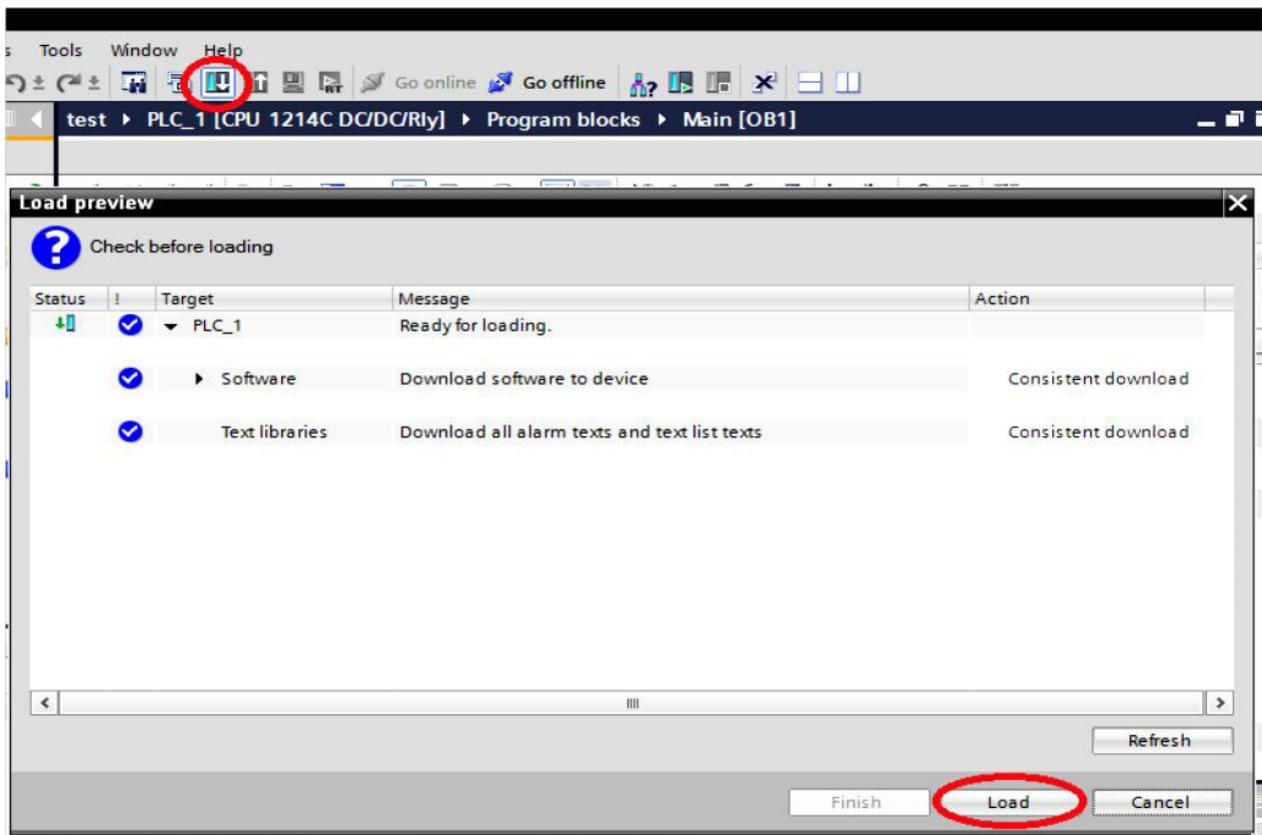


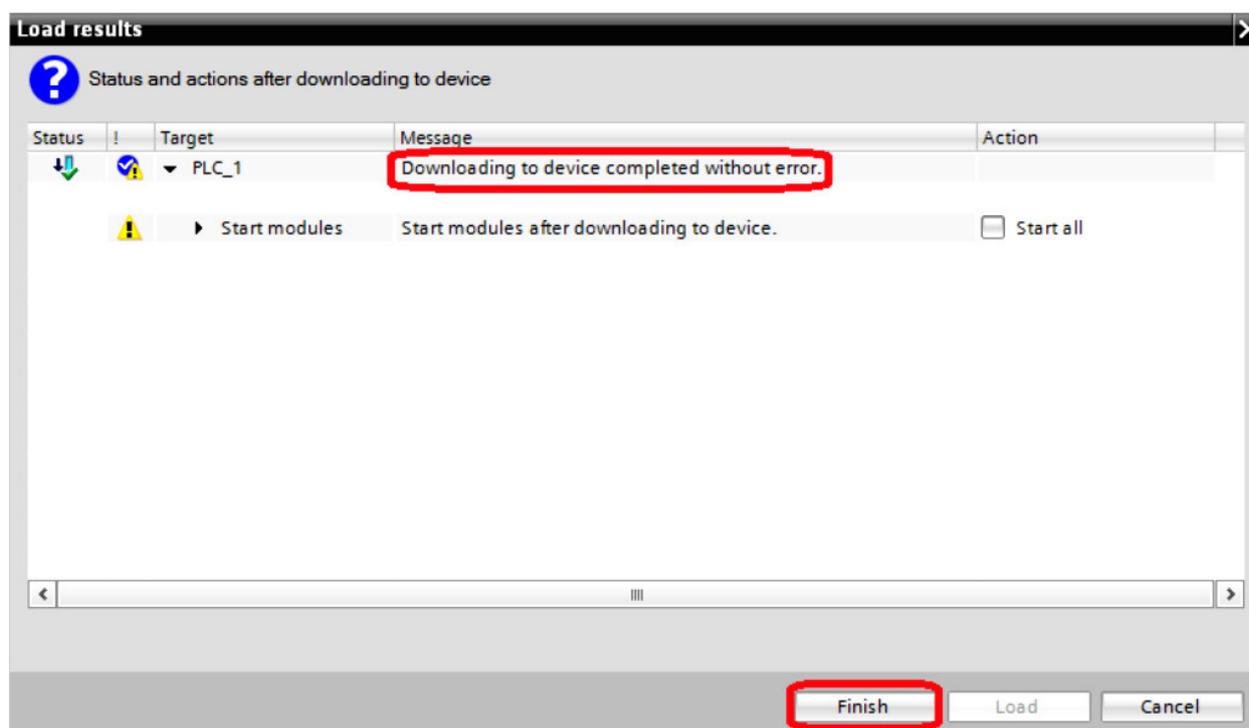
Le bloc s'ouvre automatiquement, il nous reste plus qu'à programmer le mode manuel[10].





Puis charger le programme dans CPU [11].





Chapitre 3

Commande et programmation du système PTP ROBOT PNEUMATIQUE TRAINER

3.1 Introduction

Dans ce chapitre on va présenter la description de cahier des charges et la programmation du système PTP ROBOT PNEUMATIQUE TRAINER par un automate programmable siemens (SIMATIC S7-1200) à l'aide d'un logiciel de programmation TIA PORTAL.

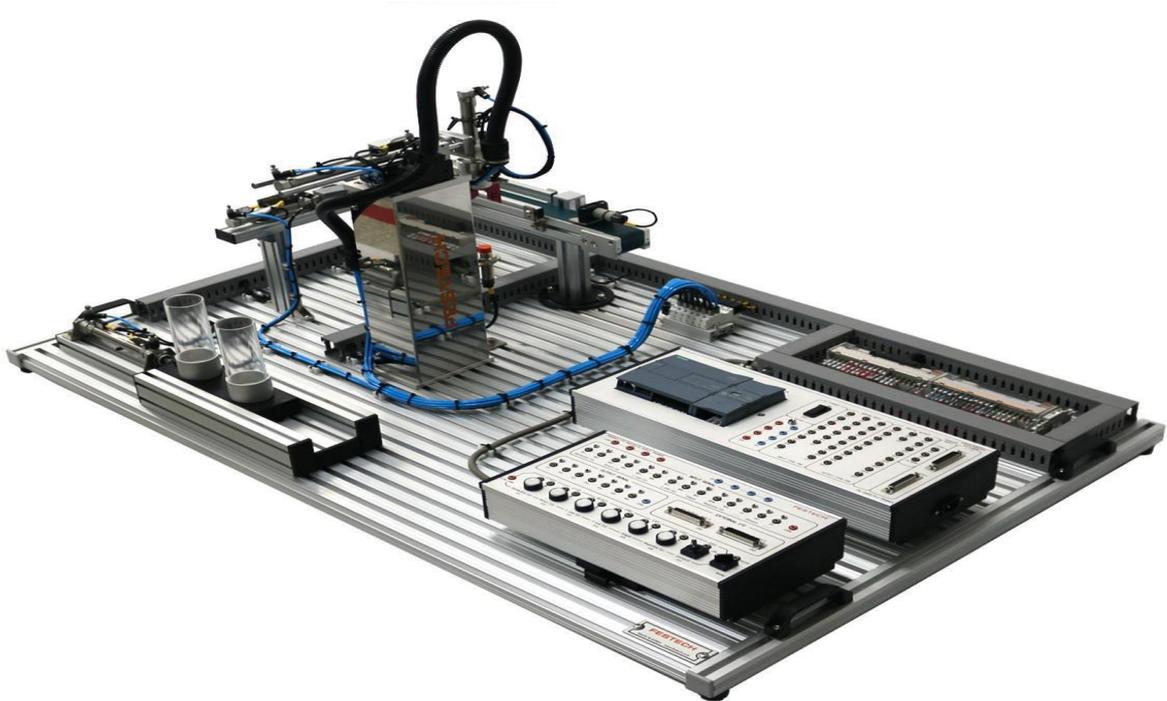


Figure 3.1 : PTP ROBOT PNEUMATIQUE TRAINER [11].

Fiche technique :

- 1) Diamètres : 1200*750*650 mm
- 2) Poids : 40Kg.
- 3) Pression : 4/6 bar.
- 4) Tension d'alimentation : 220V.

3.2 Les composants utilisés

Fin de cours	
Distributeur d'air	
Vérin pneumatique	
Capteur à aimant	
Vérin pneumatique	
Compresseur	
Automate Siemens 1200	
Capteur	

Tableau 3.1 : Les éléments utilisés dans le ROBOT PNEUMATIQUE TRAINER [10].

3.3 Cahier des charges

Un capteur de proximité (S1) indique la présence d'un échantillon des pièces au magasin. S'il existe le vérin numéro 1 de distribution déplace l'échantillon vers le convoyeur jusqu'au l'activation du capteur Dis Avancé 1B2 le moteur démarre et déplace le convoyeur jusqu' au fin de course S2. Le vérin numéro 2 de l'axe X est avancé ensuite le vérin 3 de l'axe Z descend et un autre vérin des pinces prend l'échantillon après fermeture et puis le vérin 3 se lève , le vérin 2 recule puis le vérin 3 descend jusqu'à le détecteur des matériaux ferreux et non ferreux après le test en certain temps le vérin 3 se lève et un autre vérin 4 fait une rotation ensuite le vérin 2 avancée.

Si l'échantillon est en plastic le vérin (Gripper) ouvre le pince et jette le au nouveau magasin sinon il est en aluminium le vérin 5 de magasin déplace un autre convoyeur jusqu'au l'activation du capteur Mag avacé 1B11 et puis le vérin des pinces ouvre et jette l'échantillon au magasin des matériaux ferreux, le vérin 3 de l'Axe Z se lève et le vérin 4 fait une rotation pour revenir à sa position initial, le cycle peut recommencer.

3.4 Description des entrées/sorties

a) Les entrées

INIT : Initialisation du grafctet.

Ret Dis, 1B1 : Capteur indique que le vérin de distribution des pièces est fermé.

Avance Dis, 1B2 : Capteur indique le vérin 1 ouvert.

1S1 : Capteur de proximité indique la présence d'échantillon au magasin.

Rotary mag, 1B3 : Capteur indique que le vérin 4 est orienté vers le magasin.

Rotary con, 1B4 : Capteur indique le vérin 4 est orienté vers le convoyeur.

X Ret, 1B5 : Capteur indique que le vérin 2 de l'axe X recule.

X Avancé, 1B6 : Capteur indique le vérin 2 de l'axe X avancé.

Z Up, 1B7 : Capteur indique que le vérin 3 de l'axe Z recule.

Z Down ,1B8 : Capteur indique le vérin 3 de l'axe Z avancé.

Magzin Cylinder, 1B10 : Capteur indique que le vérin 5 ferme.

Magazin Cylindre ,1B11 : Capteur indique le vérin 5 ouvert.

S3 : Détecteur des matériaux ferreux et non ferreux.

Gripper, 1B9 : Capteur de fermeture des pinces.

S2 : Fin de course.

b) Les sorties

Dis Cyl : Vérin 1.

X Axis : Vérin 2.

Z Axis : Vérin 3.

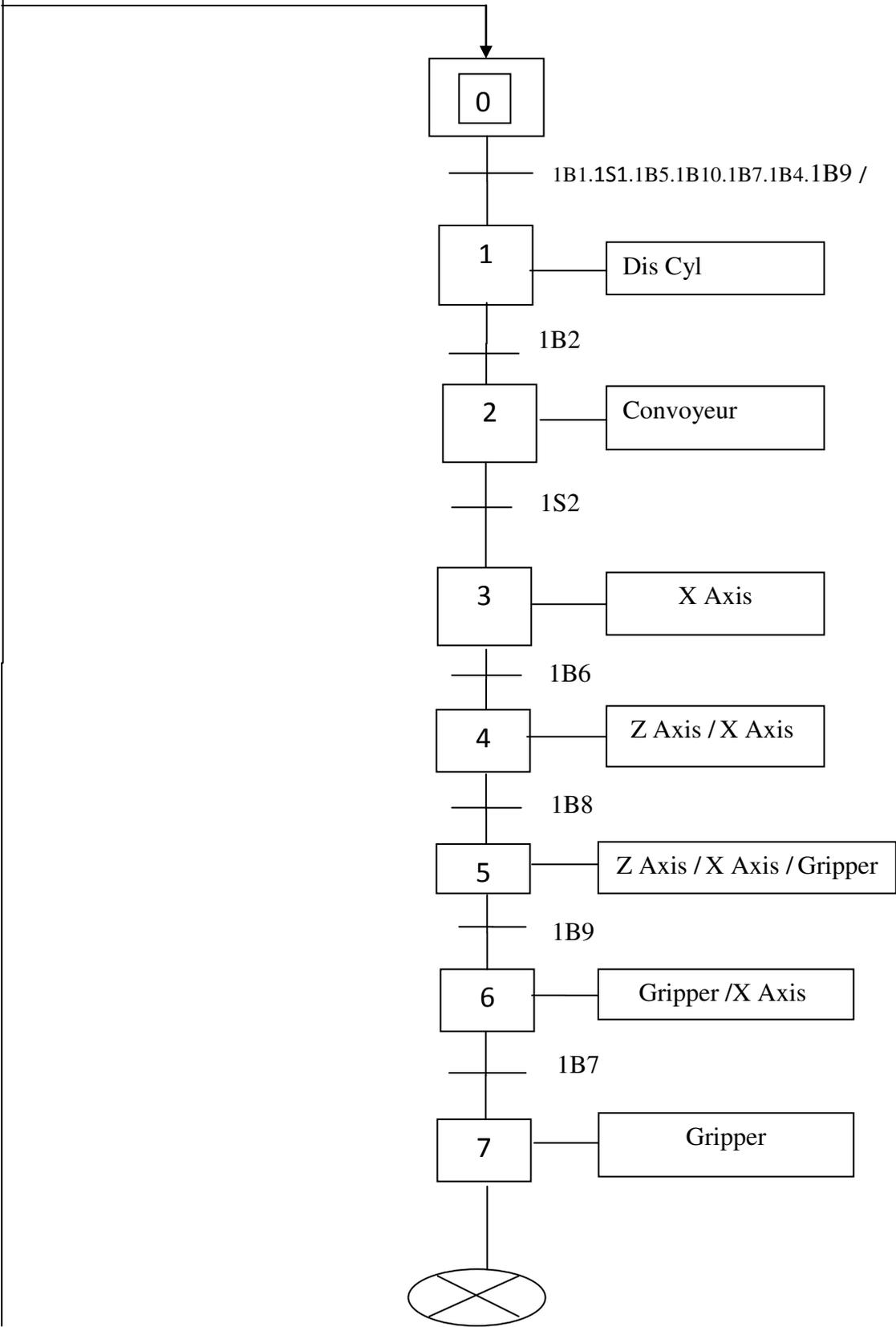
Gripper : Vérin de fermeture des pinces.

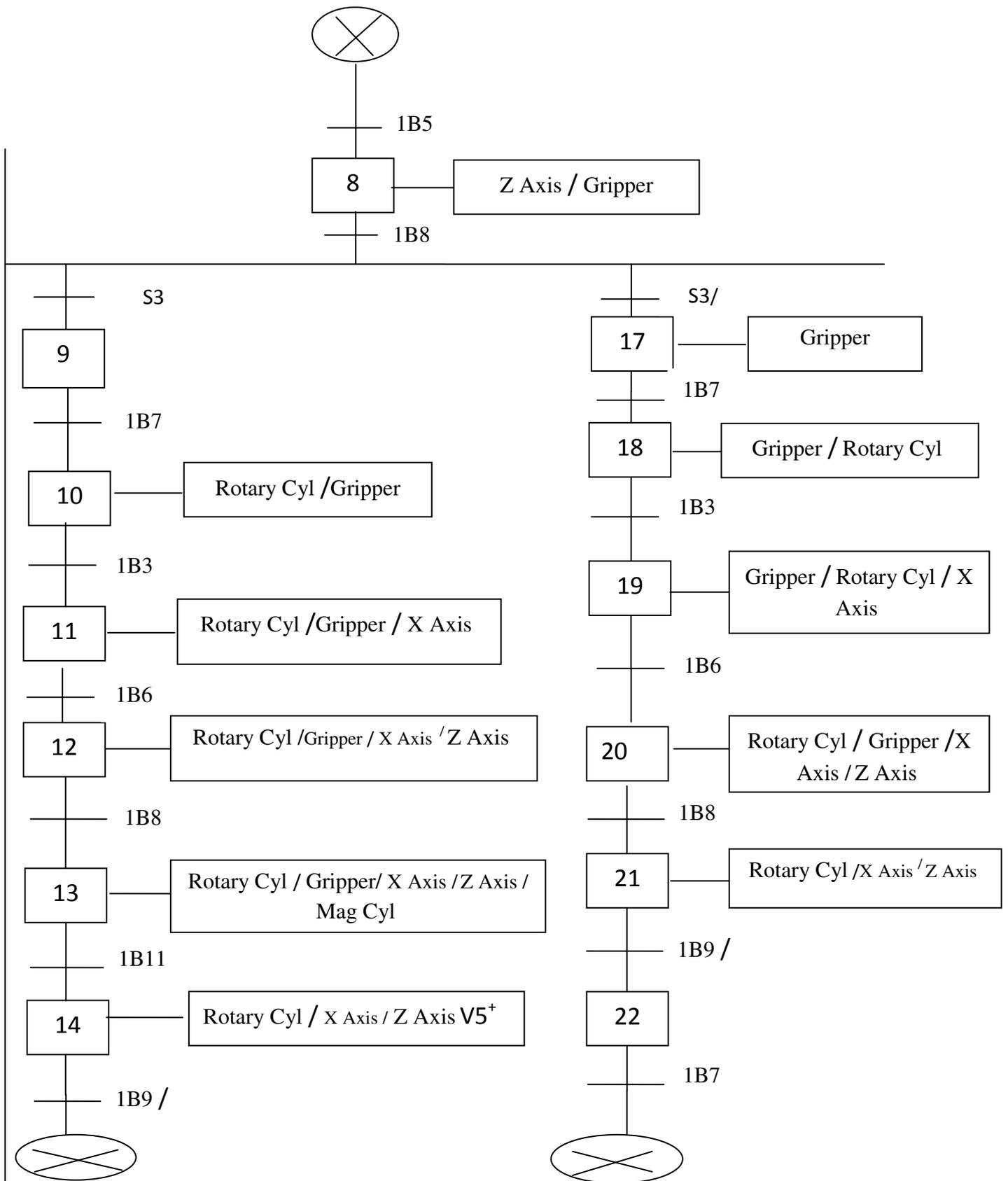
Mag Cyl : Vérin 5.

Rotary Cyl : Vérin 4 de rotation.

Convoyeur : Démarrage moteur convoyeur.

3.5 Description du cahier des charges par Grafset





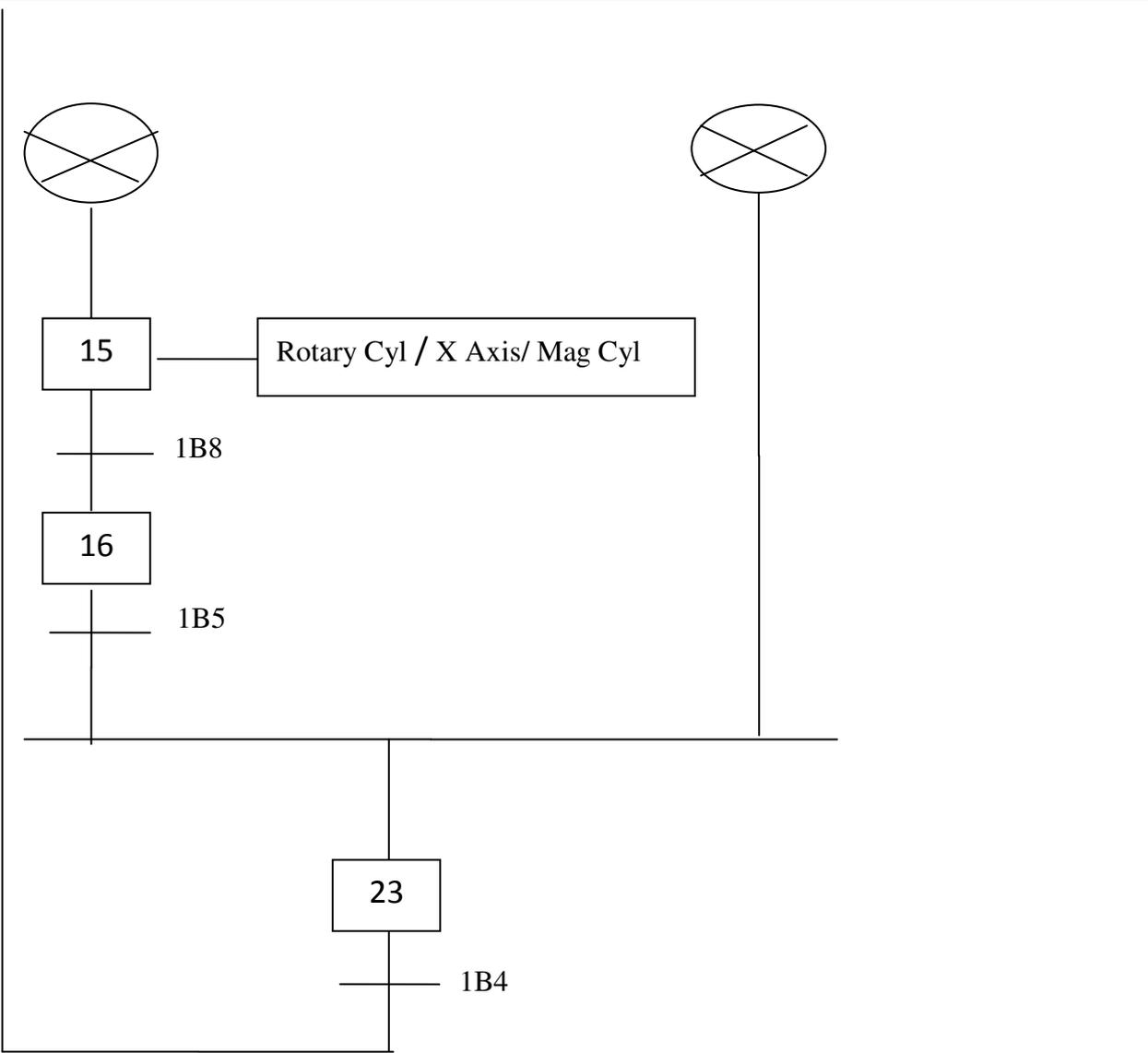


Figure 3.2 : Grafcet de fonctionnement du système PTP ROBOT PNEUMATIQUE TRAINER

3.6 Organigramme

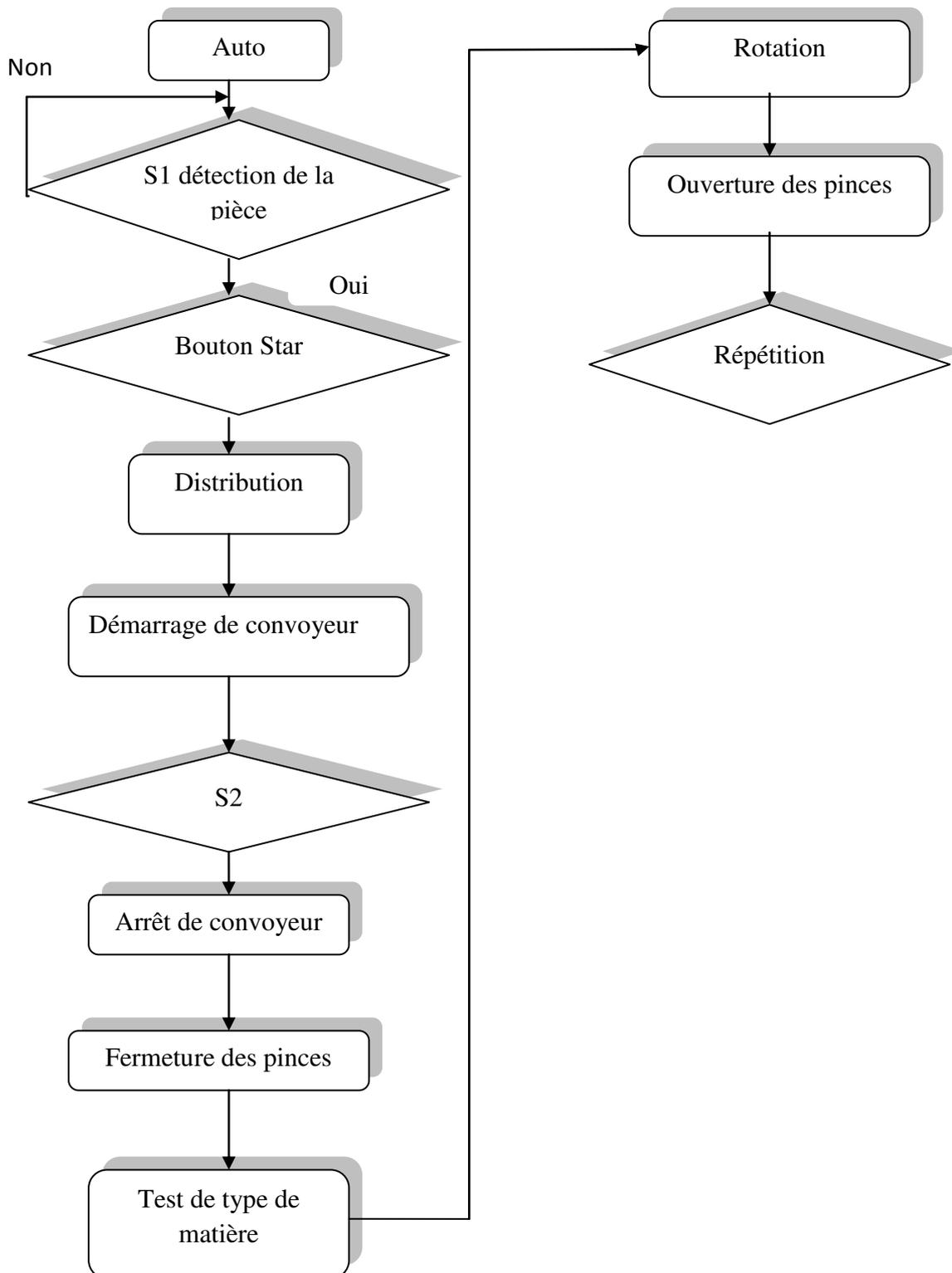


Figure 3.2 : Organigramme du système PTP ROBOT PNEUMATIQUE TRAINER [11].

3.7 Les adresses des entrées/sorties

a) Les entrées

Start Button	I2.0
Mag Ret, 1B1	I0.0
Mag adv, 1B2	I0.1
1S2	I1.4
1S1	I1.3
Rotary mag, 1B3	I0.3
Rotary con, 1B4	I0.2
X Ret, 1B5	I0.5
X adv, 1B6	I0.4
Z up, 1B7	I0.7
Z Down, 1B8	I0.6
Magazine retrieve, 1B10	I1.2
Mag Advanced, 1B11	I1.1
S3	I1.5
Gripper ,1B9	I1.0
Stop button	I2.1

Tableau 3.2 : Les adresses des entrées du système PTP ROBOT PNEUMATIQUE TRAINER

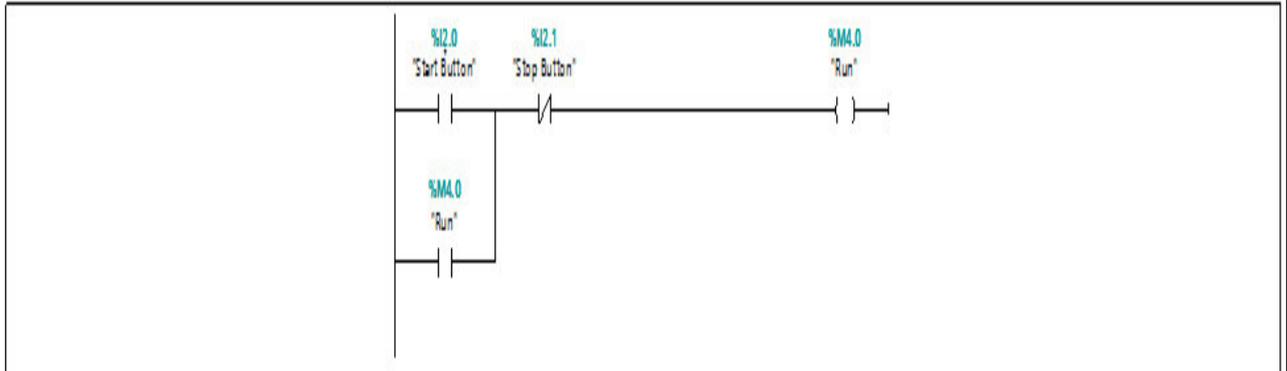
b) Les sorties

Q0.0	Dis Cyl
Q0.2	X Axis
Q0.3	Z Axis
Q0.1	Rotary Cyl
Q0.5	Mag Cyl
Q0.4	Gripper
Q0.6	Convoyeur

Tableau 3.3 : Les adresses des sorties du système PTP ROBOT PNEUMATIQUE TRAINER

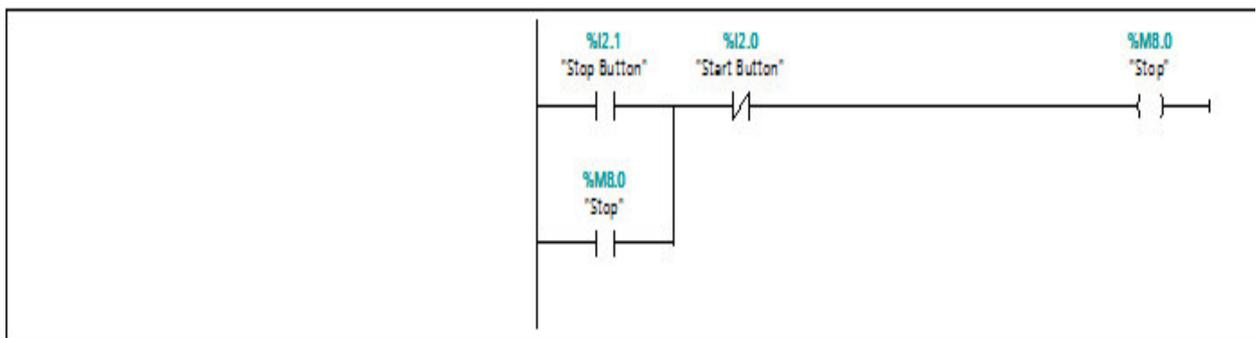
3.8 Programmation en langage a CONTACT (LADDER)

Réseau 1 :



Mnémonique	Adresse	Type	Commentaire
"Run"	%M4.0	Bool	
"Start Button"	%I2.0	Bool	

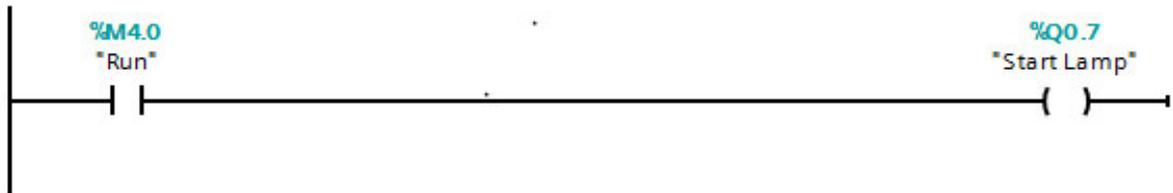
Réseau 2 :



Mnémonique	Adresse	Type	Commentaire
"Start Button"	%I2.0	Bool	
"Stop Button"	%I2.1	Bool	
"Stop"	%M8.0	Bool	

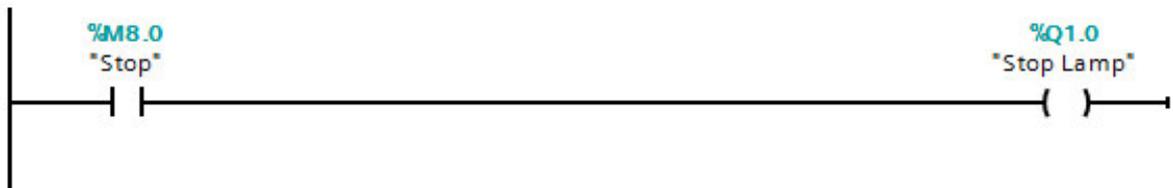
▼ Réseau 3 :

Commentaire



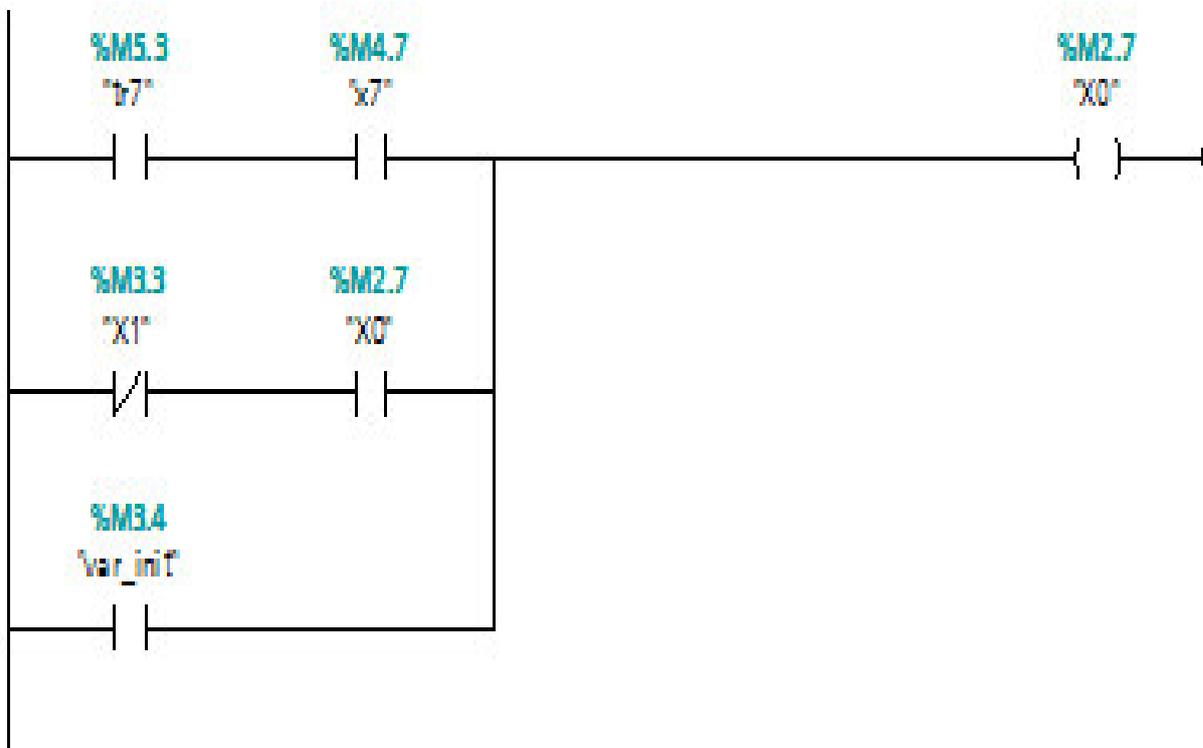
▼ Réseau 4 :

Commentaire



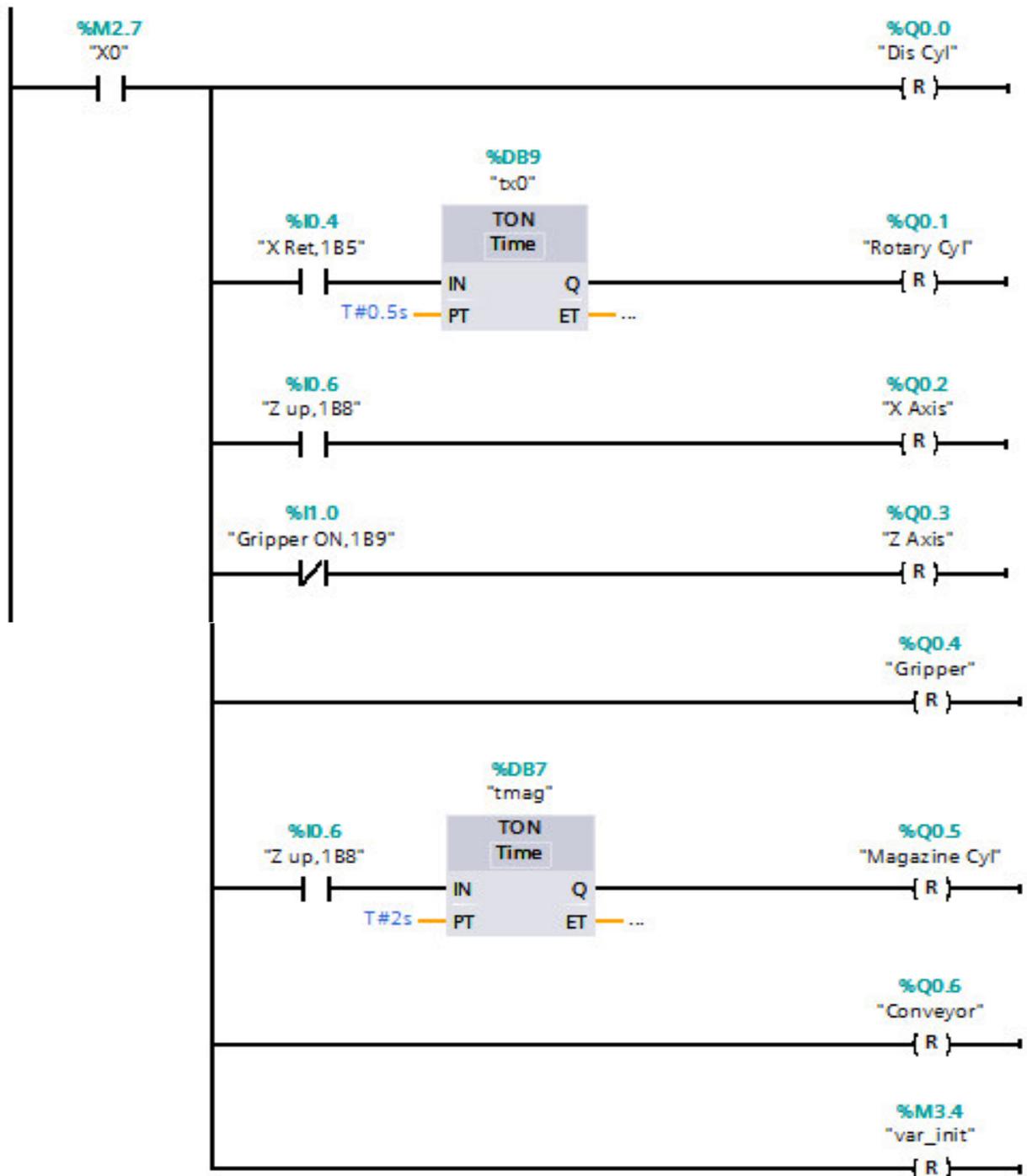
Réseau 5 :

Etape Initial



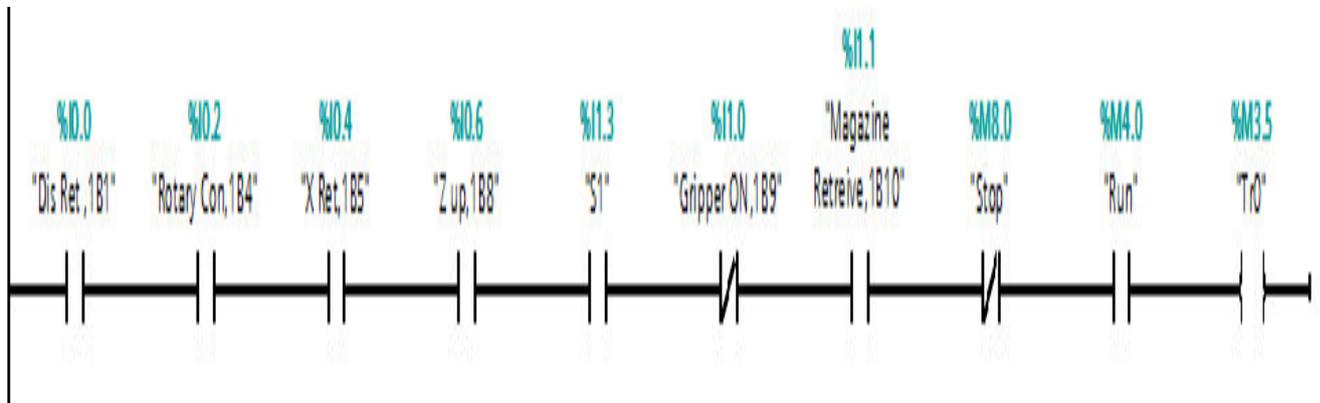
Réseau 6 :

Action Initial



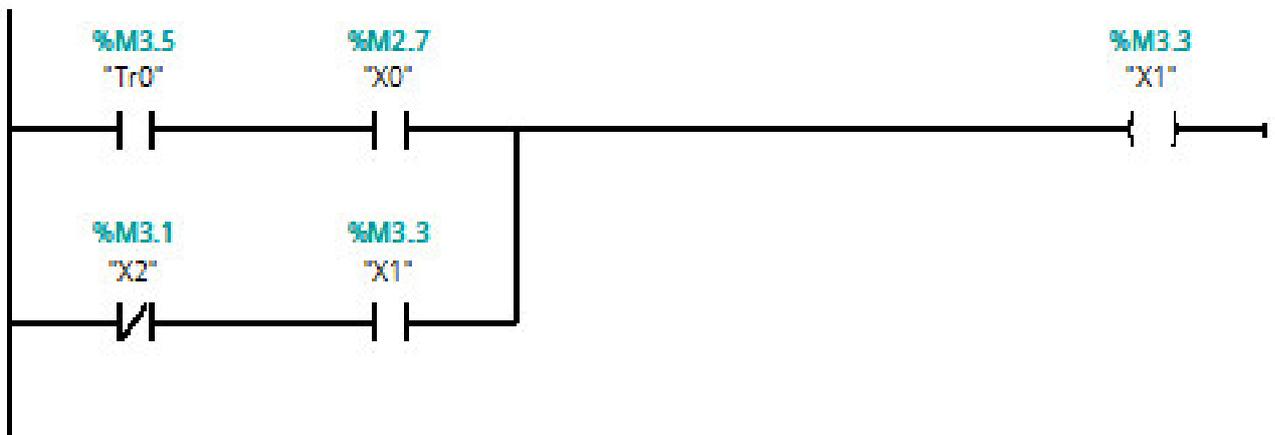
Réseau 7 :

Transition Initial



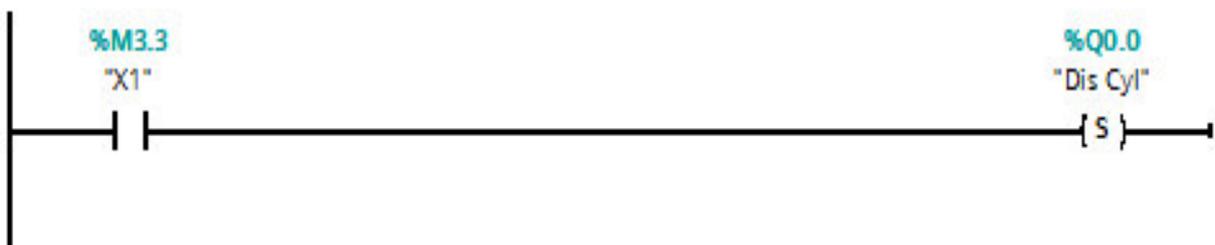
Réseau 8 :

Etape x1:Distribution des pièces



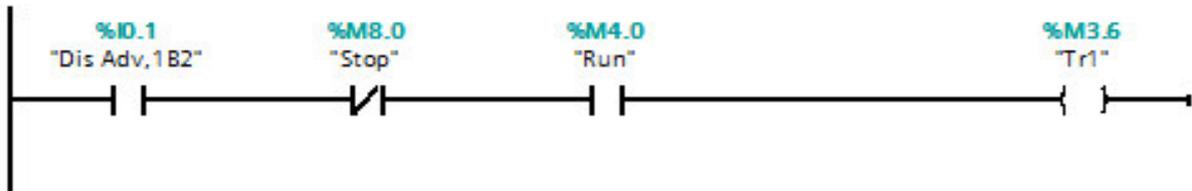
Réseau 9:

Action de distribution des pièces



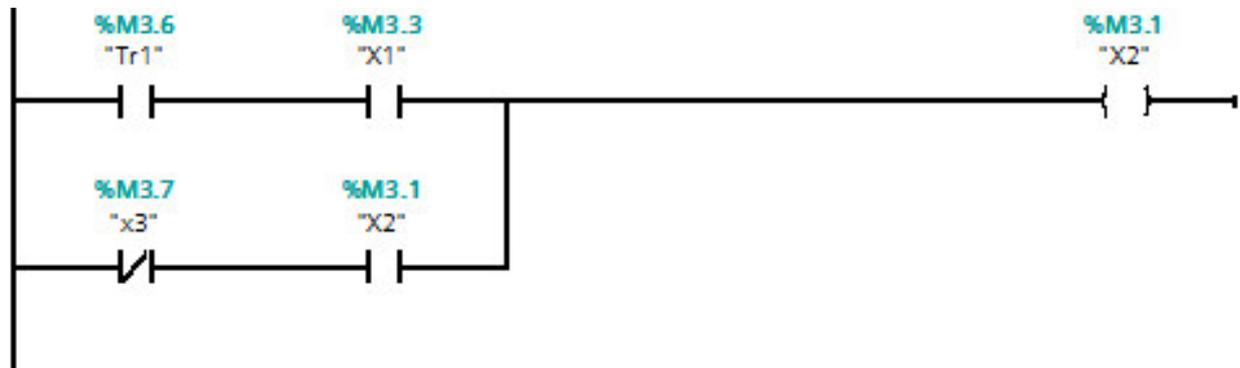
Réseau 10 :

Transition de distribution des pièces



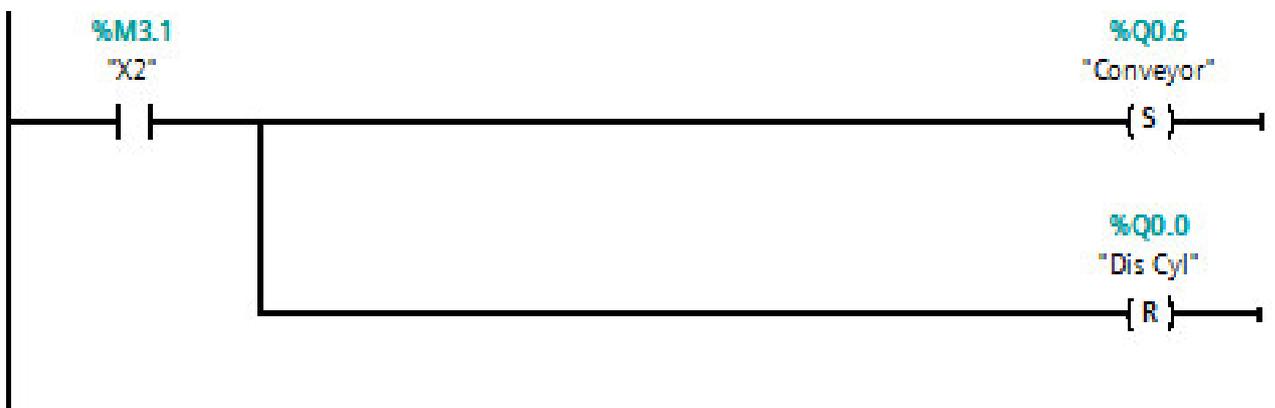
Réseau 11 :

Etape2 : Démarrage convoyeur



Réseau 12 :

Action 2 : de démarrage convoyeur



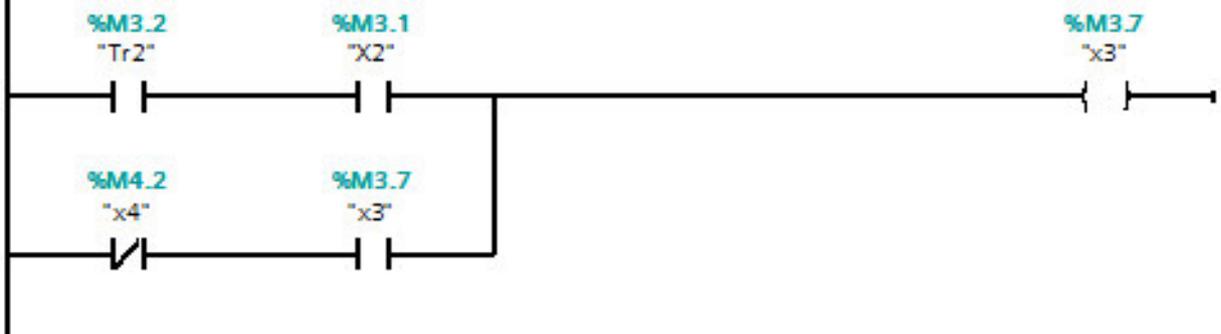
Réseau 13 :

Transition de démarrage convoyeur



Réseau 14 :

Etape3 : Avancement axe X, Descend axe Z



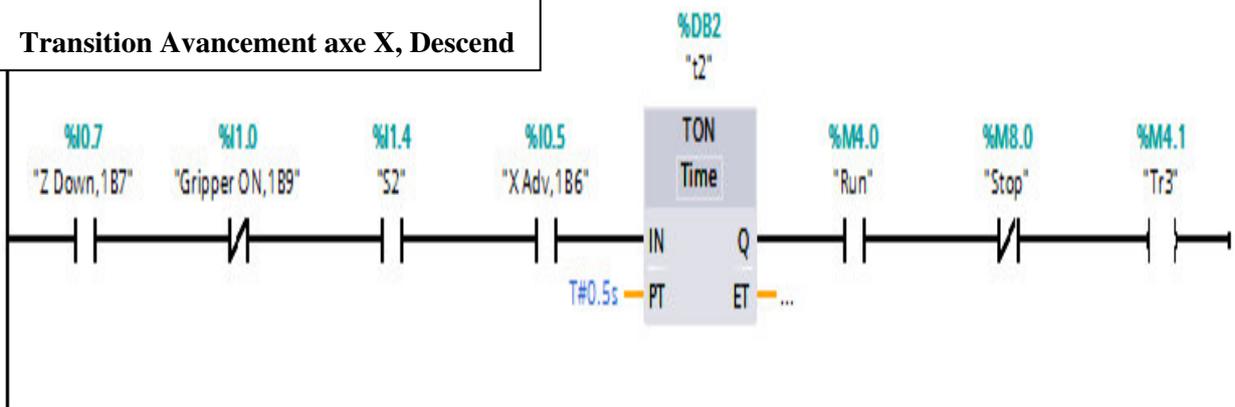
Réseau 15 :

Action 3 : Avancement axe X, Descend axe Z



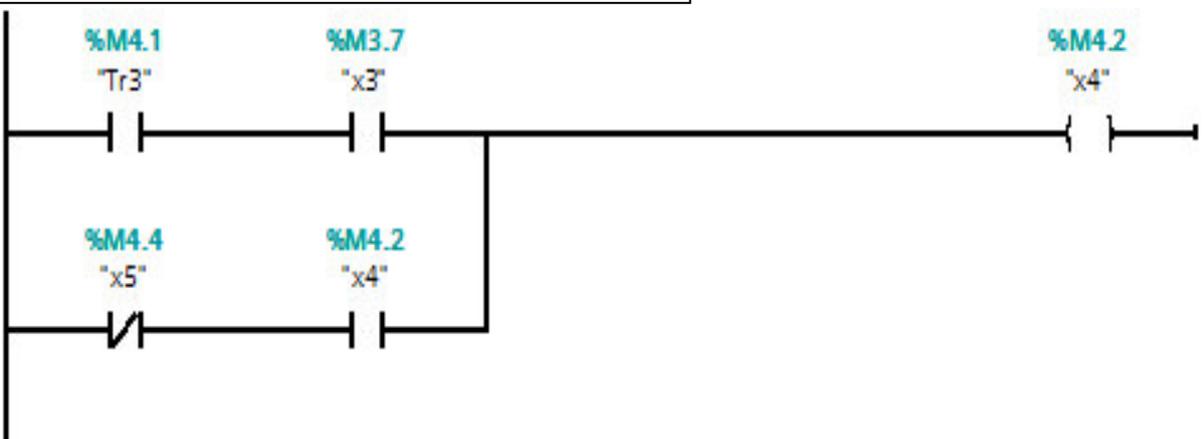
Réseau 16 :

Transition Avancement axe X, Descend



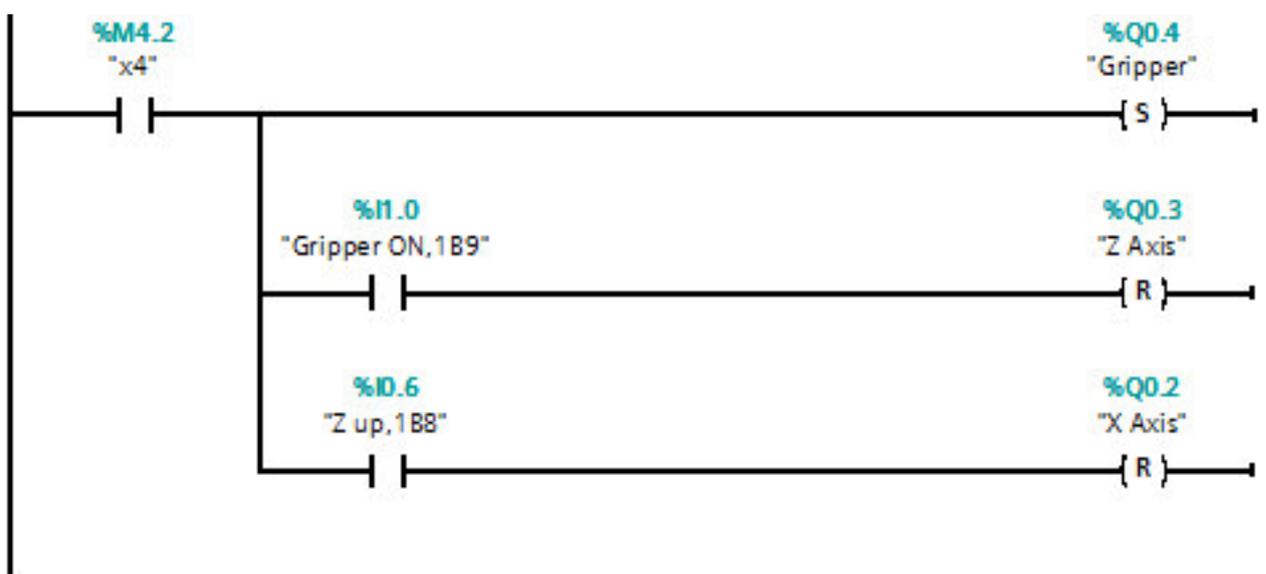
Réseau 17 :

Etape4 : fermeture pince, recul axe Z, recul axe X



Réseau 18 :

Action Fermeture pinces+recul Z+recul X



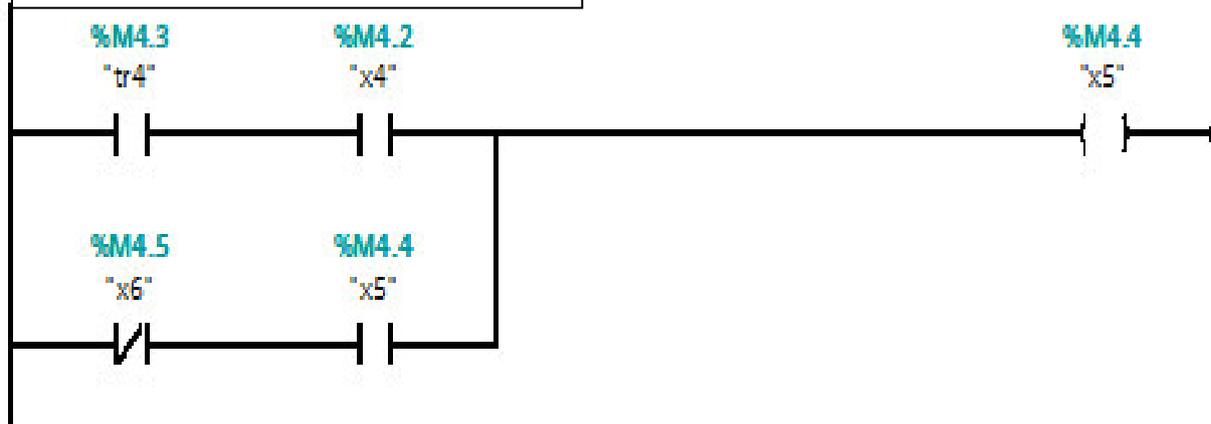
Réseau 19 :

Transition fermeture pince, recul axe Z, recul axe X



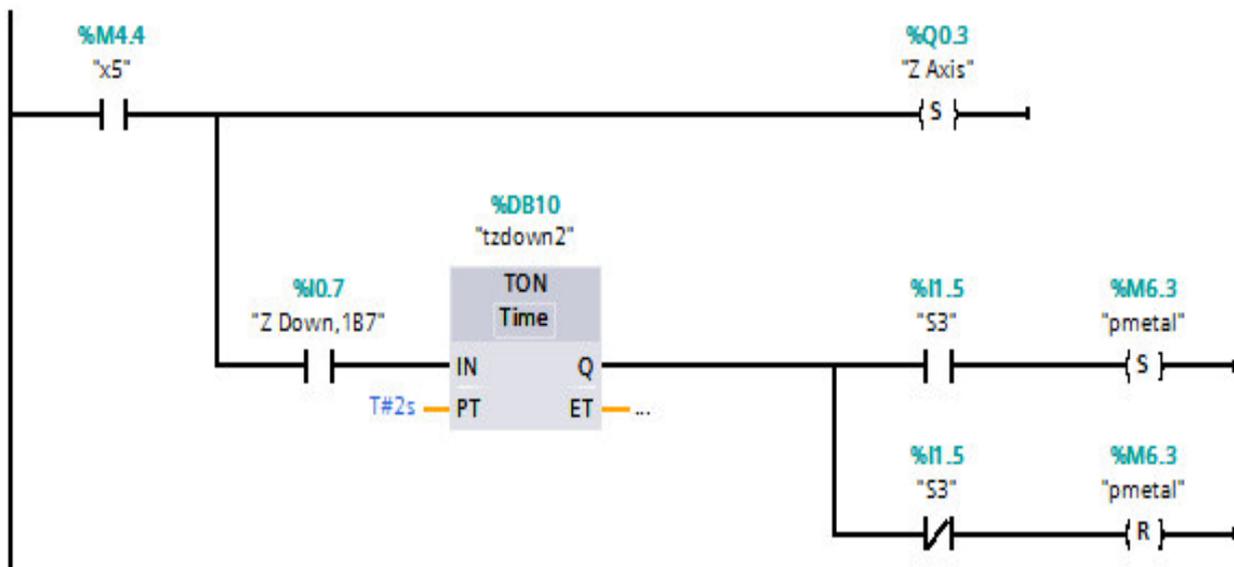
Réseau 20 :

Etape5 : Descend axe Z, test type matière



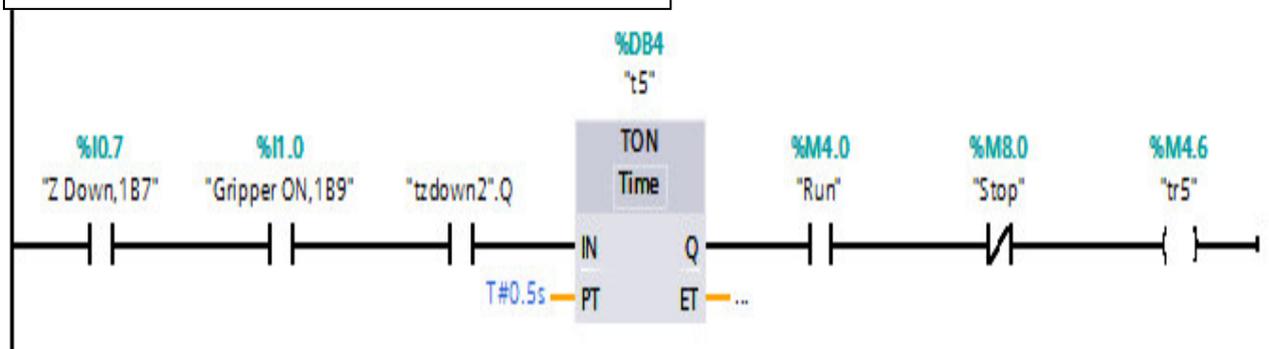
Réseau 21 :

Action 5 : Descend axe Z, test type matière



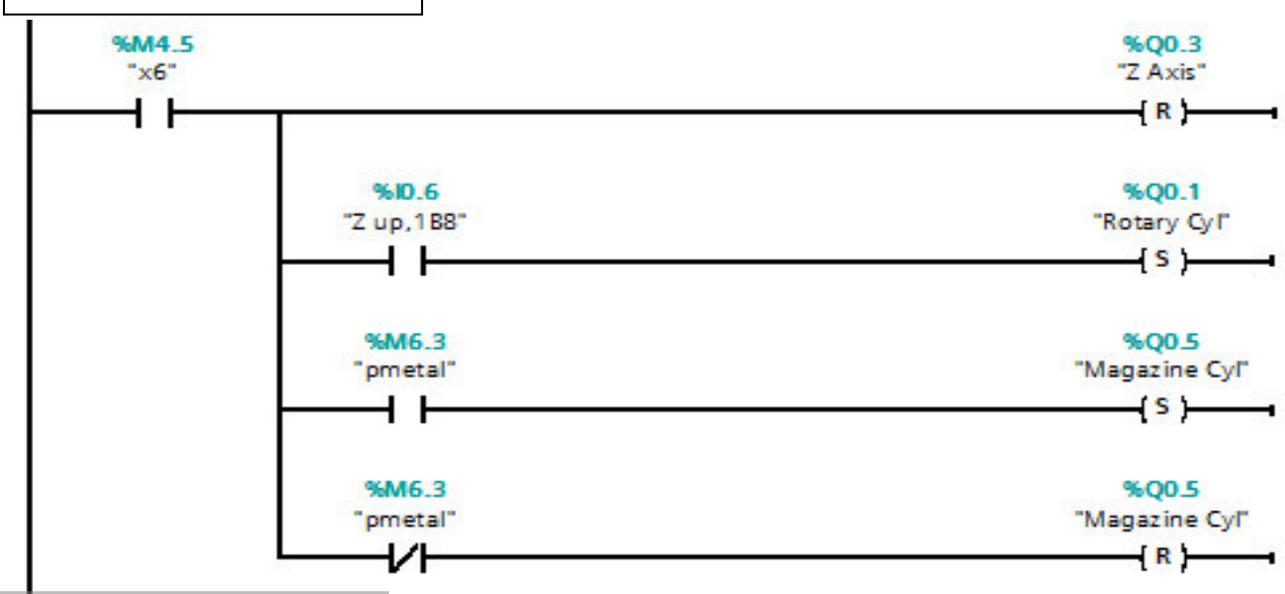
Réseau 22 :

Transition22 : Descend axe Z, Test type Matière



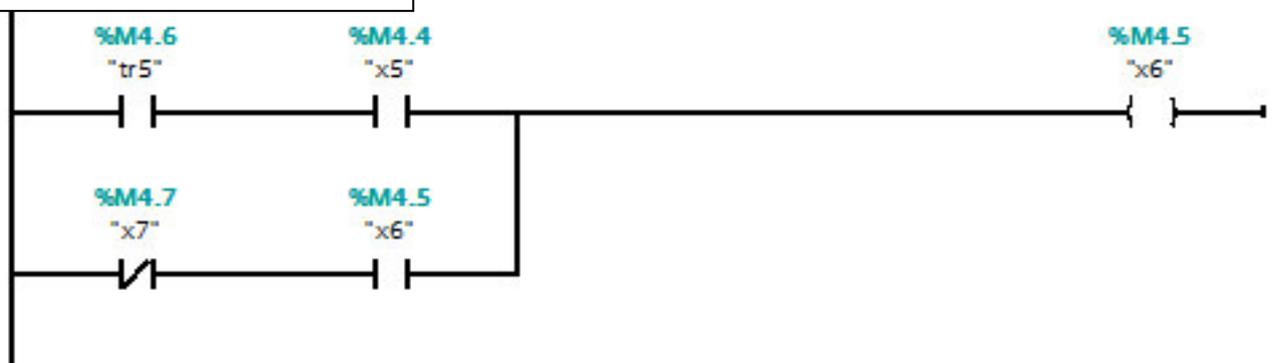
Réseau 24 :

Action 6 : axe Z UP, Rotation



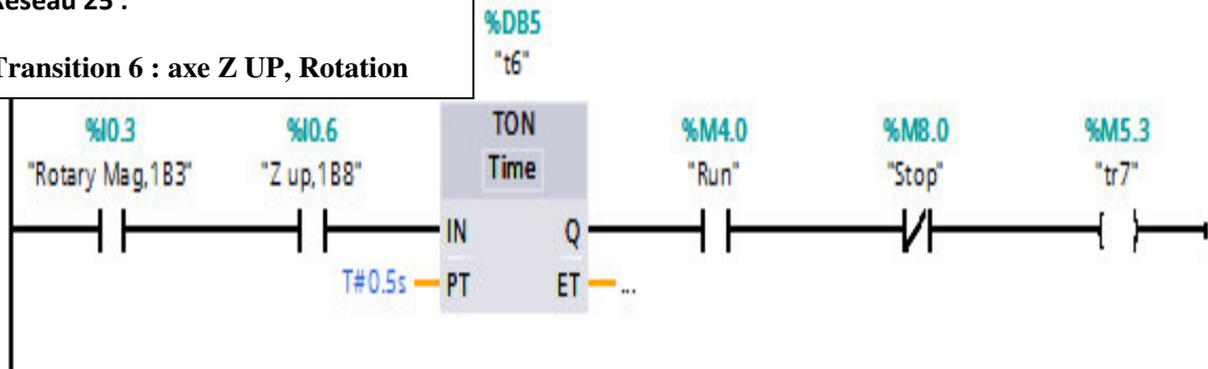
Réseau 23 :

Etape6: axe Z UP, Rotation



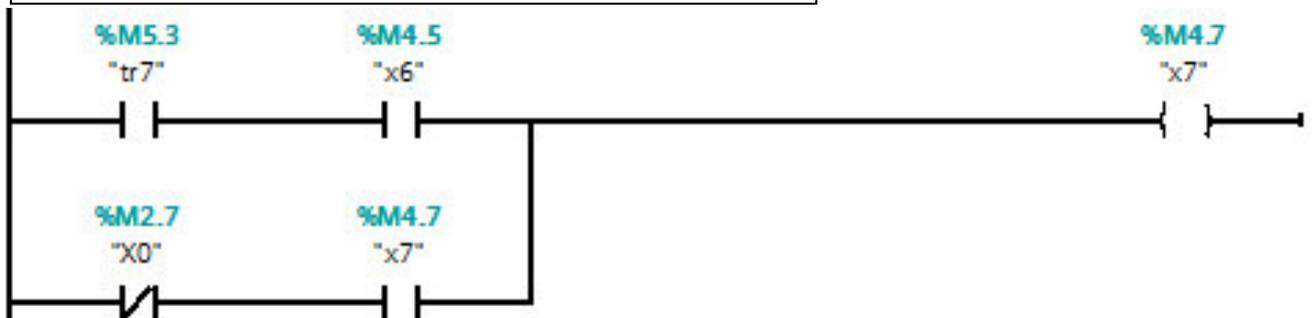
Réseau 25 :

Transition 6 : axe Z UP, Rotation



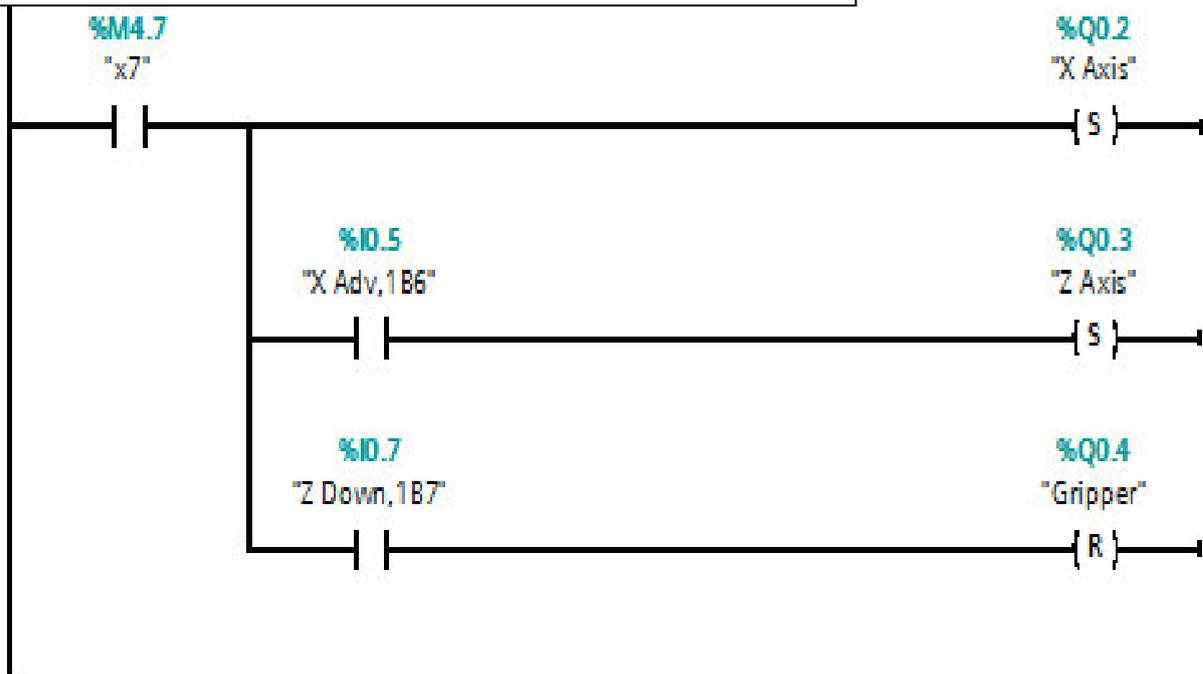
Réseau 26 :

Etape7: Avancement axe X, Descend axe Z, Ouverture pince



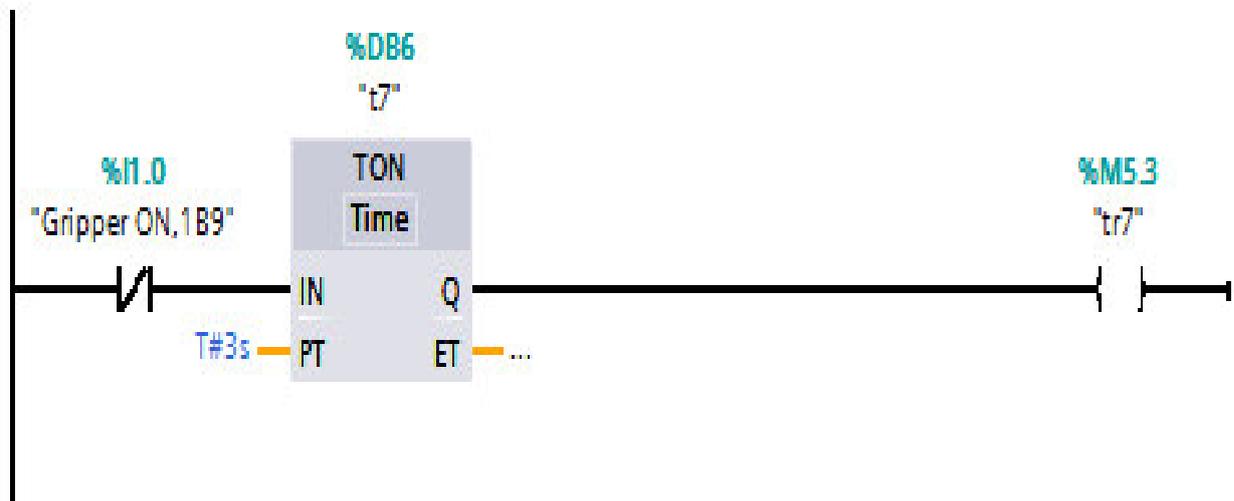
Réseau 27 :

Action 7 : Avancement axe X, Descend axe Z, Ouverture pince



Réseau 28 :

Transition22 : Avancement axe X, Descend axe Z, Ouverture pince



3.9. Conclusion :

Ce travail nous a permis d'explorer le côté pratique de l'automatique et de voir les difficultés réelles au sein du milieu industriel.

Conclusion générale

Ce travail est consacré à la commande d'un système didactique (PTP ROBOT PNEUMATIQUE TRAINER) par un automate programmable industriel siemens (SIMATIC S7-1200) disponible au niveau de notre laboratoire d'automatique. En utilisant le logiciel de programmation TIA PORTAL V14.

Dans notre projet nous avons réussi à écrire le cahier des charges du système PTP ROBOT PNEUMATIQUE TRAINER, la description de cahier des charges par grafcet et de faire la programmation de ce système en langage à contact. Le programme est vérifié par le simulateur PLCSIM et ensuite validé par la commande du système PTP ROBOT PNEUMATIQUE TRAINER.

Ce travail nous a permis de comprendre les différentes étapes suivies pour l'automatisation des systèmes et maîtriser le logiciel de programmation TIA PORTAL des automates de la gamme S7-1200.

A la fin nous souhaitons que ce travail, concernant l'automate SIMATIC S7-1200 soit un exemple aux futurs étudiants, qui vont sans doute mettre en œuvre d'autres projets plus développé.

BIBLIOGRAPHIE

- [1]: auto.polytech.univ-tours.fr/automatique/.../Presentation%20Automatisme%202007-08...
- [2] : Robot manipulateur. Relais RH ET Automate programmable. Volume A.
- [3] : Henri Ney « Électronique et normalisation ,2.éléments d'automatismes »Fernande Nathan1985.
- [4] :[http://pagesperso-orange.fr/Jacque.boubier/C-online/Informatique-industrielle/Automate cours%2008.htm](http://pagesperso-orange.fr/Jacque.boubier/C-online/Informatique-industrielle/Automate%20cours%2008.htm) .
- [5]:<http://www.courselec.free.fr>.
- [6] : M.GRISLIN « Travaux pratique Automate programmables industriels 1^{ère} série exercice Grafcet ».
- [7] : www.uvt.rnu.tn/resources-uvt/cours/Automatismes/chapitre2_api.pdf Chapitre 2 LES AUTOMATES PROGRAMMABLES INDUSTRIELS (API)
- [8] : mémoire fin d'étude Etude d'un système automatisé avec la réalisation et la commande d'un bras pneumatique
- [09]: https://support.industry.siemens.com/cs/attachments/.../s71200_system_manual_fr-FR.
- [10]:[https://support.industry.siemens.com/cs/attachments/.../s71200_system_manual_en_US en US](https://support.industry.siemens.com/cs/attachments/.../s71200_system_manual_en_US_en_US).
- [11]:manual PTP Pneumatic robot trainer (2017.04.21).