



Université Mohamed Khider de Biskra
Faculté des Sciences et de la Technologie
Département de génie électrique

MÉMOIRE DE MASTER

Sciences et Technologies
Electrotechnique
Commande Electrique

Réf. : Entrez la référence du document

Présenté et soutenu par :
HOUAHOUA Nouredine

Le : samedi 23 juin 2018

Commande par un automate d'une chaîne de transfert, triage et déplacement des pièces par un robot pneumatique

Jury :

Dr. BATKA Achour	Pr	Université de Biskra	Président
Dr. KRAA Okba	MCB	Université de Biskra	Rapporteur
Dr. SAADOUNE Achour	MCA	Université de Biskra	Examineur

Résumé :

Notre projet de fin d'étude de cycle master commande de machine consiste d'établir un cahier de charge pour une chaîne de distribution, transfert et triage de pièce. Puis la modélisation graphique de cette chaîne en utilisant le langage GRAFCET afin de nous permettre de contrôler ce système par l'automate programmable S7-1200.

À la fin de cette réalisation, on a ajouté utilise la technique de supervision de type PC du système avec l'application de SIMATIC wincc pour nous permettre de la télésurveillance, le télédiagnostic et la télémaintenance à longue distance.

الملخص

الهدف من هذا الانجاز هو وضع وكتابة دفتر شروط لسلسلة فاعلة مهمتها توزيع و تحويل وفرز القطع ثم إنشاء التمثيل التخطيطي لهذه السلسلة باستعمال لغة الوصف التخطيطي حتى نتمكن من التحكم في هذا النظام باستعمال آلية البرمجة س7 1200.

كما قمنا في نهاية هذا المشروع بإضافة تقنية المتابعة و التحكم عن بعد لتسهيل عملية الصيانة وتحديد الاعطاب لهذه السلسلة.

Introduction générale

Un automatisme est un sous-ensemble d'une machine, destinée à remplacer l'action de l'être humain dans des tâches en générales simples et répétitives, réclamant précision et rigueur. On est passé d'un système dit manuel, à un système mécanisé, puis au système automatisé. Dans l'industrie, les automatismes sont devenus indispensables : ils permettent d'effectuer quotidiennement les tâches les plus ingrates, répétitives et, dangereuses. Parfois, ces automatismes sont d'une telle rapidité et d'une telle précision, qu'ils réalisent des actions impossibles pour un être humain. L'automatisme est donc synonyme de productivité et de sécurité.

L'automate programmable API est au cœur de toute industrie automatisée, c'est un équipement spécialement conçu pour l'industrie et sa mission principale est de piloter des chaînes de montage, production, maintenance, robots industriels, machines à outils.

Des grandes compagnies industrielles au monde sont installées à nos pays, tel que l'industrie de l'automobile, l'industrie alimentaire et surtout la production des matériaux de construction comme les briqueteries et les cimenteries. La céramique tout son projet nous offre une occasion d'étudier et de maîtriser l'automatisme industriel à fin de trouver les solutions aux problèmes qui interviennent lors du fonctionnement.

Le mémoire est constitué de trois chapitres organisés de façon séquentielle. Le premier chapitre est une introduction aux systèmes automatisés, leur description, objectifs et comment un tel système d'automatisme industriel est structuré.

Dans le deuxième chapitre nous décrivons l'automate programmable industrielle API sa structure interne ; et sa communication avec son entourage par les différentes interfaces. Précisément on a choisi le SIMATIC S7-1200 de SIEMENS, qui utilise le logiciel Tia portal comme outil de programmation.

Le troisième chapitre est une étude d'un système automatisé qui ressemble à une chaîne de transfert, triage et déplacement des pièces par le Robot Trainer. Un cahier de charge a été élaboré pour nous aider à développer le programme final de la chaîne.

Sommaire

Introduction générale.....	3
CHAPITRE I : Introduction aux systèmes automatisés	
I.1 Introduction.....	4
I.2 Description d'un système automatisé.....	4
I.2.1 Définition d'un système automatisé.....	4
I.2.2 Objectif de l'automatisation.....	6
I.3 Structure des systèmes automatisés.....	6
I.3.1 Partie opérative (PO).....	
I.3.1.1 Actionneurs.....	
I.3.1.1. a. Actionneurs pneumatiques (les vérins).....	
I.3.1.1. b. Actionneur électrique.....	
I.3.1.2. Pré actionneurs.....	
I.3.1.2. a. Pré actionneurs pneumatiques.....	
I.3.1.2. b. Pré actionneurs électriques.....	
I.3.2 Partie commande (PC).....	
I.3.2.1 Les Capteurs.....	
I.3.2.2 Types de capteur.....	
I.3.2.3 Les effecteurs.....	
I.3.3 Dialogue homme / machine.....	
I.4 Chaîne fonctionnelle.....	
I.4.1 Définition.....	
I.4.2 La chaîne d'information.....	
I.4.3 La chaîne d'énergie.....	
I.5 Conclusion.....	
CHAPITRE II Automates programmables industrielles (API)	
II.1 Introduction.....	
II.2 Définition d'un automate.....	
II.3 Structure générale des API	
II.3.1 Caractéristiques d'un automate.....	
II.3.2 Architecteur interne d'un l'API.....	

II.3.3	Cycle de fonctionnement d'un automate.....	:
II.3.4	Cartes d'entrées tout ou rien (TOR).....	
II.3.5	Cartes de sorties TOR.....	
II.3.6	Cartes d'entres sorties analogiques.....	
II.3.7	Choix d'un API.....	
II.4.	Langages de programmation des API.....	
II.4.1	Les composants du langage.....	
II.5	Langage SFC GRAFCET.....	
II.5.1	Règles du GRAFCET.....	
II.5.2	Composants de base (symboles graphiques) du GRAFCET.....	
II.5.3	Types d'action.....	
II.5.4	Avantages du GRAFCET.....	
II.5.5	Étapes et actions associée.....	
II.5.6	Gestion des modes Marche/ Arrêt ET Arrêts d'urgences.....	
II.5.7	Mise en équation d'un GRAFCET.....	
II.6	Développement et téléchargement du programme.....	
II.6.1	La programmation en Tia portal.....	
II.7	Conclusion.....	
CHAPITRE III modélisation et contrôle du système à étudier		
III.1	Description du système.....	
III.1.1	Le système de contrôle.....	
III.1.2	L' outil de programmation.....	
III.2	Cahier de charge	
III.2.1	Description du cycle de fonctionnement.....	
III.2.3	Mise en équations du GRAFCET	
III.2.2	Etablissement du GRAFCET).....	
III.3	Réalisation du programme (phase de programmation).....	
III.4	Le Programme	
III.5	La supervision du processus avec SIMATIC Win CC.....	
III.6	Câblage de l'automate	
III.7	Conclusion	

Ce chapitre permet de se familiariser avec la structure matérielle et logicielle des automates programmables. Particulièrement SIEMENS ; et l'analyse des cahiers de charges.

Afin de produire des solutions programmées à des problèmes d'automatismes séquentiels simples utilisant des capteurs et des actionneurs discrets.

À l'issue de ce chapitre, on a réussi de programmer un automate de la marque SIEMENS S7 1200 dans un contexte industriel.

Les objectifs sont atteints à des étapes très avancés. Ils sont comme suit :

- déterminer les différentes possibilités et limites d'un automate programmable Industriel ;
- utiliser une méthode graphique de conception ;
- établir la communication et à configurer un automate programmable ;
- réaliser les branchements d'éléments de commandes sur les cartes d'entrées et de sorties.

Les principaux thèmes abordés sont : les caractéristiques et les composants physiques d'un automate ; l'utilisation d'un logiciel de programmation ; la programmation d'instructions de base des langages LD (Ladder) ainsi que SFC

(Sequential Function Chart) ; le mode de représentation et d'analyse GRAFCET.

I.1 Introduction

Les systèmes automatisés sont partout dans notre environnement quotidien. L'automatique c'est l'ensemble des sciences et des techniques utilisées dans la conception et la réalisation des systèmes automatisés simples ou complexes,

Dans ce chapitre, en présentant une description bien détaillée sur l'automatisme industriel et les systèmes automatisés, l'architecture de base de l'automatisme industriel ses parties et leurs composants sont montrés aussi.

I.2 Description d'un système automatisé

I.2.1 Définition d'un système automatisé

Un système est dit automatisé s'il exécute toujours le même cycle de travail après avoir reçu les consignes d'un operateur, il est composé de plusieurs éléments de contrôle et de commande organise et interconnecté qui assurent un fonctionnement reproductible plus ou moins indépendant des interventions humaines [1]. La figure (I.1) montre un exemple très utilisé dans l'industrie



Figure I.1 exemple d'un système automatisé [1]

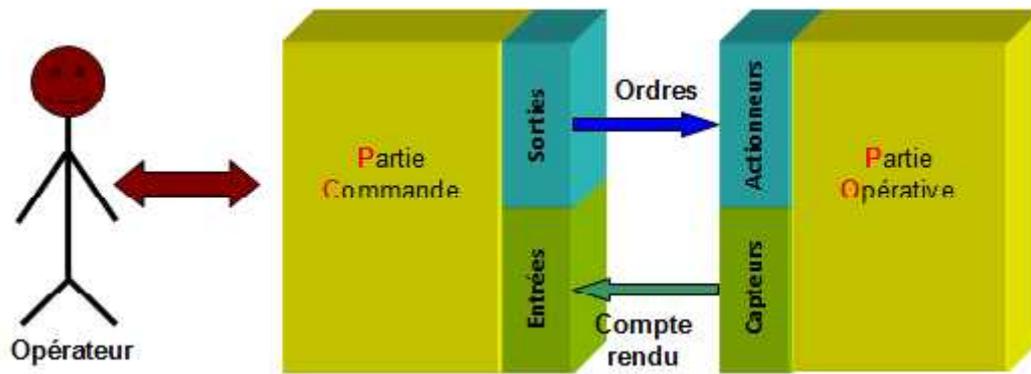


Figure I.2 Organisation d'un système automatisé

Les informations circulent d'une partie à l'autre par l'intermédiaire de dispositifs appelés interface. Ce sont parfois de simples cordons de liaison, ou un petit boîtier extérieur aux parties. Ce boîtier permet :

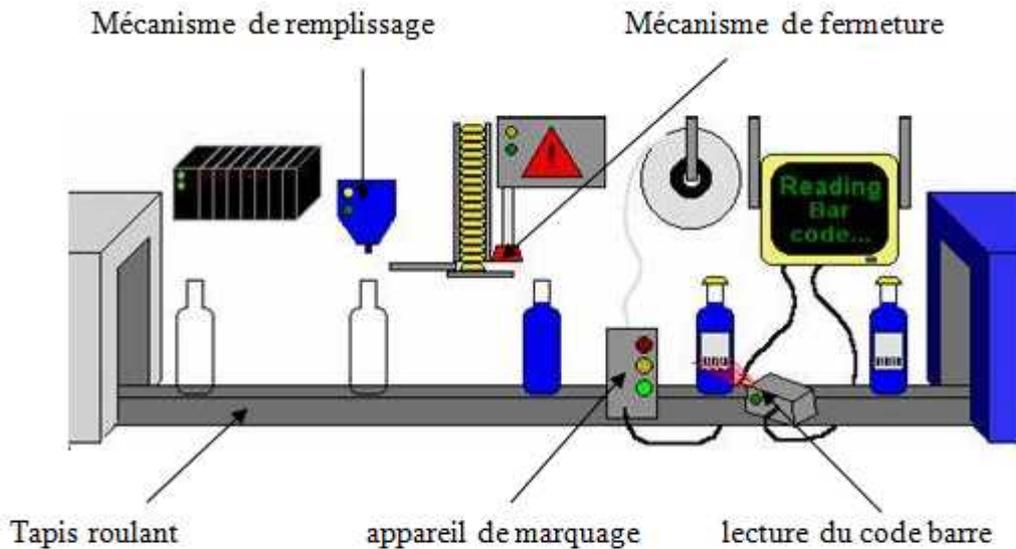
- de transformer les informations provenant des capteurs en informations reconnues par la partie de commande
- de fournir des informations qui permettent à la partie opérative la commande de ses actionneurs

Exemple : le schéma de la figure (I.3) représente une chaîne automatisée de remplissage et de fermeture de bouteilles.

I.2.2 Objectif de l'automatisation

Le but principale est de remplacer l'intervention humaine dans la plus part des tâches qui peuvent être rapides, très précises, dangereuses, répétitives et pénibles.

- L'automatisation aide à la compétitivité du produit.
- Améliorer la productivité en réduisant les coûts de production.
- Améliorer la qualité de production.
- Augmenter la production.
- Améliorer la flexibilité du système de production.
- Améliorer la qualité et la disponibilité des produits.



Figure(I.3) Image d'une chaîne de production automatisée

I.3 Structure des systèmes automatisés

Un système automatisé est décomposé en deux parties essentielles :

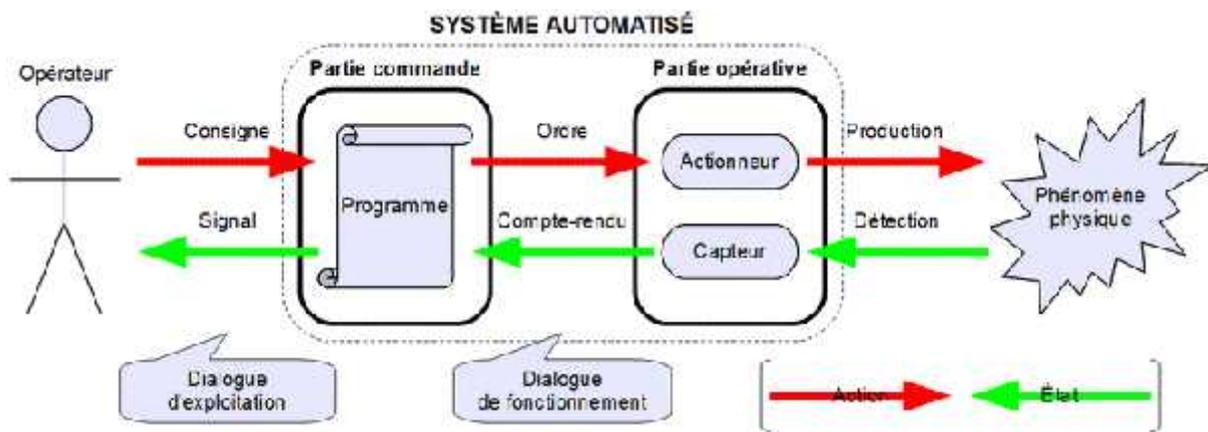


Figure I.4 Structure d'un système automatisé

I.3.1 Partie opérative (PO):

C'est la partie d'un système automatisé qui effectue ou exécute le travail. Autrement dit, c'est la machine. C'est la partie qui reçoit les ordres de la partie commande et qui les exécute.

Elle comporte les actionneurs et les pré actionneurs.

I.3.1.1 Actionneurs

Un actionneur est un élément qui transforme l'énergie qui lui est fournie en un phénomène physique qui fournit un travail, modifie le comportement ou l'état d'un système. Dans la théorie de l'automatisme, l'actionneur appartient à la partie opérative d'un système automatisé.

Energie électrique
ou pneumatique



Energie
Mécanique



Figure I.5 principe d'un actionneur

On distingue deux types d'actionneurs

I.3.1.1. a. Actionneurs pneumatiques (les vérins)

Un vérin pneumatique est un actionneur qui permet de transformer l'énergie de l'air comprimé en un travail mécanique.

Un vérin pneumatique est soumis à des pressions d'air comprimé qui permettent d'obtenir des mouvements dans un sens puis dans l'autre. Les mouvements obtenus peuvent être linéaires ou rotatifs [2].



Figure I.6 vérin pneumatique double effet

Il existe plusieurs types de vérins, les plus utilisés sont :

- Les vérins simple effet

Ce sont des vérins qui effectuent un travail dans un seul sens. Ils permettent soit de pousser soit de tirer une charge, exclusivement. Seules les positions extrêmes sont utilisées avec ce type de vérin. Un vérin pneumatique à simple effet n'a qu'une seule entrée d'air sous pression et ne développe un effort que dans une seule direction. La course de retour à vide est réalisée par la détente d'un ressort de rappel incorporé dans le corps du vérin.

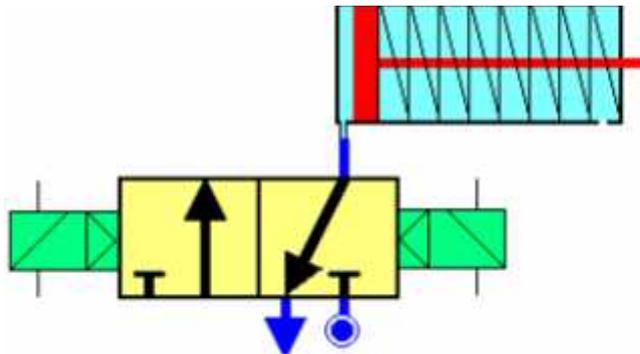


Figure I.7 vérin simple effet avec distributeur [03]

- Les vérins doubles effet

Le vérin double effet à deux alimentations possibles: soit par la chambre arrière, soit par la chambre avant.

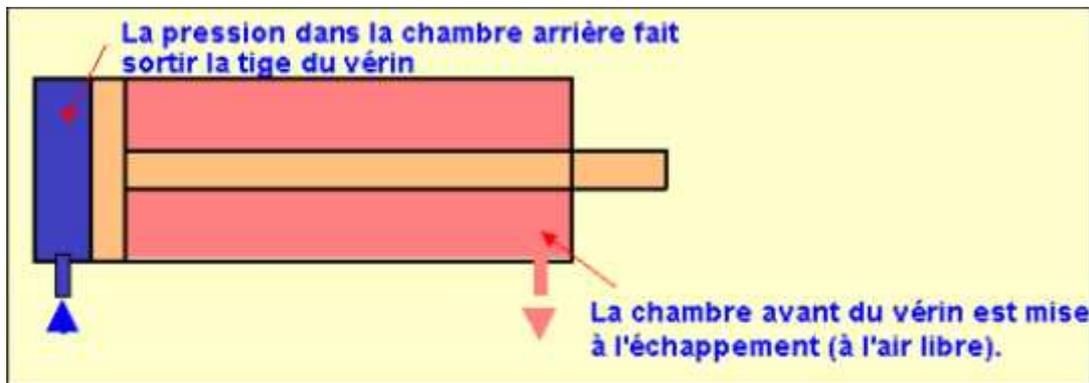


Figure I.8 vérin double effet [03]

- Les vérins spéciaux

Les fabricants proposent une grande variété de vérins spéciaux : vérins rotatifs, vibreurs, vérins sans tige, vérins à tige creuse, micro-vérin, vérins à faible course, vérins anti-rotation ...

- Quelques vérins spéciaux



Figure I.9 vérins spéciaux

I.3.1.1. b. Actionneur électrique

Selon la conversion de l'énergie électrique, on distingue des différents types d'actionneurs électriques.

Les moteurs électriques sont les actionneurs récurrents utilisés dans le domaine d'industrie automatisé, il se distingue en deux types, Moteurs à courant continu et Moteurs à courant alternatif [4].

- Les moteurs à courant continu

Ce moteur est utilisé plutôt pour les petites puissances. Ce moteur est facile à piloter en variation de vitesse. Son principe de fonctionnement repose sur la force de Laplace.

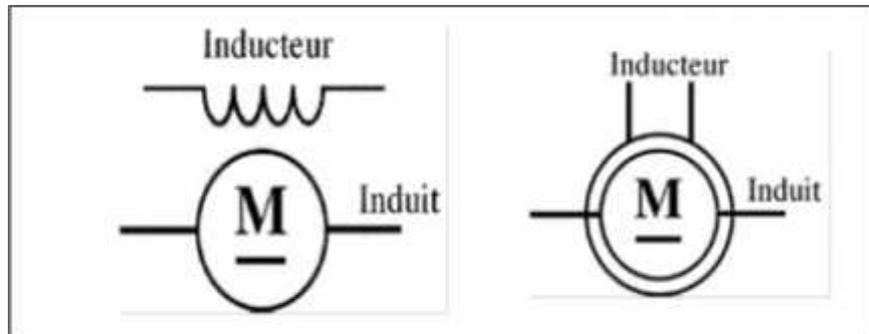


Figure I.10 représentation symbolique d'un moteur à C.C

Constitution

Cette machine est constituée :

1. D'une partie fixe le stator ou inducteur et une partie tournante, le rotor ou induit séparées par un entrefer. Le stator et le rotor sont constitués par un assemblage de tôles en fer au silicium afin de limiter les pertes par courants de Foucault et par hystérésis;
2. D'un circuit inducteur, qui est la source de champ magnétique. Il peut être formé soit par des aimants en ferrite, soit par des bobines inductrices en série (électroaimants). Les bobines sont placées autour de noyaux polaires. La machine est dite bipolaire si elle ne comporte qu'un pôle Nord et un pôle Sud.
3. D'un circuit induit au rotor. Il est formé de conducteurs logés dans des encoches ;

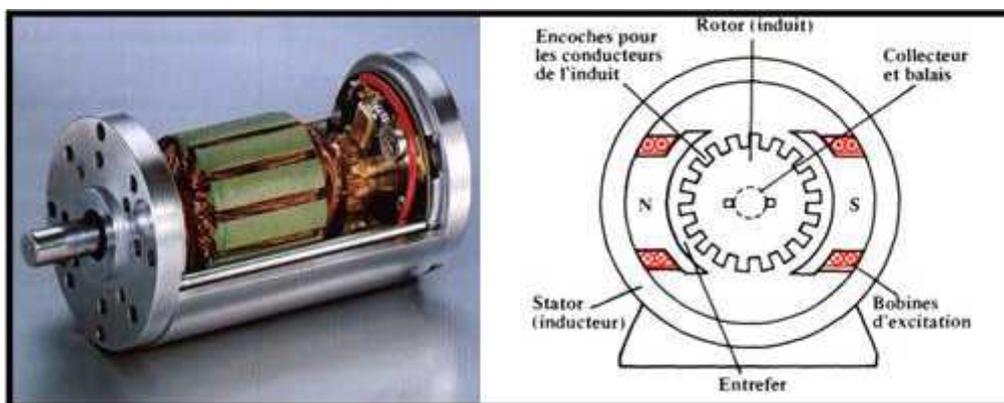


Figure I.11 Moteur à courant continu.

- **Le moteur asynchrone**

Comme les moteurs a cc ils sont destiné à transformer l'énergie électrique (courant alternative) en l'énergie mécanique avec de moyenne et grande puissance avec fiabilité et robustesse, ils fonctionnent dans une large gamme de variation de vitesse. Mais ils sont alimente en mono et triphasé.

Principe de fonctionnement

Le stator bobiné est composé de trois bobines qui créent 3 champs magnétiques. Ces bobines étant alimentées par un système de courants triphasés créent chacune un champ magnétique variable. La composition de ces trois champs magnétiques crée un champ magnétique tournant qui entraîne le rotor en rotation par la création d'une force électromagnétique.

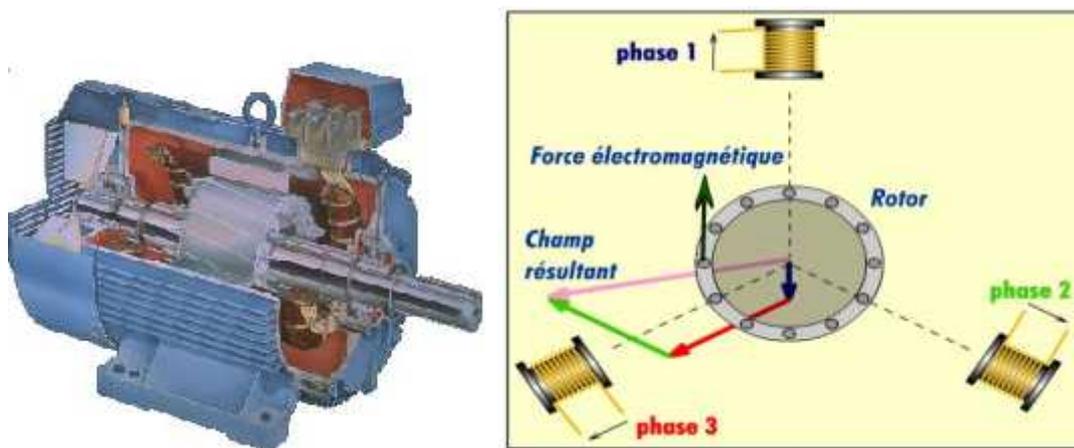


Figure I.12 moteur asynchrone principe de fonctionnement

Le moteur asynchrone monophasé :

Un moteur asynchrone monophasé est constitué de deux bobines et un condensateur pour créer un champ magnétique tournant qui permet d'entraîner le rotor en rotation.



Figure I.13 Le moteur asynchrone monophasé

- Le moteur pas a pas :

Un moteur pas à pas est un moteur qui est alimenté en courant continu. Son rotor est constitué de N pôles magnétiques (Nord et Sud). Son Stator est constitué de bobines qui sont alimentées par un circuit électronique les unes à la suite des autres pour créer des pôles nord et sud en fonction du sens d'alimentation. On trouve par exemple ce type de moteur dans les imprimantes jet d'encre pour déplacer les têtes d'impression ou entraîner le papier.

1.3.1.2 Pré actionneurs

Un pré actionneur est destiné pour distribuer, sur ordre de la P.C., l'énergie utile aux actionneurs.

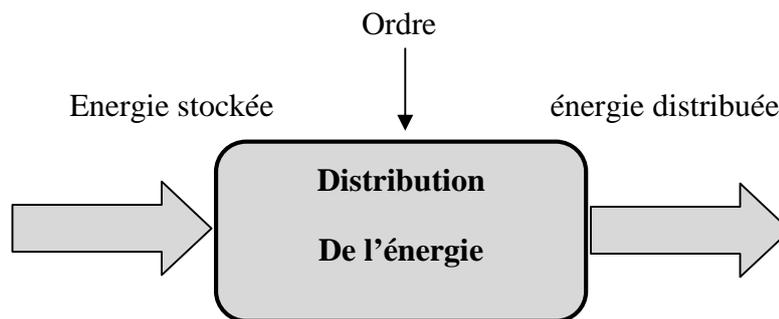


Figure I.14 principe d'un pré actionneur

On distingue deux types de pré actionneurs, pneumatiques et électriques :

a - pré actionneurs pneumatiques

Un pré actionneur électropneumatique ou pneumatique, appelé encore distributeur pneumatique, commande l'établissement et l'interruption de la circulation de l'énergie pneumatique entre la source génératrice (circuit de distribution du fluide) et l'actionneur pneumatique (vérin).

Ce distributeur est donc destiné à diriger le fluide sous pression vers inverse, et par une autre voie, il assure le retour sans pression à l'air libre (échappement).

Les électrovannes : La commande ou le pilotage des distributeurs se fait souvent par un signal électrique. Le rôle de l'électrovanne est de transformer le signal électrique en provenance de la PC en un signal pneumatique destiné à provoquer l'inversion du distributeur.

Le signal de sortie S apparaît lorsque le signal de pilotage électrique alimente la bobine. Le noyau 1 se déplace sous l'effet du champ magnétique ainsi créé



Figure I.15 Distributeur pneumatique



Figure I.16 Electrovanes

b. Les pré- actionneurs électriques

Les pré- actionneurs électriques sont principalement constitués des contacteurs électriques, aussi appelés des relais

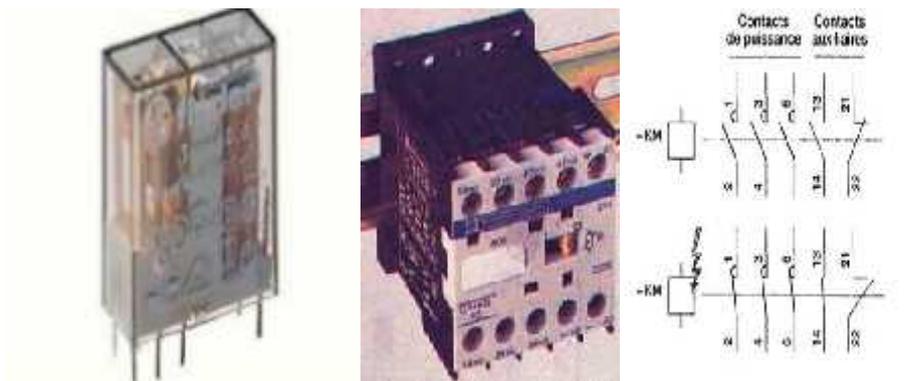


Figure I.17 relais électromécanique et Contacteur électrique de puissance

I.3.2 Partie commande (PC)

C'est la partie centrale, Elle reçoit les informations de l'extérieur ou de la partie opérative.et Donne les ordres aux actionneurs Elle peut se présenter sous 3 manières différentes : un boîtier de commande, un microprocesseur (cerveau électronique), ou un ordinateur .voir figure (a et b)

La réception des informations est assurée par des capteurs.



Figure I.18 Automate programmable et microcontrôleur

I.3.2.1 Les Capteurs:

Sont des éléments de la partie opérative mais ils sont destinés pour donner des renseignements sur l'état de la partie opérative.

Ils peuvent détecter des positions, des pressions, des températures, des débits...

- Les capteurs transforment la variation de grandeur physique liée au fonctionnement du système en informations compréhensibles par la partie commande.
- Peuvent être électriques ou pneumatiques. Signaux du type TOR, Analogique ou Numérique.

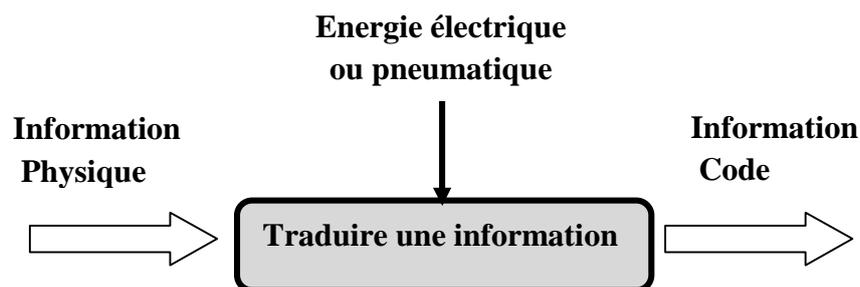


Figure I.19 principe d'un capteur

1.3.2.2 Types de capteur

Capteurs de position : Ce sont des capteurs à contact. Ils peuvent être équipés d'un galet, d'une tige souple, d'une bille. L'information donnée par ce type de capteur est de type tout ou rien et peut être électrique ou pneumatique

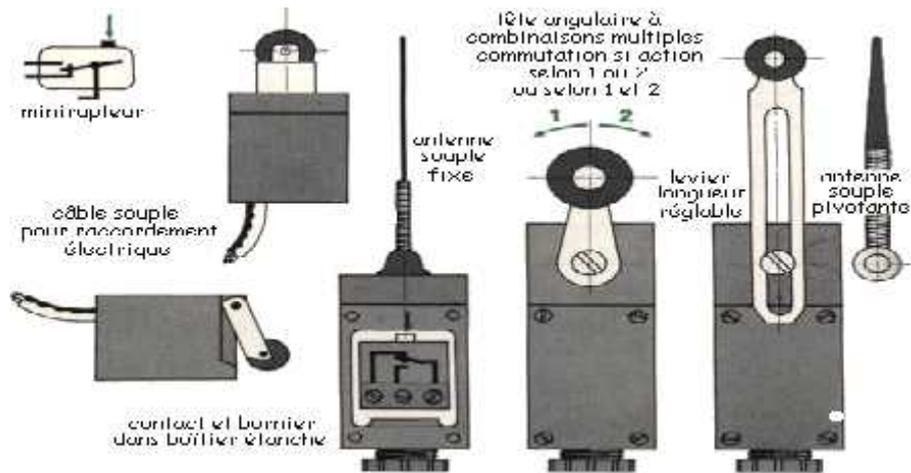


Figure I.20 capteur TOR fin de course

Capteur photoélectrique ou optique : se compose d'un émetteur de lumière associé à un récepteur. La détection d'un objet se fait par coupure du faisceau lumineux

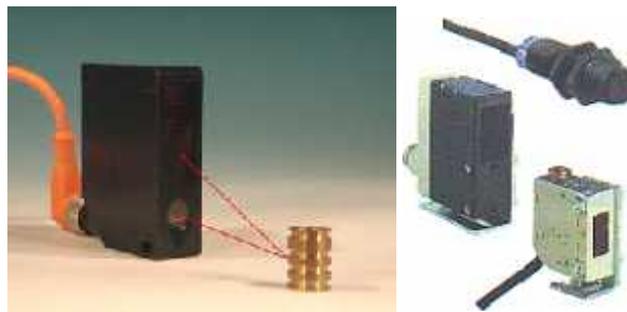


Figure I.21 capteur TOR photoélectrique ou optique

Capteur Capacitif : Capteurs de proximité qui permettent de détecter des objets métalliques ou isolants. Lorsqu'un objet entre dans le champ de détection des électrodes sensibles du capteur, la fréquence des oscillations est modifiée en même temps que la capacité de couplage du condensateur.



Symbole

Figure I.22 capteur TOR effet inductif

Capteur inductif : Ce type de capteur est réservé à la détection sans contact d'objets métalliques. L'objet est donc à proximité du capteur mais pas en contact contrairement à un détecteur de position



Figure I.23 capteur TOR effet inductif

Capteur ILS (Interrupteur à Lame Souple) : Un capteur ILS est un capteur de proximité composé d'une lame souple sensible à la présence d'un champ magnétique mobile. Lorsque le champ se trouve sous la lame, il ferme le contact du circuit provoquant la commutation du capteur. Ce capteur se monte directement sur un vérin et permet de détecter des positions autres que les positions extrêmes. Pour utiliser ce type de capteur, il est nécessaire d'utiliser un vérin comportant un aimant sur le piston.

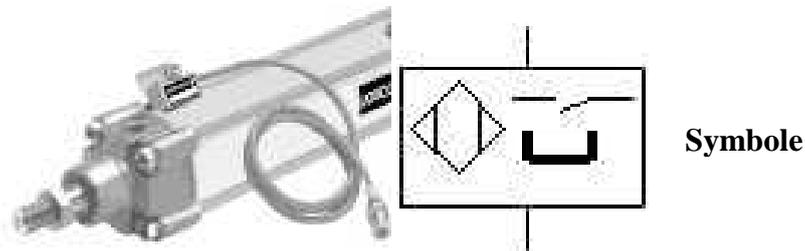


Figure I.24 capteur TOR ILS

I.3.2.3 Les effecteurs:

Les effecteurs ils utilisent l'énergie convertie par les actionneurs pour produire un effet utile. Dans une chaîne d'action, l'effecteur est l'organe terminal qui agit directement sur le produit traité par le système.

Exemple

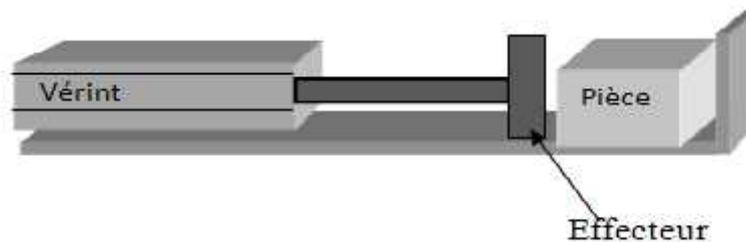


Figure I.25 présentation d'un effecteur

I.3.3 Dialogue homme / machine

Tout système automatisé doit être surveillé ou contrôlé par l'opérateur. Pour cela il faut des équipements comme : Les boutons poussoir, Les terminaux de dialogue, Les afficheurs. ces éléments constituent le mécanisme du dialogue Homme/Machine :

Il existe deux types dialogues :

Dialogue d'exploitation :

L'opérateur exploite le système dans le but de réaliser une opération.

On dit qu'il existe un dialogue entre l'opérateur et la partie commande

C'est un dialogue d'exploitation.



Figure I.25 pupitre de commande

Dialogue de fonctionnement :

C'est la communication entre l'automate et les signaux des capteurs



I.4 Chaîne fonctionnelle

Pour produire des ordres, un système automatique doit connaître les états du système lui-même et de son environnement, ainsi que les informations en provenance de l'utilisateur.

Les états et les consignes sont des grandeurs physiques détectées et mesurées par des capteurs. L'analyse de ces données est assurée par un circuit électronique qui ensuite envoie les ordres à exécuter.

I.4.1 Définition

Une chaîne fonctionnelle représente une unité élémentaire de conception et d'étude d'un système automatisé, Elle est caractérisée par un agencement fonctionnel de constituants sous forme de chaîne qui regroupe tous les éléments de la PC et de la PO concourant à la réalisation d'une tâche opérative.

Exemple :

Serrer une pièce, percer une pièce prendre un objet, déplacer une charge

Une chaîne fonctionnelle comporte deux parties.

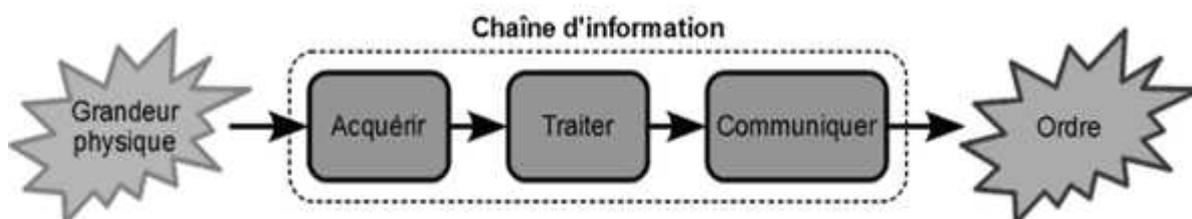
I.4.2 La chaîne d'information

C'est l'ensemble des fonctions et des éléments du système automatisé qui captent l'information et la traitent, ils participent à la prise des décisions.

On peut découper cette chaîne en plusieurs blocs fonctionnels.

- **Acquérir** : Fonction qui permet de prélever des informations à l'aide de capteurs.
- **Traiter** : C'est la partie commande composée d'un automate ou d'un microcontrôleur.
- **Communiquer** : Cette fonction assure l'interface l'utilisateur et/ou d'autres systèmes.
- **Transmettre** : fonction assure l'interface avec l'environnement de la partie commande.

Schéma fonctionnel



Exemple :

Décomposition de la chaîne d'information d'une régulation de température :

- **Grandeurs physiques** : la température souhaitée (consigne) et la température réelle (état).
- **Acquérir** : le thermostat est le capteur qui mesure la température de la maison.
- **Traiter** : le circuit intégré compare la température mesurée à celle désirée.
- **Communiquer** : la carte électronique envoie le bon ordre en fonction de la température.
- **Ordres** : déclencher ou arrêter le chauffage.

La chaîne d'information permet : - d'acquérir des informations ; sur l'état d'un produit ou de l'un de ses éléments (en particulier de la chaîne d'énergie), issues d'interfaces homme/machine ou élaborées par d'autres chaînes d'information, sur un processus géré par d'autres systèmes (consultation de bases de données, partage de ressources) – de traiter ces informations ; – de communiquer les informations générées par le système de traitement pour réaliser l'assignation des ordres destinés à la chaîne d'énergie ou (et) pour élaborer des messages destinés aux interfaces homme/machine (ou à d'autres chaînes d'information).

I.4.3 La chaîne d'énergie

C'est l'ensemble des fonctions et des éléments du système qui participent à la réalisation des opérations [6].

On peut découper cette chaîne en plusieurs blocs fonctionnels.

Les blocs fonctionnels de la chaîne d'énergie

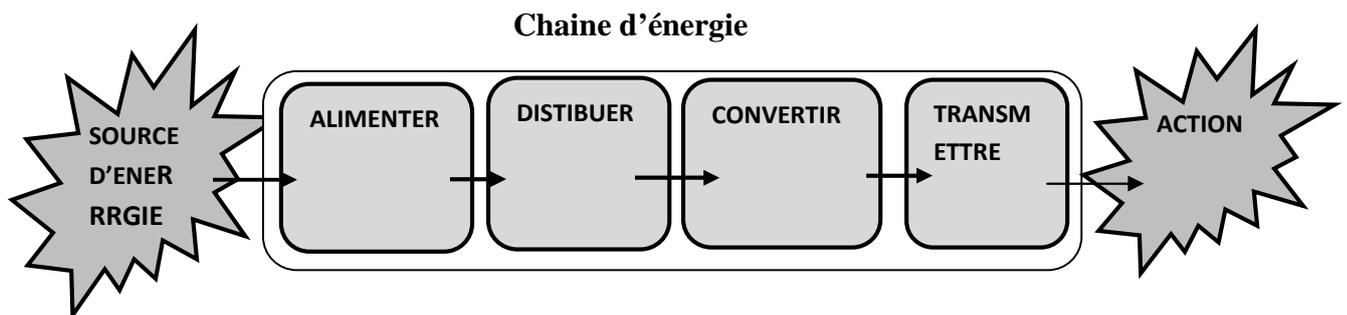
Alimenter : Mise en forme de l'énergie externe en énergie compatible pour créer une action.

Distribuer : Distribution de l'énergie à l'actionneur réalisée par un distributeur ou un contacteur.

Convertir : L'organe de conversion d'énergie appelé actionneur peut être un vérin, un moteur électrique ...

Transmettre Cette fonction est remplie par l'ensemble des organes mécaniques de transmission de mouvement

Schéma fonctionnel



Exemple

Décomposition de la chaîne d'énergie d'un store automatique :

- **Source d'énergie** : l'électricité.
- **Alimenter** : le boîtier d'alimentation fournit l'énergie électrique à tout le système (partie commande et partie opérative).
- **Distribuer** : les contacteurs amènent l'électricité au moteur en fonction des ordres reçus.
- **Convertir** : le moteur transforme l'énergie électrique en énergie mécanique.
- **Transmettre** : les engrenages réduisent la vitesse de rotation du moteur et font tourner le store.
- **Action** : l'ouverture ou la fermeture du store.

Les fonctions Alimenter, Distribuer, Convertir, Transmettre sont réalisées par des composants :

- Les pré-actionneurs qui distribuent l'énergie aux actionneurs en fonction des ordres émis par la chaîne d'information. (Distributeur, contacteur...)
- Les actionneurs qui produisent l'énergie mécanique (vérins, moteurs,...)
- Les dispositifs retenus pour la transmission de l'énergie qui transmettent l'énergie ou agissent sur la matière d'œuvre (ventouse, vannes...)

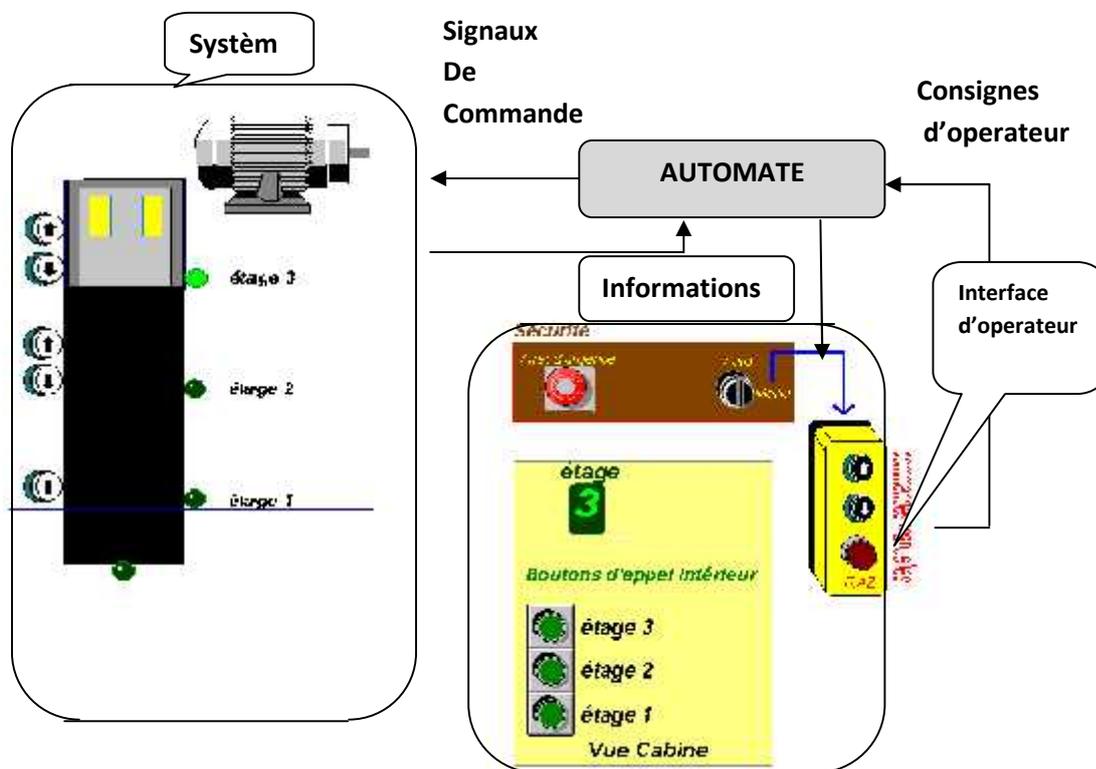


Figure I.26 Eléments de la partie opérative et de partie commande

I.5 Conclusion

Ce 1^{er} chapitre est consacré pour la description générale des systèmes automatisés industriels, leurs objectifs et l'importance de l'automatisation dans le domaine économique. La structure d'un tel système avec les missions de chaque partie et comment communique entre eux. Aussi, pour la connaissance des éléments mécaniques, électriques ou pneumatiques et hydrauliques qui constituent un tel système, comment ils fonctionnent, comment sont ils organisés et le passage de l'information entre les parties. Ainsi que dialogue homme machine.

II.1 Introduction

Les automatismes occupent une place grandissante dans l'activité économique des pays industrialisés. L'automate programmable industriel (API) est au cœur de la production dans les secteurs de l'automobile, de l'alimentation, de la pétrochimie et de la métallurgie, ainsi que dans l'industrie pharmaceutique et dans le traitement des eaux.

Sont apparus aux Etats-Unis vers 1969 où ils répondaient aux désirs des industries de l'automobile de développer des chaînes de fabrication automatisées qui pourraient suivre l'évolution des techniques et des modèles fabriqués [7].

II.2 Définition d'un automate

Un automate programmable est un appareil électronique qui fait un traitement des signaux à l'entrée et envoie d'autre à la sortie, destiné au contrôle d'une machine ou d'un processus industriel, il comporte une mémoire programmable par un utilisateur non informaticien, à l'aide d'un langage adapté.

En d'autres termes, un automate programmable est un calculateur logique, ou ordinateur, destiné à la conduite et la surveillance en temps réel de processus industriels. L'API est distingué par trois caractéristiques fondamentales :

- Il peut être directement connecté aux capteurs et pré-actionneurs grâce à ses entrées/sorties industrielles.
- Il est conçu pour fonctionner dans des ambiances industrielles sévères (température, vibrations, microcoupures de la tension d'alimentation, parasites, et.
- Et enfin, sa programmation à partir de langages spécialement développés pour le traitement de fonctions d'automatisme fait en sorte que sa mise en œuvre et son exploitation ne nécessitent aucune connaissance en informatique

II.3 Structure générale des API :

Les automates programmables industrielle sont construits avec des différentes structures, il existe

- Les compacts,
- Les racks tables
- Les modulaires

SIEMENS LOGO	CROUZET MILLENIUM	SCHNEIDER ZELIO	SCHNEIDER TWIDO
			

Figure II.1 Automate programmables compact.

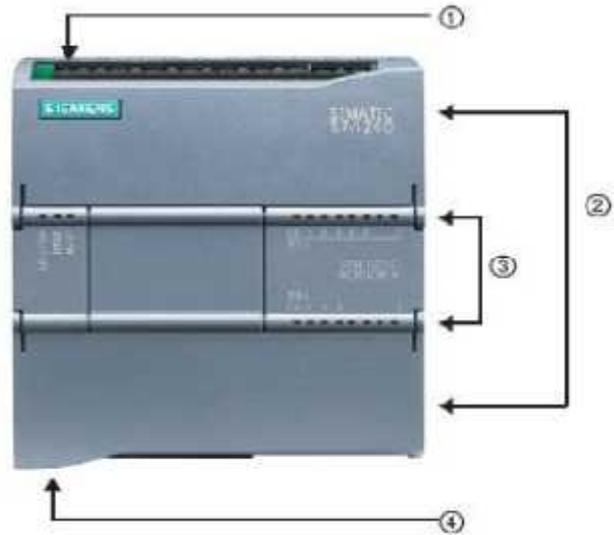
LES RACKTABLES ET MODULAIRES			
SIEMENS S7-300	SCHNEIDER TSX 37	MOELLER	SCHNEIDER TSX 57
			

Figure II.2 Automate programmables modulaire

II.3.1 Caractéristiques d'un automate

- Le type de l'automate Compact ou modulaire
- La tension d'alimentation
- La taille de la mémoire
- Sauvegarde (EPROM, EEPROM, pile, ...)
- Les modules d'entrées / sorties
- Les modules complémentaires (analogique, communication,..)
- Le langage de programmation

Figure II.3 Aspect extérieur d'un automate SIEMENS S7-1200 CPU 1214 AC/DC RELAY



1. Alimentation 24V
2. Borniers insérables pour un câblage utilisateur (derrière les caches plastiques)
3. Diodes électroluminescentes pour les I/O intégrées et le mode de fonctionnement de la CPU
4. Connexion TCP/IP (sous la CPU)

II.3.2 Architecteur interne d'un l'API

- Description générale

L'automate programmable reçoit les informations relatives à l'état du système et les ordres de l'opérateur puis commande les pré-actionneurs suivant le programme inscrit dans sa mémoire. Un API se Compose donc de cinq parties : le processeur, la zone mémoire, les interfaces Entrées/Sorties, L'alimentation et module périphériques (voir figure II.4).

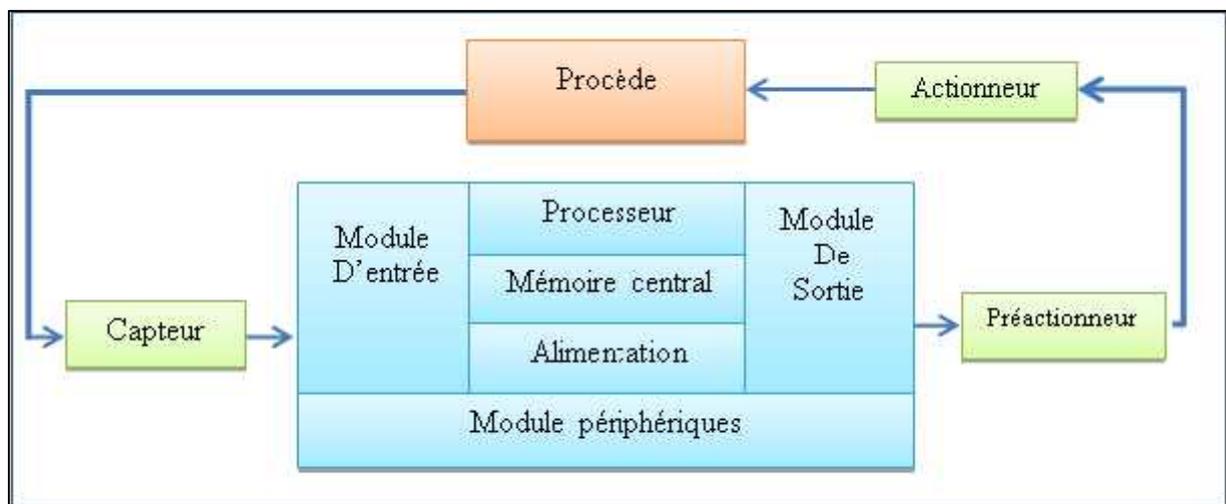


Figure II.4 Architecteur d'un API.

- **Alimentation de l'automate programmable industriel**

L'alimentation, intégrée dans l'API, fournit à partir des tensions usuelles de réseau (230V, 24 V) les tensions continues nécessaires au fonctionnement des circuits électroniques. Le module d'alimentation assure la distribution d'énergie aux différents modules.

- **Microprocesseur**

Le processeur, ou unité centrale (UC), a pour rôle principal le traitement des instructions qui constituent le programme de fonctionnement de l'application (les fonctions logiques ET, OU, les fonctions de temporisation, de comptage, de calcul PID, etc..). Mais en dehors de cette tâche de base, il réalise également d'autres fonctions :

- **Gestion des entrées/sorties.**

Surveillance et diagnostic de l'automate par une série de tests lancés à la mise sous tension ou cycliquement en cours de fonctionnement.

Dialogue avec le terminal de programmation, aussi bien pour l'écriture et la mise au point du programme qu'en cours d'exploitation pour des réglages ou des vérifications des données.

Un ou plusieurs processeurs exécutent ces fonctions grâce à un micro logiciel préprogrammé dans une mémoire de commande, ou mémoire système. Cette mémoire morte définit les fonctionnalités de l'automate. Elle n'est pas accessible à l'utilisateur.

Mémoires

Elles sont utilisées pour le stockage des programmes de fonctionnement de l'automatisme, ainsi que des données qui peuvent être :

Des informations susceptibles d'évoluer en cours de fonctionnement de l'application. C'est le cas par exemple de résultats de traitements effectués par le processeur et rangés dans l'attente d'une utilisation ultérieure. Ces données sont appelées variables internes ou mots internes.

Des informations qui n'évoluent pas au cours de fonctionnement, mais qui peuvent en cas de besoin être modifiées par l'utilisateur : textes à afficher, valeurs de présélection, etc.. Ce sont des mots constants.

Les mémoires d'état des entrées/sorties, mises à jour par le processeur à chaque tour de scrutation du programme.

Deux familles de mémoires sont utilisées dans les automates programmables :

Les mémoires vives, ou mémoires à accès aléatoire « Random Access Memory (RAM) ». Le contenu de ces mémoires peut être lu et modifié à volonté, mais il est perdu en cas de manque de tension (mémoire volatiles). Elles nécessitent par conséquent une sauvegarde par batterie.

Les mémoires vives sont utilisées pour l'écriture et la mise au point du programme, et pour le stockage des données.

Elles sont à lecture seule, les informations ne sont pas perdues lors de la coupure de l'alimentation des circuits. On peut citer les types suivants

ROM « Read Only Memory » : Elle est programmée par le constructeur et son programme ne peut être modifié.

PROM « Programmable ROM » : Elle est livrée non enregistrée par le fabricant. Lorsque celle-ci est programmée, on ne peut pas l'effacer

EPROM « Erasable PROM » : C'est une mémoire PROM effaçable par un rayonnement ultraviolet intense.

EEPROM « Electrically EPROM » : C'est une mémoire PROM programmable plusieurs fois et effaçable électriquement.

Mémoire Flash : C'est une mémoire EEPROM rapide en programmation.

L'utilisateur peut effacer un bloc de cases ou toute la mémoire.

La mémoire morte est destinée à la mémorisation du programme après la phase de mise au point. La mémoire programme est contenue dans une ou plusieurs cartouches qui viennent s'insérer sur le module processeur ou sur un module d'extension mémoire.

Interfaces entres/sorties

Les entrées et les sorties assurent la communication directe de l'automate avec son environnement industriel en réalisant la liaison entre le processeur et le processus elles peuvent être :

- Logique : entrées et sorties tout ou rien
- Analogique : génératrice tachymétrique en entrée et variateur de vitesse en sortie
- Numérique : comptage rapide sur un codeur incrémental.

Chaque entrée ou sortie devra être adaptée au capteur ou au pré actionneur.

Les cartes assurent l'isolation galvanique entre l'unité centrale et le système.

Les cartes de sortie peuvent être à relais ou à transistor.

- Celles à relais assurent une coupure entre l'alimentation et le pré actionneur mais sont relativement lentes.

- Celles à transistor commutent plus rapidement mais n'assurent pas de séparation électrique.

Elles ont toutes, de base, une double fonction :

- Une fonction d'interface pour la réception et la mise en forme de signaux provenant de l'extérieur (capteurs, boutons poussoirs, etc.).

- La conception de ces interfaces avec un isolement galvanique ou un découplage opto-électronique assure la protection de l'automate contre les signaux parasites.
- Une fonction de communication pour l'échange des signaux avec l'unité centrale par l'intermédiaire du bus d'entrées/sorties.

Le fonctionnement de l'interface d'entrée **figure II.5** peut être résumé comme suit :

- 1- Lors de la fermeture du capteur ;
- 2- La « Led 1 » signale que l'entrée de l'API est actionnée.
- 3- La « Led D' » de l'optocoupleur « Opto 1 » s'éclaire.
- 4- Le phototransistor « T' » de l'optocoupleur « Opto 1 » devient passant.
- 5- La tension $V_s=0V$.

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

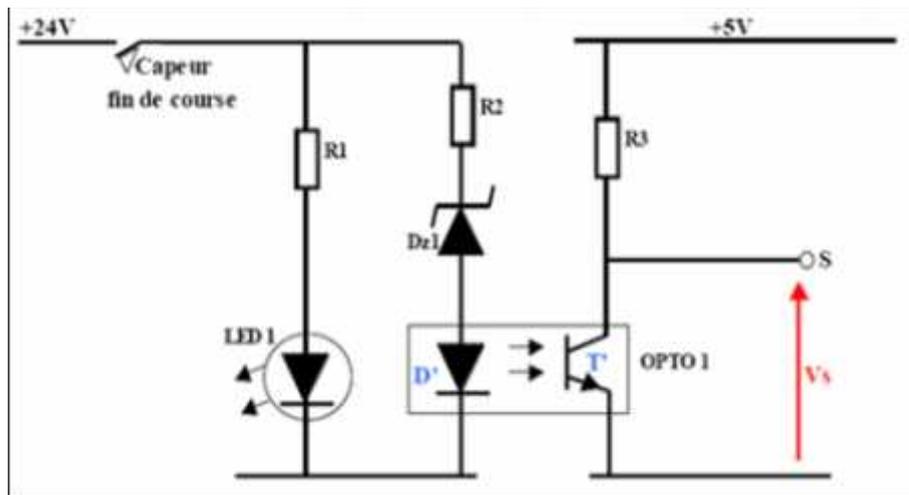


Figure II.5 Exemple d'une carte d'entrées typique d'un API.

- Le fonctionnement de l'interface de sortie figure II-5 peut être résumé comme suit

Lors de commande d'une sortie automate ;

- L'unité de commande envoie un « 1 » logique (5V).
- « T1 » devient passant, donc la « Led D' » s'éclaire
- Le photo-transistor « T' » de l'optocoupleur « Opto1 » devient passant.
- La « Led1 » s'éclaire.
- « T2 » devient passant.
- La bobine « RL1 » devient sous tension et commande la fermeture du contact de la sortie « Q0.1 ».

Donc pour commander un API, l'unité de commande doit envoyer :

- Un « 1 » logique pour actionner une sortie API
- Un « 0 » logique pour stopper la commande d'une sortie API

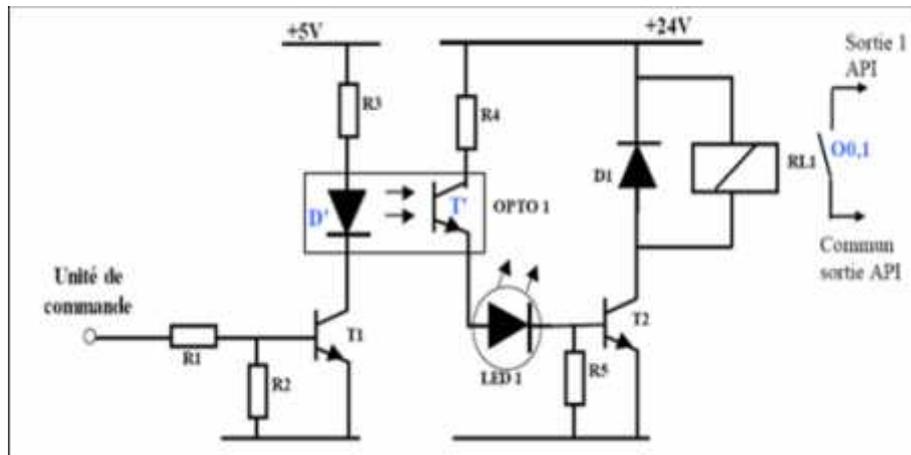
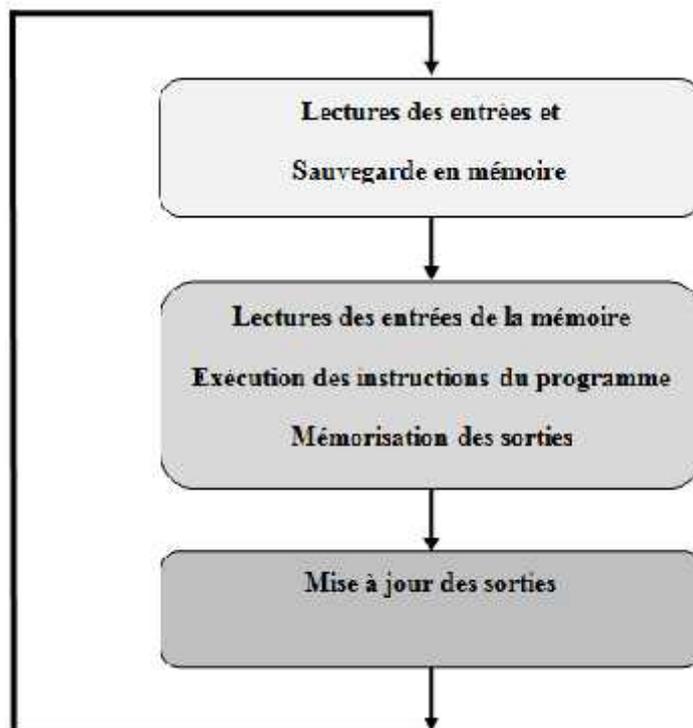


Figure II.6 Exemple d'une d'interface de sortie typique d'un API [08].

Les interfaces entre/sortie contiennent des borniers de raccordement avec les capteurs et les prés actionneurs et un ensemble des LEDs pour la visualisation de l'état de chaque voie.

II.3.3 Cycle de fonctionnement d'un automate :

L'automate programmable fonctionne par déroulement cyclique du programme. Le cycle comporte trois opérations successives qui se répètent normalement comme suit :



La structure modulaire d'un API donne une configuration souple qui est compatible aux besoins de l'utilisateur avec un diagnostic et une maintenance facile.

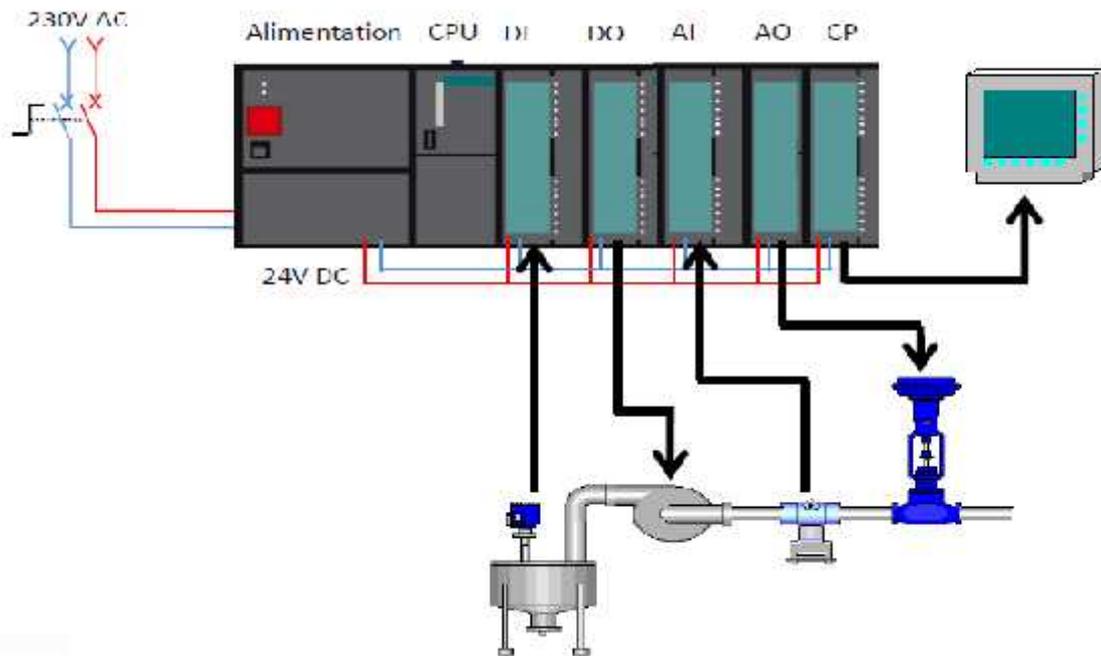


Figure II.7 principe de lecture et commande de l'API

II.3.4 Cartes d'entrées tout ou rien (TOR)

Une carte d'entrée comporte généralement 8, 16,32 entrées logiques comme le montre le schéma fonctionnel ci-dessous.

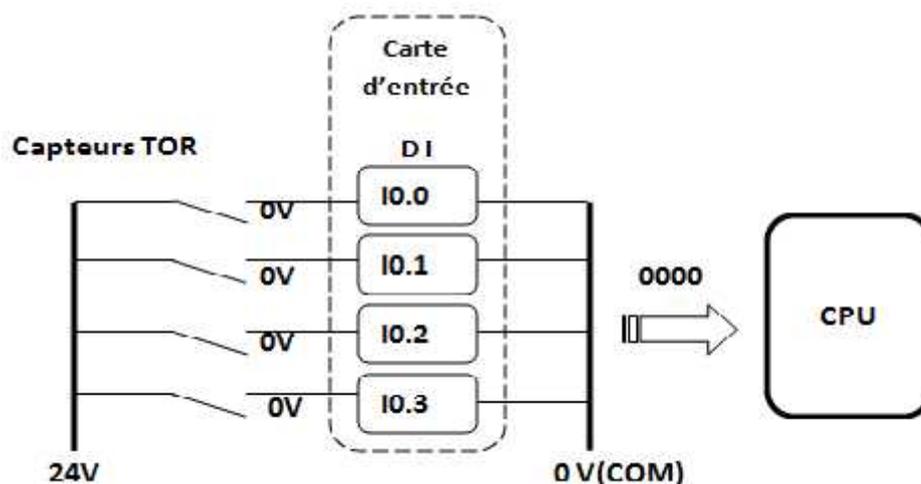


Figure II.8 a principe de connexion des entrées état au repos

Il existe deux types de contacts, des contacts à ouvertures qui sont fermés au repos et des contacts à fermetures qui sont ouverts au repos.

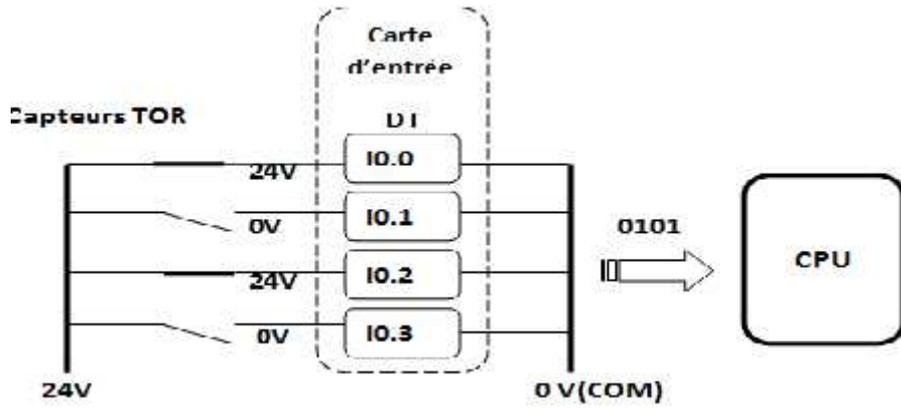


Figure II.8b principe de connexion des entrées état actionnée

Toutes les capteurs TOR sont alimenter par une tentions de 24 V fournie par l'automate et le retour doit être branché aux entrées IO.1, IO.2, IO.3.....de l'automate.

II.3.5 Cartes de sorties TOR :

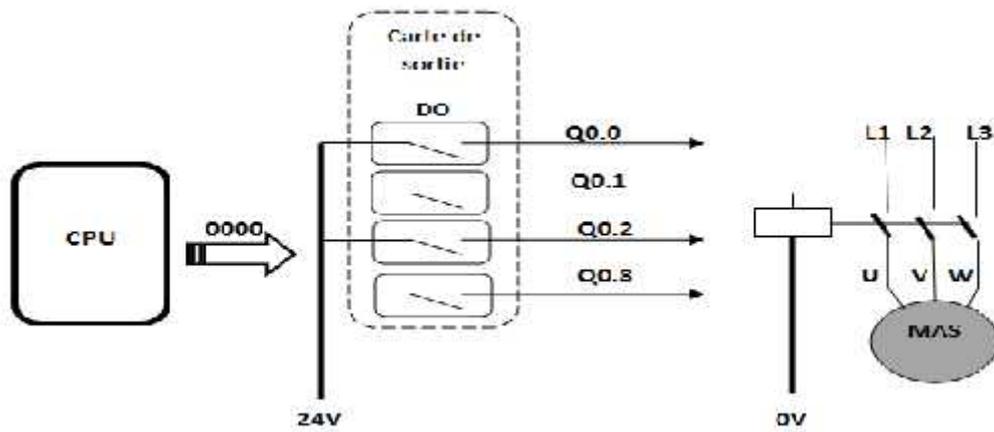


Figure II.9a câblage des sorties en état au repos

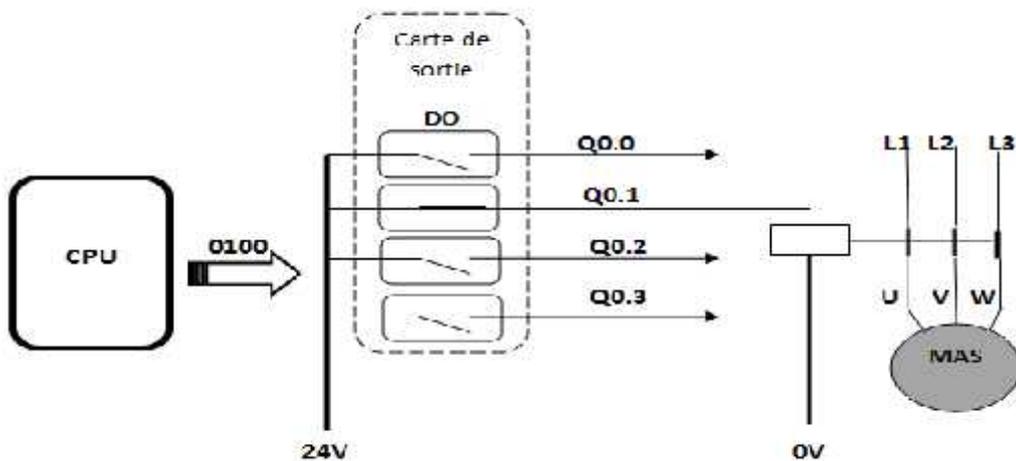


Figure II.9b câblage des sorties en état actionné

Les éléments du système sont commandés par l'automate a travers les prés actionneurs connecter aux bornes de l'interface de sortie avec des tentions de 24v par exp C'est avec les signaux sortants de l'automate qu'on peut démarrer ou arrêter un moteur, avancer ou reculer un vérin, allumer ou éteindre une lampe. C'est ne fonctionne que si le programme stocker dans l'automate autorise la fermeture des relais des sorties.

II.3.6 Cartes d'entres sorties analogiques :

Un signal analogique est un signal (une information codée) qui peut varier de manière continue dans le temps. Autrement dit, à chaque instant, le signal peut avoir n'importe quelle valeur comprise dans les limites d'un certain intervalle. On l'oppose généralement au signal numérique qui, lui, n'est codé qu'avec un nombre limité de valeurs (par exemple soit "0", soit "1") [09].

Exemples

- Une température qui varie entre -20° et $+150^{\circ}$
- Une pression varie entre 0 bar et 10 bars
- Un débit varie entre 0 litre et 100 litre

L'automate communique avec ce type de variable par l'intermédiaire d' transducteur

II.3.7 Choix d'un API

Le choix d'un API est fonction de la partie commande à programmer. On doit tenir compte de plusieurs critères.

- Nombres d'entrées/sorties intégrés.
- Temps de traitement (scrutation).
- Capacité de la mémoire.
- Nombre des blocs nécessaire compteurs et temporisateurs.

II.4 Langages de programmation des API

Cinque langages peuvent être utilisé pour la programmation des automates programmables industrielles (API) selon la norme IEC 1131-3, sont les suivantes

1. Langage FBD Fonction Block Diagram
2. Langage IL Instruction List
3. Langage ST Structured Text
4. Langage LD ladder diagram
5. Langage SFC GRAFCET

II.4.1 Les composants du langage

Il existe trois types d'élément de langage :

1. les entrées (contact) : permettent de lire la valeur d'une variable booléenne
2. les sorties (bobines) : permettent d'écrire la valeur d'une variable booléenne,
3. les blocs fonctionnels : permettent de réaliser des fonctions avancées.

Le tableau ci dessous donne les principaux éléments (contacts et bobines) d'un réseau LD.

Symbole graphique	désignation
- -	Contact normalement ouvert
- / -	Contact normalement fermé
- P -	Contact fermé au front montant
- N -	Contact fermé au front descendant
-()-	Bobine normalement ouverte
-(/)-	Bobine normalement fermée
-(S)- ou -(L)-	Bobine Latch (maintenu à 1 une fois actionné)
-(R)-ou -(U)-	Bobine Reset (remise à 0 de la bobine latch)
-(P)-	Bobine active au front montant de son entrée
-(N)-	Bobine active au front descendant de son entrée
-<return>	Retour incondionnel (vers le sous-programme appelant)
-<cond-<return>	Retour conditionnel
->>Label	Saut incondionnel
-<cond->>Label	Saut conditionnel

Figure II.10 : les principaux éléments d'un réseau LD.

II.5 Langage SFC GRAFCET

Le langage SFC (Séquentiel Fonction Charte), ou GRAFCET [11].

C'est une description graphique utilisé pour décrire des taches séquentielles. Il est représenté comme une suite contient des étapes associé à des actions, reliées entre elles par des transitions associé à des réceptivités.

Une condition booléenne est attachée à chaque transition. Les actions dans les étapes sont décrites avec les langages ST, IL, LD ou FBD.

II.5.1 Règles du GRAFCET

- Un GRAFCET doit contenir au moins une étape initiale.
- Une étape ne peut pas être suivie d'une autre étape.
- Une transition ne peut pas être suivie d'une autre transition.

II.5.2 Composants de base (symboles graphiques) du GRAFCET sont :

1. Étapes et étapes initiales.
2. transitions.
3. liaisons orientées.
4. renvoi à une étape.

II.5.3 Types d'action

- Action booléenne (Elle est forcée à chaque fois que le signal d'activité de l'étape change d'état.)
- Action impulsionnelle programmée en ST, LD ou IL (c'est une liste d'instructions ST, IL ou LD, exécutée à chaque cycle pendant toute la durée d'activité de l'étape).
- Action normale programmée en ST, LD ou IL ;
- Action SFC (Une action SFC est une séquence fille SFC, lancée ou tuée selon les évolutions du signal d'activité de l'étape. Elle peut être décrite avec les qualificatifs d'action N (non mémorisée), S (set), ou R (reset).)
- Plusieurs actions (de même type ou de types différents) peuvent être décrites dans la même étape. Un appel de fonctions ou de blocs fonctionnels permet d'intégrer des traitements décrits dans d'autres langages (FBD, LD, ST ou IL).

II.5.4 Avantages du GRAFCET

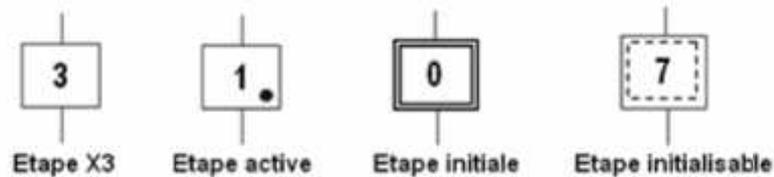
1. *Il est* indépendant de la matérialisation technologique;
2. *Traduit* le cahier des charges *de* façon cohérente;
3. Le GRAFCET est fortement *adapté aux systèmes automatisés*.

II.5.5 Étapes et actions associées

- Etape

Une étape symbolise un état ou une partie de l'état du système automatisé. L'étape possède deux états possibles : active représentée par un jeton dans l'étape ou inactive. L'étape *i*, représentée par un carré repéré numériquement, possède ainsi une variable d'état, appelée variable d'étape *X_i*. Cette variable est une variable booléenne valant 1 si l'étape est active, **0** sinon.

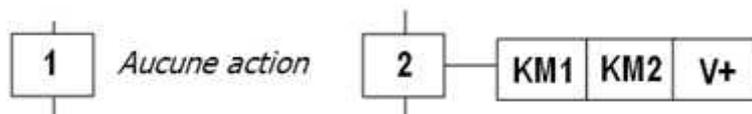
La situation initiale d'un système automatisé est indiquée par une étape dite étape initiale et représentée par un carré double [12].



Remarque : Dans un GRAFCET il doit y avoir au moins une étape initiale.

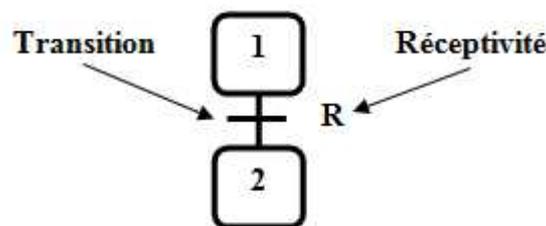
- Actions associées aux étapes

A chaque étape est associée une action ou plusieurs, c'est à dire un ordre vers la partie opérative ou vers d'autres GRAFCETS. Mais on peut rencontrer aussi une même action associée à plusieurs étapes ou une étape vide (*sans action*).



- Transition

Une transition indique la possibilité d'évolution qui existe entre deux étapes et donc la succession de deux activités dans la partie opérative. Lors de son franchissement, elle va permettre l'évolution du système. A chaque transition est associée une condition logique appelée réceptivité qui exprime la condition nécessaire pour passer d'une étape à une autre.



- Réceptivité

C'est une information d'entrée qui est fournie par :

- L'opérateur : pupitre de commande,
- La partie opérative : états des capteurs,
- Du temps, d'un comptage ou toute opération logique, arithmétique...
- Du GRAFCETS : d'autres GRAFCET pour la liaison entre GRAFCETS ou de l'état courant des étapes du GRAFCET (les X_i),
- D'autres systèmes : dialoguent entre systèmes,
- Si la réceptivité n'est pas précisée, alors cela signifie qu'elle est toujours vraie (=1)

- **L'étape initiale**

Toute étape initialement activée est repérée par un carré double.

- **Liaisons orientées:**

Elles relient les étapes aux transitions et les transitions aux étapes. Le sens général d'évolution n est du haut vers le bas. Dans le cas contraire, des flèches doivent être employées

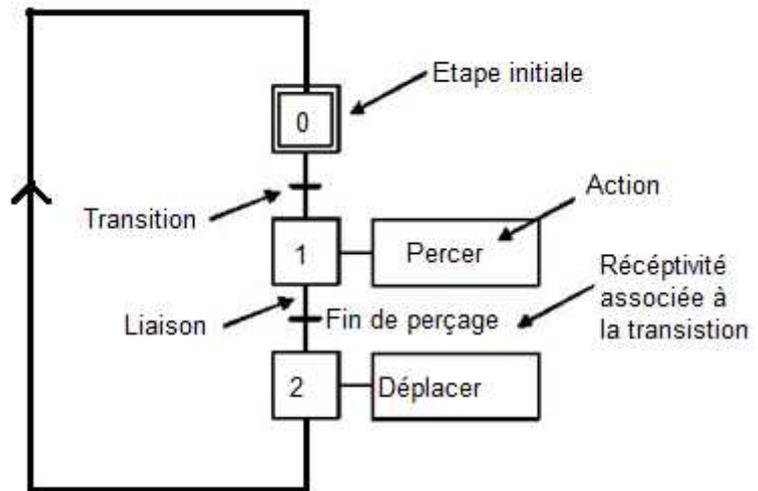


Figure II.11 Exemple d'un model type du GRAFCET [13].

II.5.6 Gestion des modes Marche/ Arrêt ET Arrêts d'urgences

Une seule étape doit être active au début d'un GRAFCET : c'est l'étape initiale et toutes les autres étapes doivent être désactivées.

On appelle la variable d'initiation Init telle que :

- Init=1 initiation du cycle de fonctionnement c'est le mode Arrêt
- Init=0 déroulement de cycle c'est le mode Marche

On appelle Arrêt d'urgence (AU dur et Audoux) telle que :

- AU dur=1 : désactivation de toutes les étapes
- Audoux=1 : désactivation des actions les étapes restent actives
- **Equation d'une étape initial**

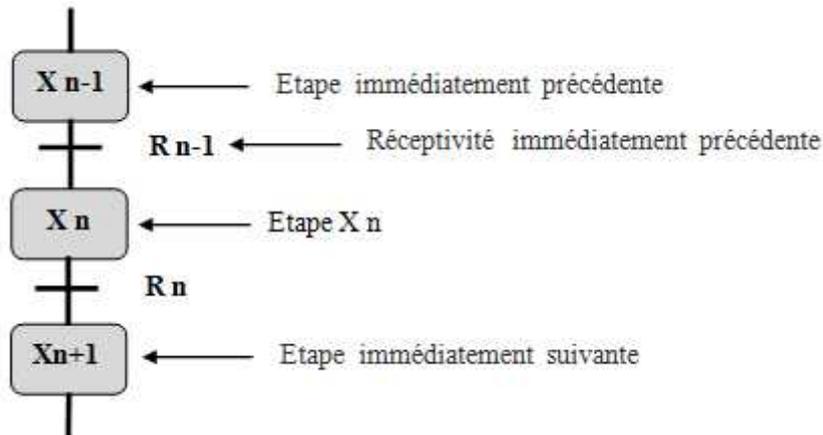
$$Xi = (\overline{CAXi + CDXi} \cdot \overline{Xi + init}) \cdot \overline{AUdur}$$

- **Equation d'une étape non initial**

$$Xi = (\overline{CAXi + CDXi} \cdot Xi) \cdot \overline{init} \cdot \overline{AUdur}$$

II.5.7 Mise en équation d'un GRAFCET :

Soit le système automatisé qui est représenté par le GRAFCET suivant :



Pour que l'étape X n soit activée il faut que :

1. L'étape immédiatement précédente soit active ;
2. La réceptivité immédiatement précédente soit vraie ;
3. L'étape immédiatement suivante soit non active ;
4. Après activation l'étape mémorise son état.

L'équation logique qui assure ces règles est comme suit :

$$X_n = (X_{n-1} \cdot R_{n-1} + X_n) \cdot \overline{X_{n+1}}$$

II.6. Développement et téléchargement du programme utilisateur

Le développement de tel programmes nécessite des logiciels spéciaux qui assurent le codage des instructions .chaque fabricant fournit des logiciels (outil de développement) adapter a ces propres produits. Ces logiciels implanter sur PC (ou console de programmation) sont utilises par l'automaticien comme outils pour développer des programmes spécifiques selon l'application souhaite .une fois mis au point ce programme est alors transférer vers l'API par une liaison de communication RS-232, ou (câble Ethernet) entre le PC et l'API.

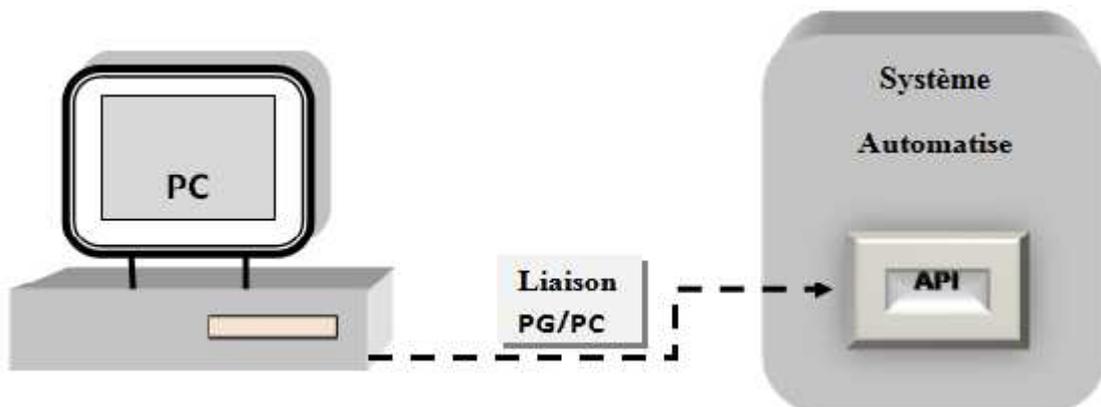


Figure II.12 l'opération du chargement du programme dans la CPU

II.6.1 La programmation en Tia portal

- Introduction

Le Tia portal est une plateforme pour un nouvel environnement de travail SIEMENS. Afin de trouver de solutions d'automatisation avec un système d'ingénierie intégré [14].

Dans ce travail toutes les expériences sont faites sur un automate SIEMENS model s7 1200 qui utilise le logiciel de programmation TIA portal comme outil de développement de Programmes.

- Définition

L'automate SIEMENS model s7 1200 est un mini contrôleur modulaire, composé d'une CPU qui est équipée d'entrées et de sorties numériques ou analogiques. Il est extensible car il permet d'ajouter et installer plusieurs modules additionnels d'entrées/sorties (modules IO), si les entrées et sorties intégrées ne sont pas suffisantes pour l'application désirée.



Modules centraux CPU 1214C

Module de puissance PM

- Vue du portail et vue du projet

Le logiciel TIA Portal, possède deux types de vue différentes : la vue du portail et la vue du projet.

- Vue du portail

C'est une vue qui permet l'axées rapide des taches à exécuter et appelé l'outil qui servira à accomplir la tache voulue. Cette vue simplifie principalement la préparation et la mise en place du projet.



Figure VI.1 vue du portail

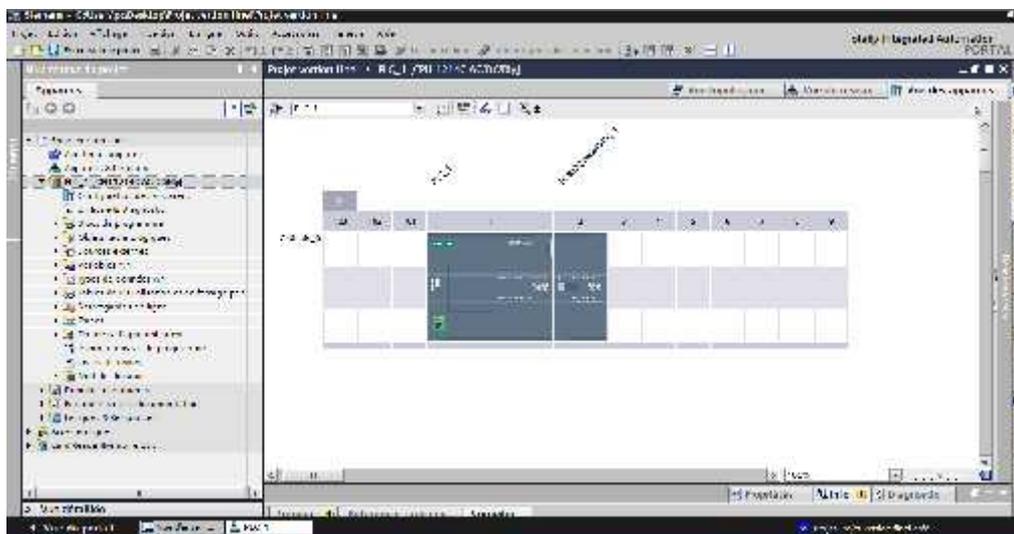


Figure VI.2 vue du projet

➤ **Vue du projet**

Elle comporte une arborescence avec les différents éléments du projet.

La barre de menu avec les barres de fonction sont situées en haut de la fenêtre.

La sélection du projet et toutes ces éléments sont sur la gauche et les menus associés aux différents tâches sur l'adroite.

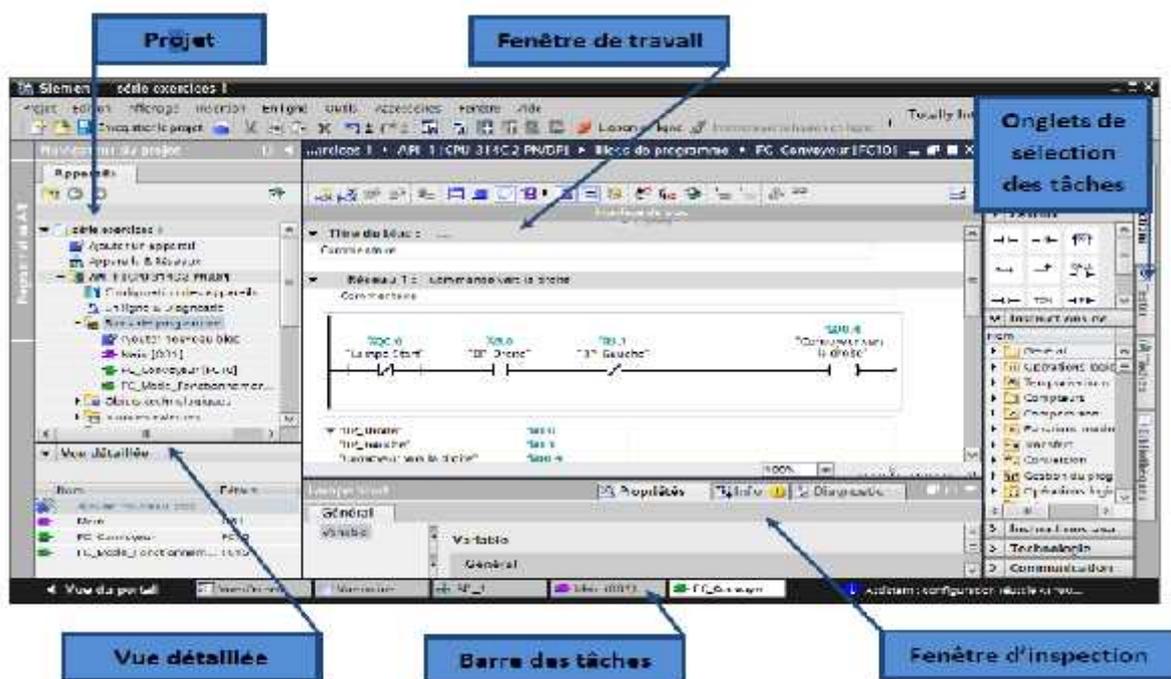
Tout élément sélectionné, est affiché au centre et y peut être manipulé. Comme il est indiqué sur la figure

- La fenêtre de travail permet de visualiser les objets sélectionnés dans le projet pour être traités. Il peut s'agir des composants matériels, des blocs de programme, des tables des variables, des HMI,...

- La fenêtre d'inspection permet de visualiser des informations complémentaires sur un objet sélectionné ou sur les actions en cours d'exécution (propriété du matériel sélectionné, messages d'erreurs lors de la compilation des blocs de programme,...).
- Les onglets de sélection de tâches ont un contenu qui varie en fonction de l'objet sélectionné (configuration matérielle - bibliothèques des composants - bloc de programme - instructions de programmation).

Cet environnement de travail contient énormément de données. Il est possible de masquer ou réduire certaines de ces fenêtres lorsque l'on ne les utilise pas.

Il est également possible de redimensionner, réorganiser, désancrer les différentes fenêtres

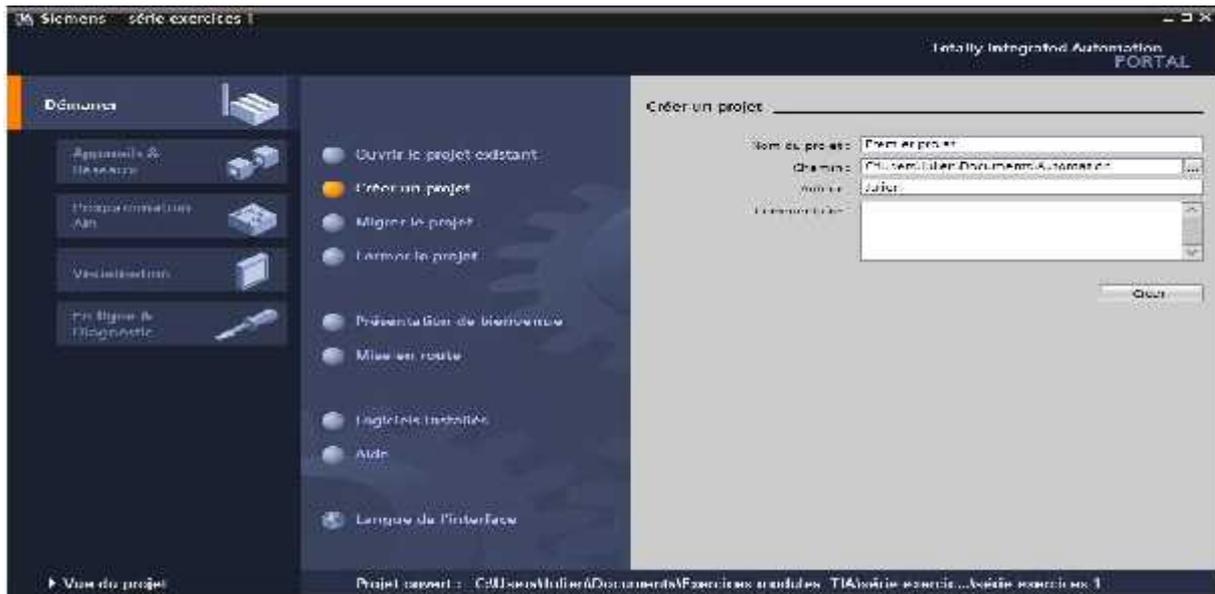


- Création d'un projet

Elle est faite par la sélection de l'action « Créer un projet ».

L'onglet nous donne la main de donner un nom au projet, choisir un chemin pour l'enregistrer, rédiger un commentaire ou encore définir l'auteur du projet.

Une fois que ces informations sont entrées, il suffit de cliquer sur le bouton « créer »



« Mise en route » est une instruction qui débute la création du projet.

- **Configuration et paramétrage d'un appareil**

La première étape consiste à définir le matériel existant. Pour cela, on peut distinguer deux possibilités pour la configuration d'un appareil

- **Dans la vue portail :** on sélectionne « appareils réseaux » et en clique sur « ajouter un appareil »



- **Dans la vue du projet :** on fait un double cliquer sur « ajouter un appareil » sous le nom de projet



Boîte de dialogue « Ajouter un appareil »



La liste des éléments que l'on peut ajouter apparait (API, HMI, système PC). On commencera par faire le choix de notre CPU pour ensuite venir ajouter les modules complémentaires (alimentation, E/S TOR ou analogiques, module de communication AS-i,...)

- **Insertion d'une CPU**

Vous créez votre configuration d'appareil en insérant une CPU dans votre projet. Veuillez à Bien insérer le modèle et la version de firmware corrects à partir de la liste. Sélectionner la CPU dans la boîte de dialogue "Ajouter un appareil" crée le châssis et la CPU.

- **Chargement d'éléments programme dans la CPU**

L'or du chargement d'un projet dans la CPU, on peut charger les éléments du projet aussi.



Le chargement du programme (projet) de la console de programmation à la CPU se fait a partir de l'un des endroits suivants :

"Arborescence du projet" : Cliquez avec le bouton droit de la souris sur l'élément de programme, puis cliquez sur la commande "Charger" du menu contextuel.

Menu "En ligne" : Cliquez sur la commande "Charger dans l'appareil".

Barre d'outils : Cliquez sur l'icône "Charger dans l'appareil".

II.7 Conclusion

Ce chapitre permet de se familiariser avec la structure matérielle et logicielle des automates programmables. Particulièrement SIEMENS ; et l'analyse des cahiers de charges. Afin de produire des solutions programmées à des problèmes d'automatismes séquentiels simples utilisant des capteurs et des actionneurs et un API.

CHAPITRE III : Modélisation et contrôle du système PTP ROBOT TRAINER

III.1 Description du système

Le système PTP ROBOT TRAINER est une chaîne fonctionnelle qui fait le Transfer de pièces, le triage et le stockage dans un magasin.

Il se compose de trois parties principales

- 1) Une partie contient un magasin des pièces, un convoyeur (tapis) et le bras avec la pince. et un magasin de stockage de pièces qui se déplace horizontalement : C'est la chaîne fonctionnelle.
- 2) Une partie de commande et de contrôle, c'est un automate programmable de type SIEMENSE S7 1200.
- 3) Une interface opérateur qui est relié avec l'automate et qui permet la commande manuelle du système.



Figure III.1 représentation du système PTP PNEUMATIC ROBOT TRAINER-TK[15].

Les éléments actionneurs du système sont définis dans le tableau suivant :

N°	ACTIONNEUR	MISSION
1	Vérin de distribution	distribution de pièce du magasin vers le tapis
2	Moteur convoyeur	transporte la pièce jusqu'au bout du tapis
3	Vérin de translation axe X	Déplacement horizontale de la pince
4	Vérin de translation axe Z	Déplacement verticale de la pince
5	Vérin de fermeture pince	Attrape la pièce
6	Vérin rotatif bras	Assure la rotation bras
7	Vérin magasin	Faire déplacer le magasin horizontalement

Les éléments actionneurs de ce système sont :

- Un moteur a courant continu d'ont la tension $U_n=24$ v
- 05 vérins doubles effets
- Un vérin rotatif doubles effets.

Les éléments pré actionneurs :

- Un kit de 6 électro distributeurs avec un système de mesure et de protection
- **P max**=6 bar.

III.1.1 Le système de contrôle

Le système de contrôle (la partie commande) est représenté par un bloc comprend un automate S7-1200 avec une interface entres/sorties numeriques.et des entres analogiques.



Figure III.2 la partie commande [15].

III.1.2 L'outil de programmation

L'outil de programmation est un logiciel simatic TIA portal installer dans un PC dont le système d'exploitation est le windows 8.1. avec liaison PG/PC entre le PC ET l'automate.

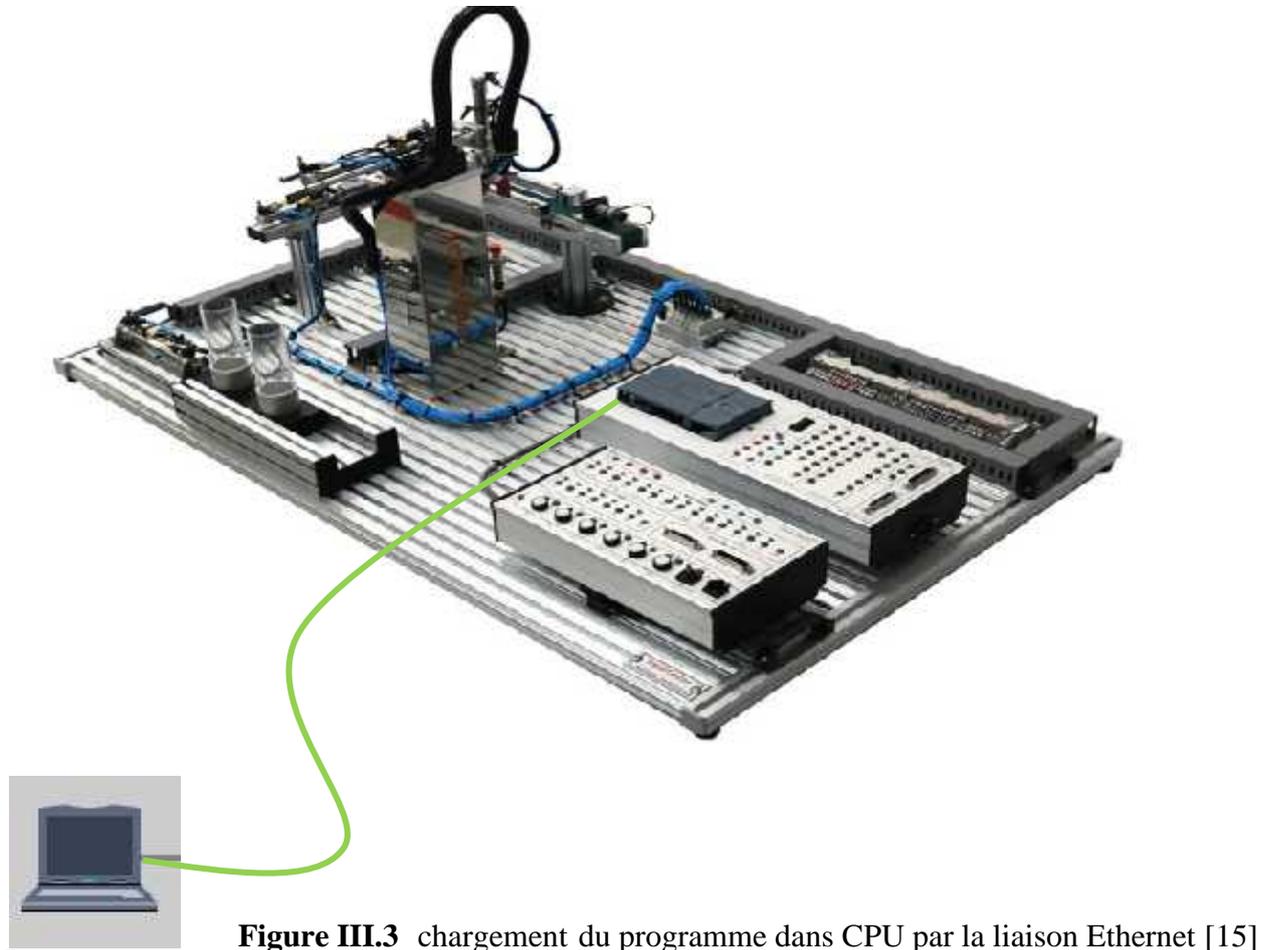


Figure III.3 chargement du programme dans CPU par la liaison Ethernet [15]

III.2 Cahier de charge

Le système représente une chaîne fonctionnelle automatisée qui s'occupe de transférer, trier et mesurer les pièces, une par une, dans un magasin déplaçable. Le déroulement du fonctionnement passe par les étapes suivantes.

III.2.1 Description du cycle de fonctionnement

Le cycle de fonctionnement est réalisé sur trois chaînes

- Distribution de la pièce

- le démarrage du cycle une fois qu'on appuie sur le bouton Start, le système commence à fonctionner tout seul (automatiquement),
- La détection de la pièce est assurée par le capteur S1.
- Un vérin distributeur pousse la pièce vers un tapis roulant (convoyeur), en se déplaçant horizontalement de la position 1B1 à la position 1B2.

- ce dernier commence à rouler et amène la pièce vers l'autre extrémité du tapis, le capteur S2 la détecte.

- Le Triage

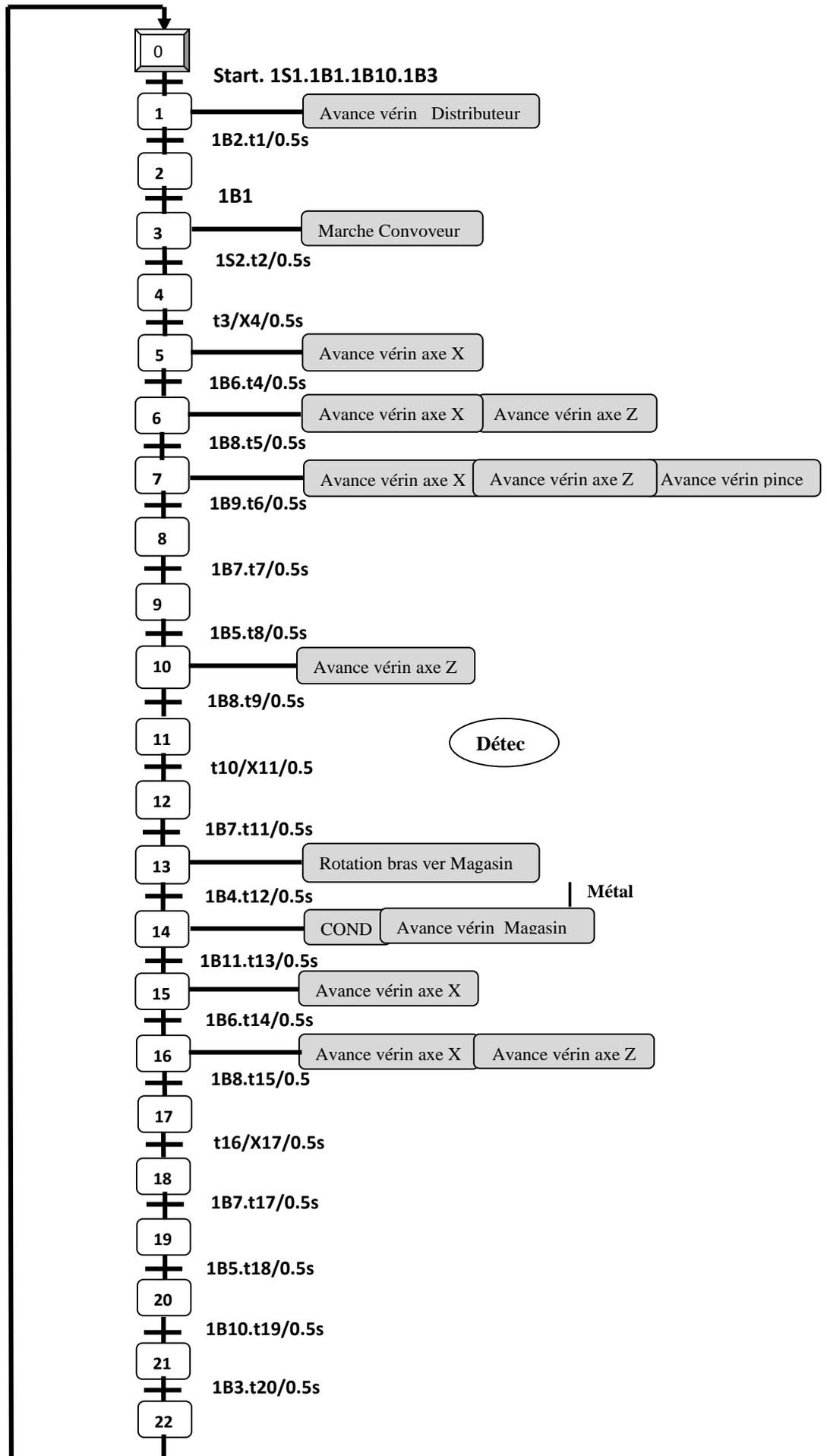
- Le vérin axe X du bras de la pince se déplace de la position 1B5 à la position 1B6.
- Puis le vérin axe Z du bras, de la position 1B7 à la position 1B8. Juste au-dessus de la pièce
- la pince se ferme à la position 1B9 du vérin pour prendre la pièce.
- Le vérin axe Z recule à la position 1B7.
- Le vérin axe X recule à la position 1B5.
Le vérin axe Z avance à la position 1B8, juste au-dessus du détecteur de métal 1S2

- Le transfert de la pièce dans le magasin

- Si la pièce est métallique le magasin avance pour l'acquiescer, le vérin du magasin avance de l'état 1B10 vers l'état 1B11.
- Le vérin axe Z recule à la position 1B7.
- Le vérin rotatif tourne à la position 1B4.
- Le vérin axe X à la position 1B6.
- Le vérin axe Z avance à la position 1B8.
- La pince relâche la pièce.
- Le vérin axe Z recule à la position 1B7.
- Le vérin axe X recule à la position 1B5.
- Le vérin de magasin recule à la position 1B10 s'il est avancé.
- Le vérin rotatif reprend sa position initiale à la position 1B3.

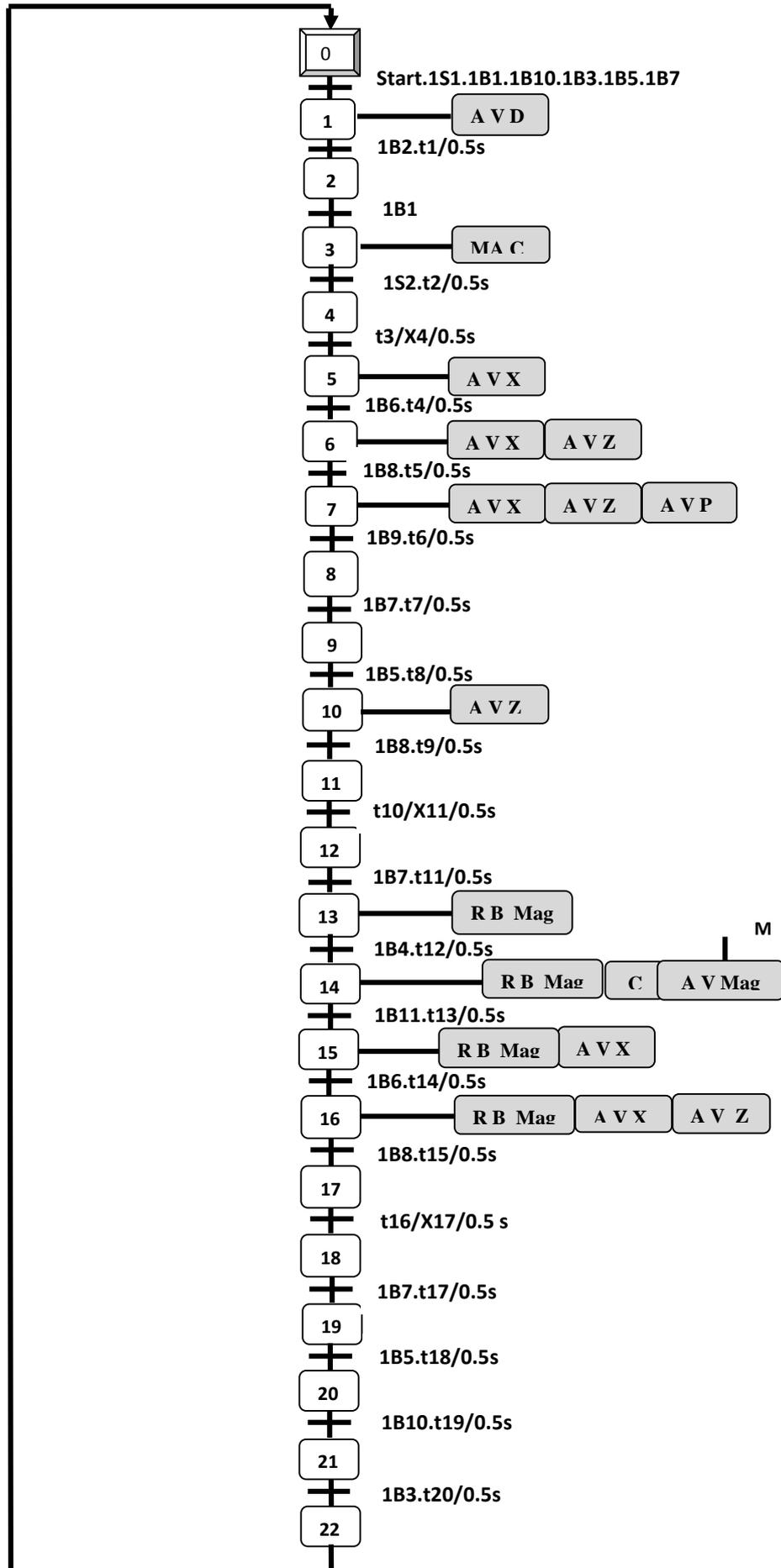
III.2.2 Etablissement du GRAFCET)

GRAFCET
NIVEAU 1



GRAFCE T

NIVEAU 2



Les actions du système sont symbolisées comme suit :

L'action	symbole
Avance vérin distribution	AV D
Marche convoyeur	MA C
Avance vérin axe X	AV X
Avance vérin axe Z	AV Z
Fermeture pince	A V P
Rotation bras ver magasin	R B Mag
Avance vérin magasin	A V Mag

Adressage des entres

transition	Référence	adresse	Désignation
bouton Start	Start	I8.0	Bouton poussoir Start
bouton stop	Stop	I8.1	Bouton poussoir Stop
Vérin distributeur reculé	1B1	I0.0	Capteur ILS
Vérin distributeur avancé	1B2	I0.1	Capteur ILS
Vérin rotatif position Mag	1B3	I0.2	Capteur ILS
Vérin rotatif position Con	1B4	I0.3	Capteur ILS
Vérin axe X avancé	1B5	I0.4	Capteur ILS
Vérin axe X reculé	1B6	I0.5	Capteur ILS
Vérin axe Z avancé	1B7	I0.6	Capteur ILS
Vérin axe Z reculé	1B8	I0.7	Capteur ILS
Pince fermé	1B9	I1.0	Capteur ILS
Vérin magasin reculé	1B10	I1.1	Capteur ILS
Vérin magasin avancé	1B11	I1.2	Capteur ILS
Présence pièce distributeur	1S1	I1.3	Capteur de proximité S1
Présence pièce convoyeur	1S2	I1.4	Capteur de proximité S2
Pièce métallique	1S3	I1.5	Capteur de proximité inductif S3

Adressage des sorties

Adresse	Sortie
Q0.0	Avance vérin distributeur
Q0.1	Rotation bras
Q0.2	Avance vérin axe X
Q0.3	Avance vérin axe Z
Q0.4	Avance vérin pince
Q0.5	Avance vérin magasin
Q0.6	Marche convoyeur

III.2.3 Mise en équations du GRAFCET

Elles sont représenté par le tableau suivant :

L'étape X _n	CA X _n	CD X _n	$X_n = CAX_n + \overline{CDX_n} \cdot X_n$
X _n	T n-1.X n-1	X n+1	$X_n = t_{n-1} \cdot X_{n-1} + \overline{X_{n+1}} \cdot X_n$
X0=M0.0	X22+init	X1.init	$X_0 = X_{22} + X_0 \cdot \overline{X_1} + \text{init}$
X1=M0.1	Start.1S1.1B1.1B10.1B3.1B5.1B7.X0.init	X2+init	$X_1 = (\text{start.1S1.1B1.1B10.1B.1B5.1B7} \cdot X_0 + X_1 \cdot \overline{X_2}) \cdot \text{init}$
X2=M0.2	t1.1B2.X1.init	X3+init	$X_2 = (t_1 \cdot 1B2 \cdot X_1 + X_2 \cdot \overline{X_3}) \cdot \text{init}$
X3=M0.3	1B1.X2.init	X4+init	$X_3 = (1B1 \cdot X_2 + X_3 \cdot \overline{X_4}) \cdot \text{init}$
X4=M0.4	1S2.t2.X3.init	X5+init	$X_4 = (1S2.t_2 \cdot X_3 + X_4 \cdot \overline{X_5}) \cdot \text{init}$
X5=M0.5	t3.X4.init	X6+init	$X_5 = (t_3 \cdot X_4 + X_5 \cdot \overline{X_6}) \cdot \text{init}$
X6=M0.6	1B6.t4.X5.init	X7+init	$X_6 = (1B6.t_4 \cdot X_5 + X_6 \cdot \overline{X_7}) \cdot \text{init}$
X7=M0.7	1B8.t5.X6.init	X8+init	$X_7 = (1B8.t_5 \cdot X_6 + X_7 \cdot \overline{X_8}) \cdot \text{init}$
X8=M1.0	1B9.t6.X7.init	X9+init	$X_8 = (1B9.t_6 \cdot X_7 + X_8 \cdot \overline{X_9}) \cdot \text{init}$
X9=M1.1	1B7.t7.X8.init	X10+init	$X_9 = (1B7.t_7 \cdot X_8 + X_9 \cdot \overline{X_{10}}) \cdot \text{init}$
X10=M1.2	1B5.t8.X9.init	X11+init	$X_{10} = (1B5.t_8 \cdot X_9 + X_{10} \cdot \overline{X_{11}}) \cdot \text{init}$
X11=M1.3	1B8.t9.X10.init	X12+init	$X_{11} = (1B8.t_9 \cdot X_{10} + X_{11} \cdot \overline{X_{12}}) \cdot \text{init}$
X12=M1.4	t10.X11.init	X13+init	$X_{12} = (t_{10} \cdot X_{11} + X_{12} \cdot \overline{X_{13}}) \cdot \text{init}$
X13=M1.5	1B7.t11X12.init	X14+init	$X_{13} = (1B7.t_{11} X_{12} + X_{13} \cdot \overline{X_{14}}) \cdot \text{init}$
X14=M1.6	1B4.t12.X13.init	X15+init	$X_{14} = (1B4.t_{12} \cdot X_{13} + X_{14} \cdot \overline{X_{15}}) \cdot \text{init}$
X15=M1.7	1B11.t13.X14.init	X16+init	$X_{15} = (1B11.t_{13} \cdot X_{14} + X_{15} \cdot \overline{X_{16}}) \cdot \text{init}$
X16=M2.0	1B6.t14.X15.init	X17+init	$X_{16} = (1B6.t_{14} \cdot X_{15} + X_{16} \cdot \overline{X_{17}}) \cdot \text{init}$
X17=M2.1	1B8.t15.X16.init	X18+init	$X_{17} = (1B8.t_{15} \cdot X_{16} + X_{17} \cdot \overline{X_{18}}) \cdot \text{init}$
X18=M2.2	t16.X17.init	X19+init	$X_{18} = (t_{16} \cdot X_{17} + X_{18} \cdot \overline{X_{19}}) \cdot \text{init}$
X19=M2.3	1B7.t17.X18.init	X20+init	$X_{19} = (1B7.t_{17} \cdot X_{18} + X_{19} \cdot \overline{X_{20}}) \cdot \text{init}$
X20=M2.4	1B5.t18.X19.init	X21+init	$X_{20} = (1B5.t_{18} \cdot X_{19} + X_{20} \cdot \overline{X_{21}}) \cdot \text{init}$
X21=M2.5	1B10.t19.X20.init	X22+init	$X_{21} = (1B10.t_{19} \cdot X_{20} + X_{21} \cdot \overline{X_{22}}) \cdot \text{init}$
X22=M2.6	1B3.t20.X21.init	X0+init	$X_{22} = (1B3.t_{20} \cdot X_{21} + X_{22} \cdot \overline{X_0}) \cdot \text{init}$

III.3 Réalisation du programme (phase de programmation)

Pour l'établissement du programme on doit passer par les étapes suivantes.

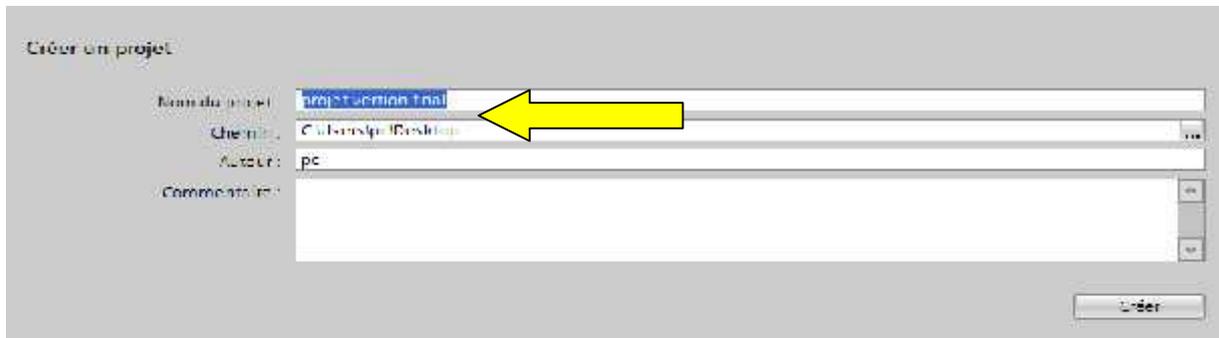
1/ créé un projet dans le logiciel Tia portal et en commence par l'accès au logiciel Tia portal



Figure III.4 icone du logiciel TIA portal

Avec un double cliqué sur l'onglet installé dans le bureau du PC

Dans la fenêtre vue du portal l'étape suivante c'est de créer un projet et qu'il soit « projet version final ».

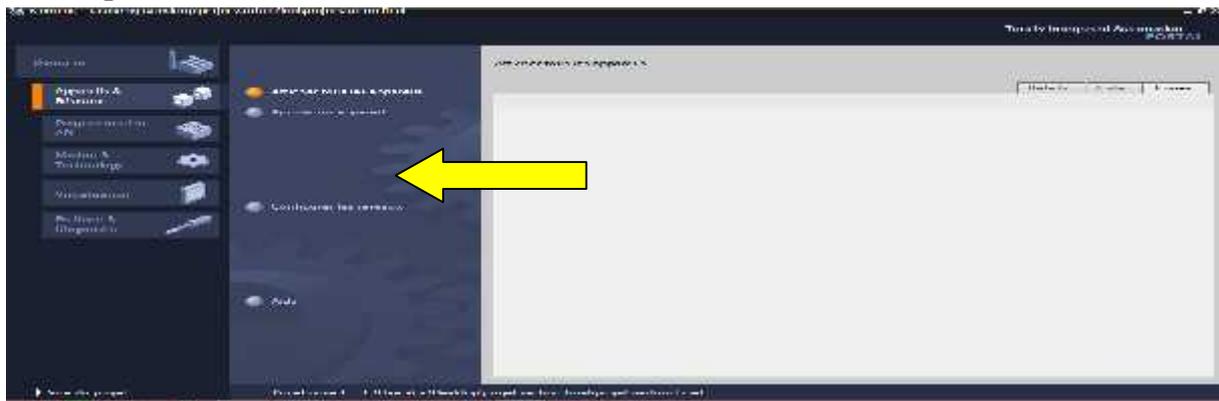


Le nom du projet s'affiche dans la fenêtre vue de portal



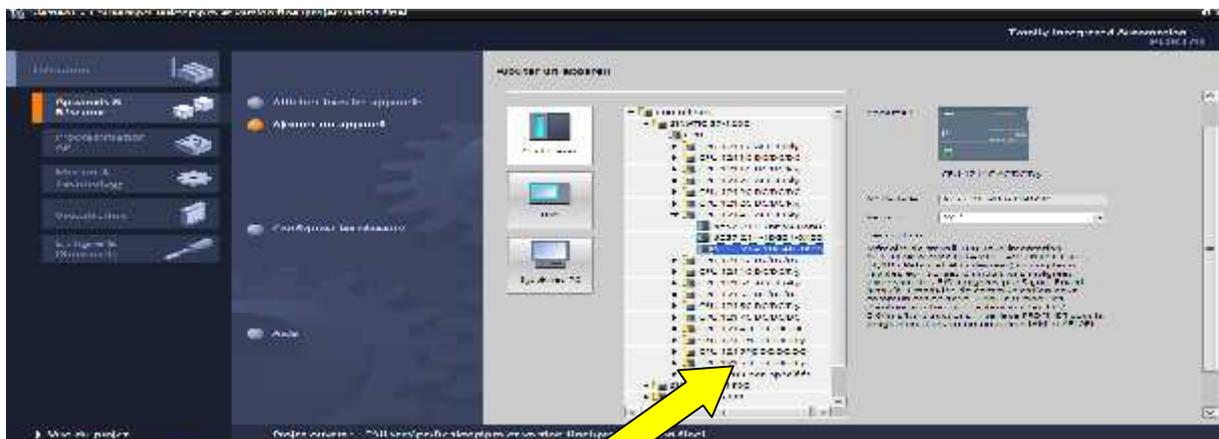
2/ En suite on passe à la configuration d'un appareil (automate).

Un clique sur « configurer un appareil ».



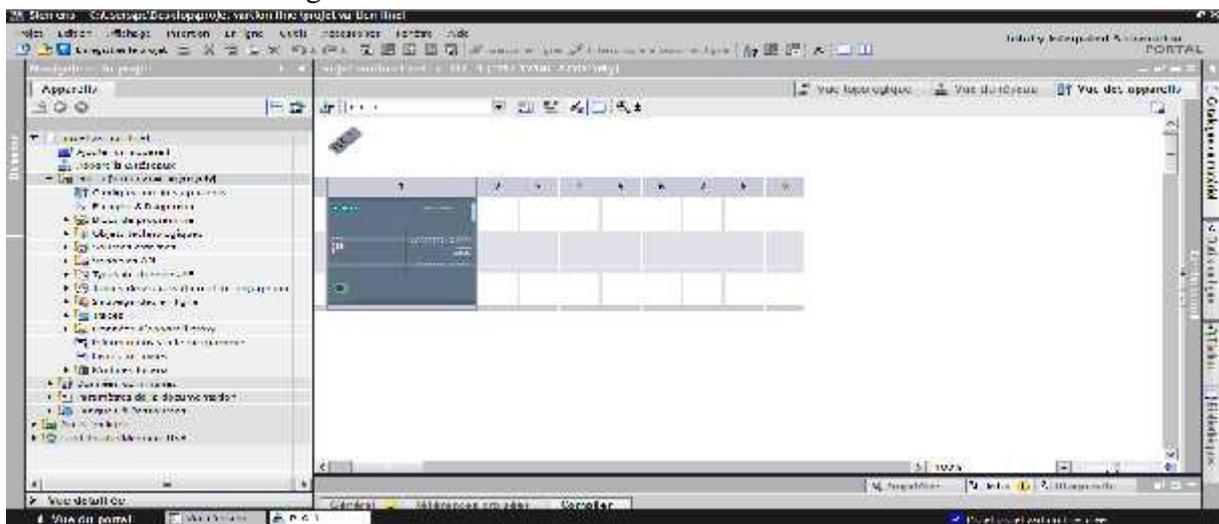
3/ Puis, un clique sur ajouter un appareil (pour collectionné l'appareil désirer)

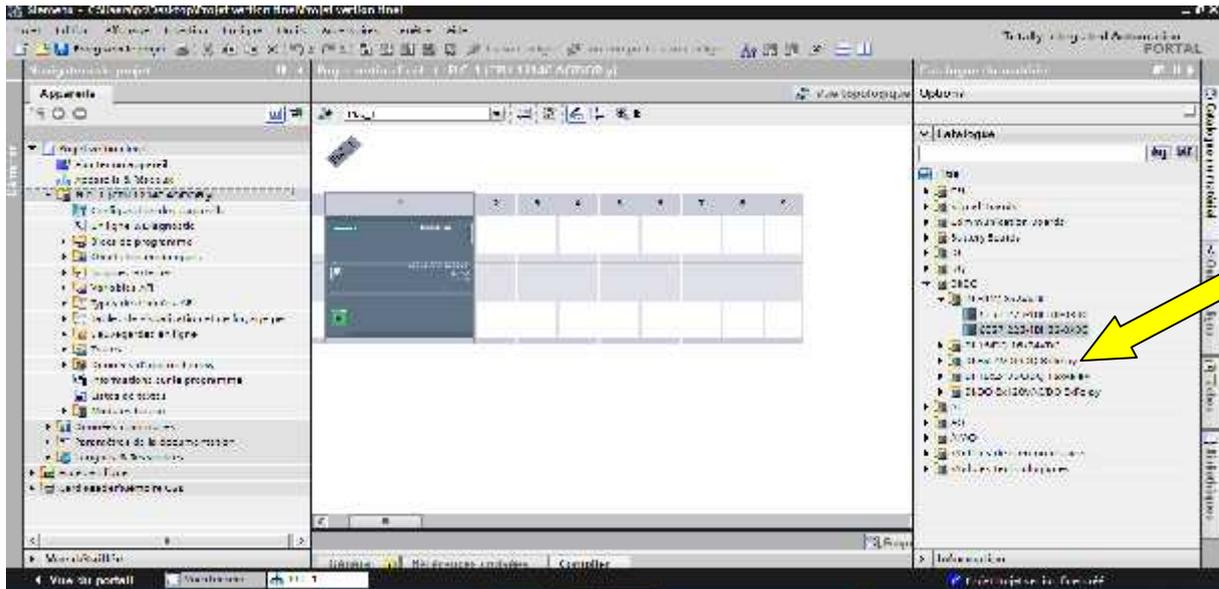
On clique sur contrôleur puis s7 1200 puis CPU et enfin choisir la série de l'automate soit CPU 1214 AC/DC RELAY.



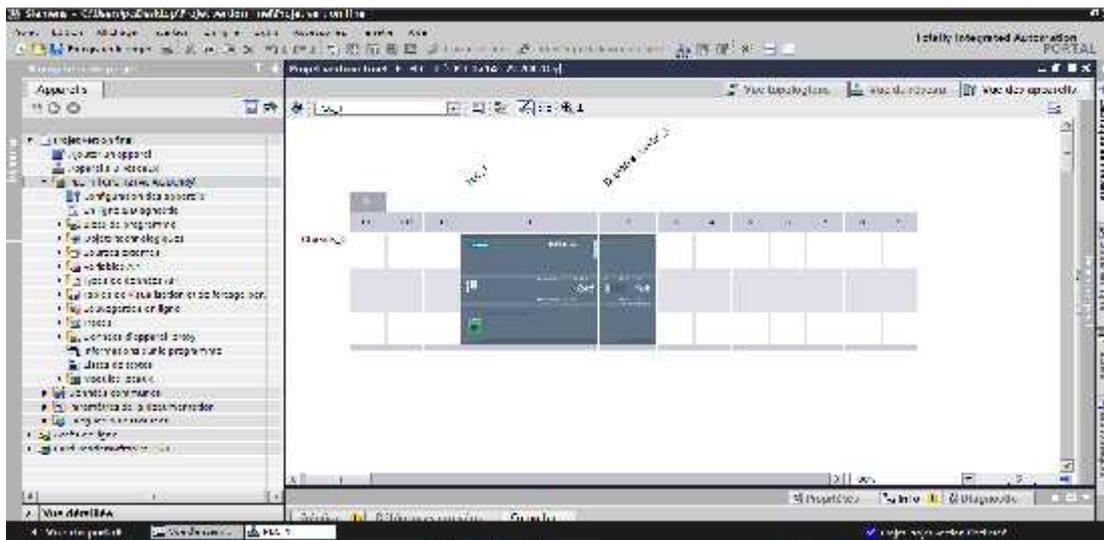
Une fois que l'appareil est afficher da la vue de projet.

4/ On ajoute le module des entres/ sorties selon no besoins DI/DQ on cherchant le module dans la fenêtre « catalogue du matériels »



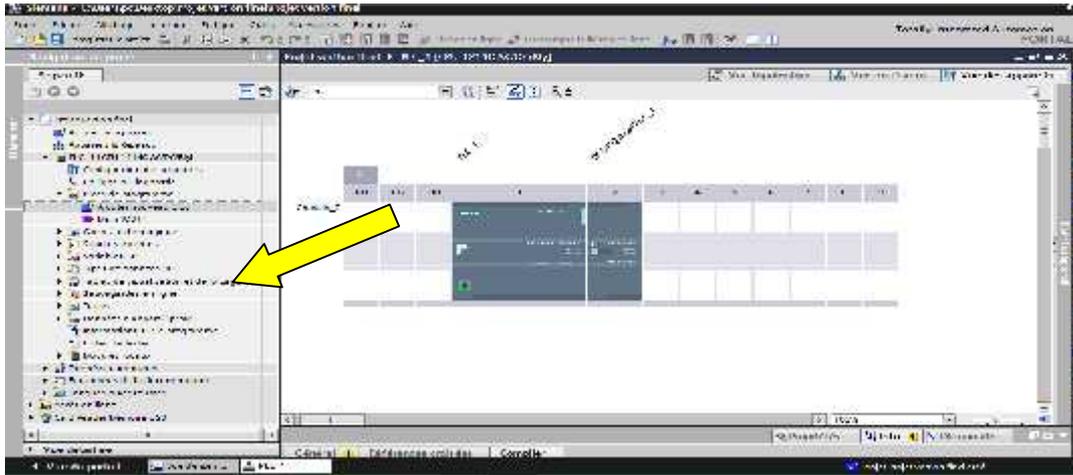


On obtient la nouvelle configuration du projet avec le plc1 et le module d'extension DI/DQ des entres sortie.

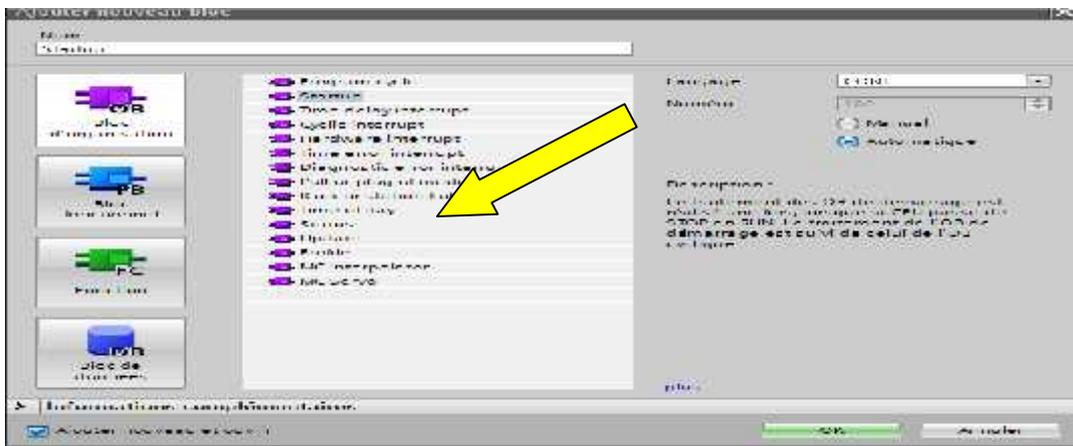


5/ Après la création du projet et la configuration des appareils commence l'étape écriture du programme.

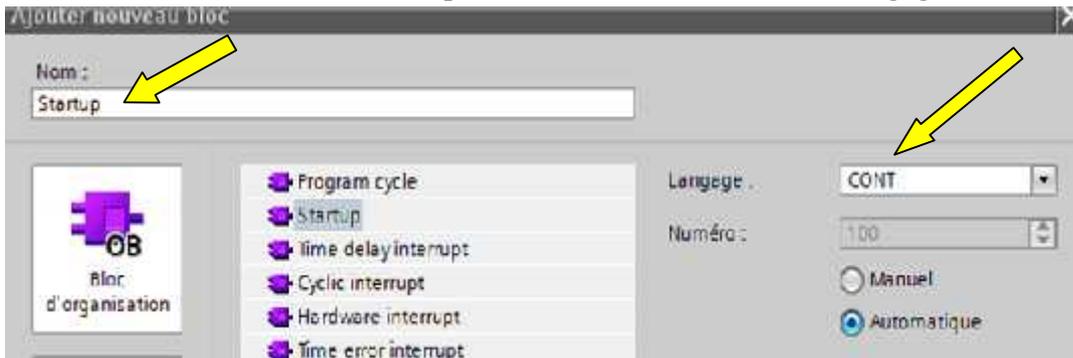
Dans l'arborescence du menu «Navigateur du projet » un cliqué sur« Blocs de programme »



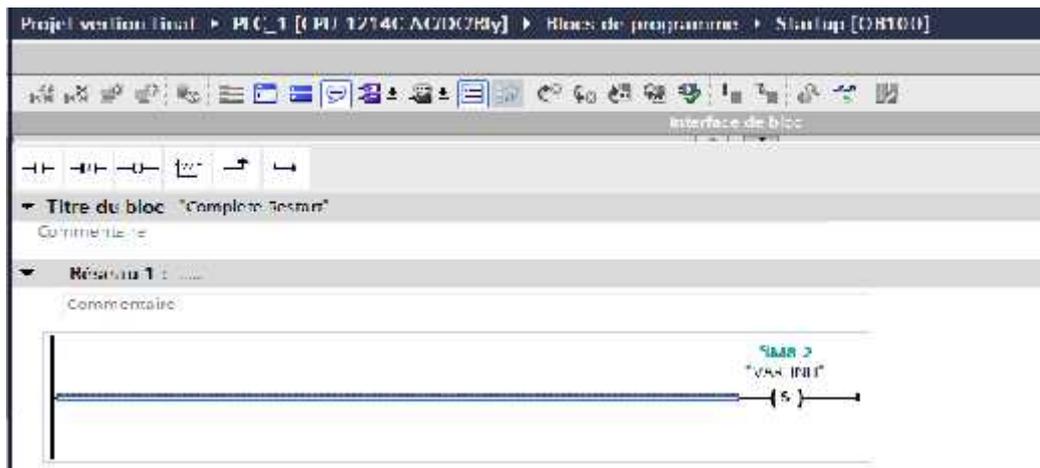
6/ Pour la création des blocs spécifiques on clique sur « Ajouter nouveau bloc » et on obtient la fenêtre suivante.



Dans cette fenêtre le bloc « Start up » est sélectionné avec un « Langage contact ».

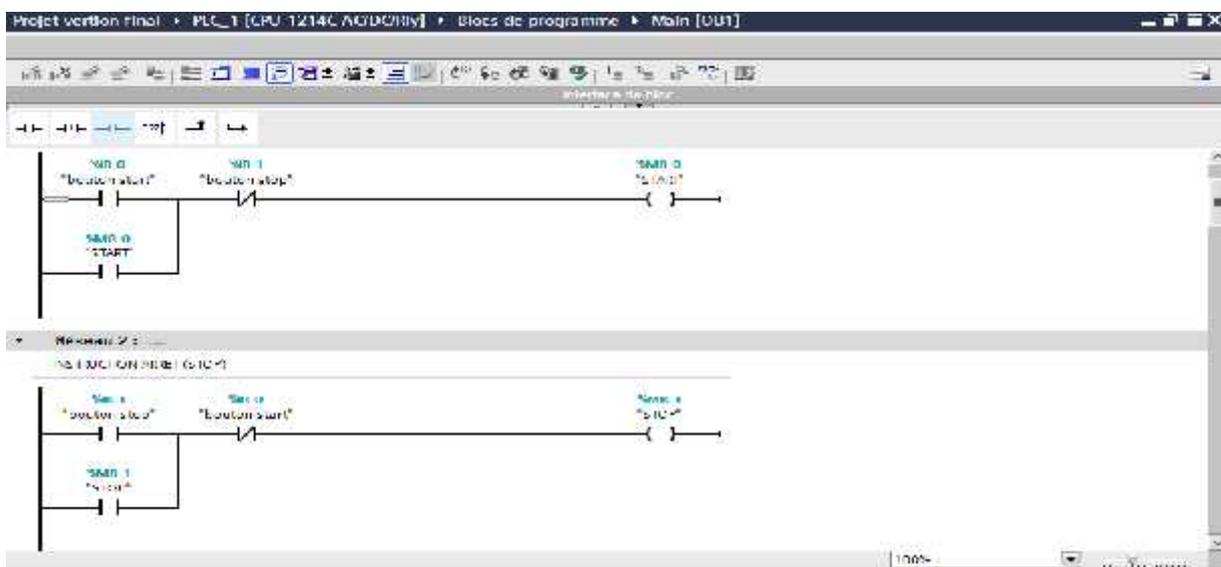


Le bloc OB100 contient le réseau de la variable initiation VAR_INIT il est choisi comme variable mémoire et elle peut être comme variable input un bouton poussoir par exemple Il est maintenu dans l'état 1



7/ Et pour le programme principale il est réalisé dans le bloc « MAIN OB1 »

Il est débute par les instructions START et STOP pour commander le système par les boutons Start et stop



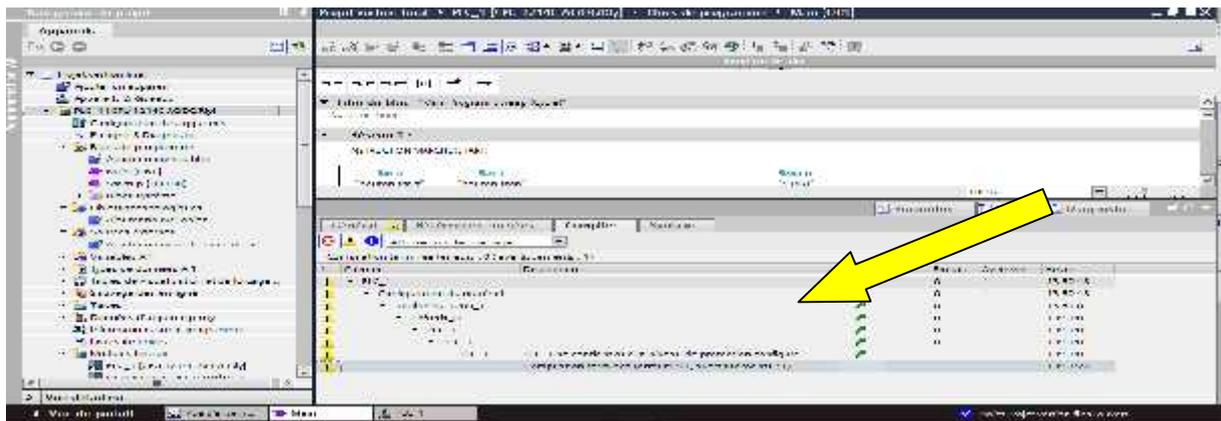
Compilation et chargement du programme dans la CPU

- **Compilation**

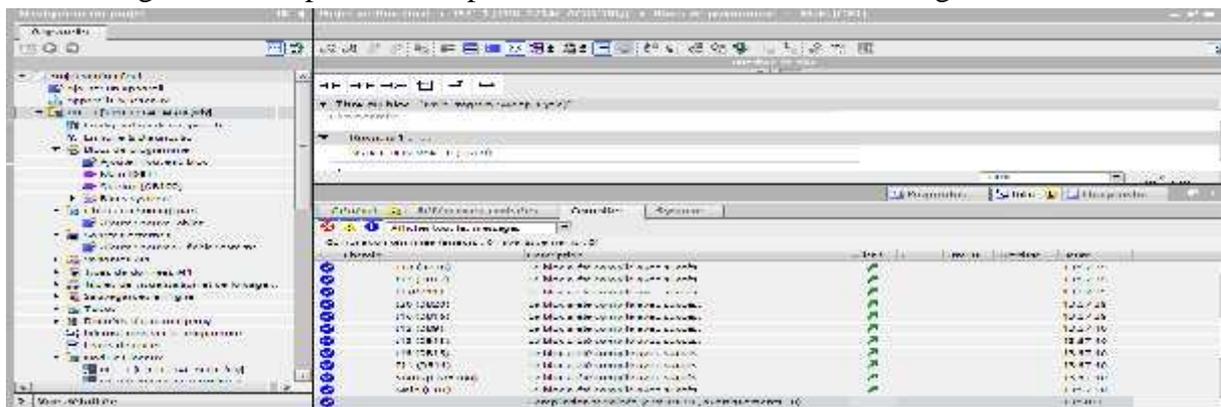
Pour que le programme soit exécutable on doit le transforme du contact en langage machine

La compilation assure la vérification du matériel ainsi que les blocs du programme c'est pour ca elle est faite sur deux niveaux.

-Niveau matériel « compilation complète du matériel » et affiche le message si dessous



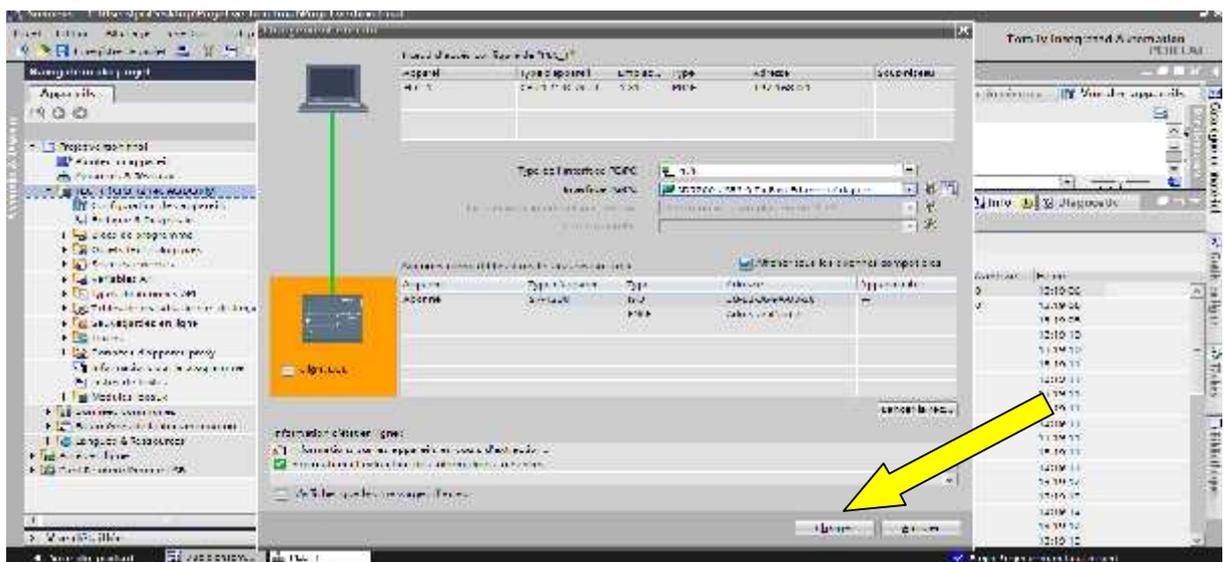
-Niveau logiciel « compilation complète des blocs » et affiche le message si dessous



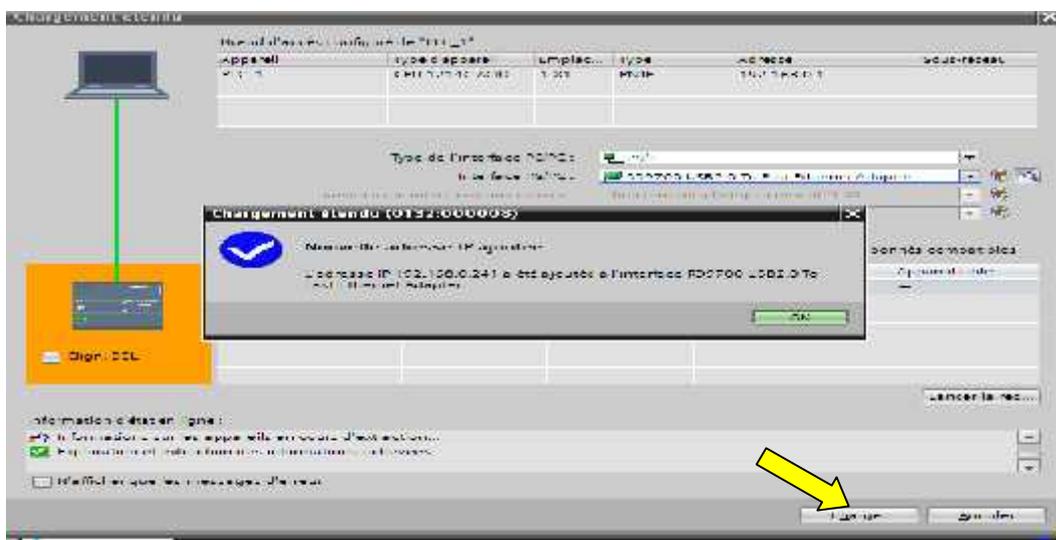
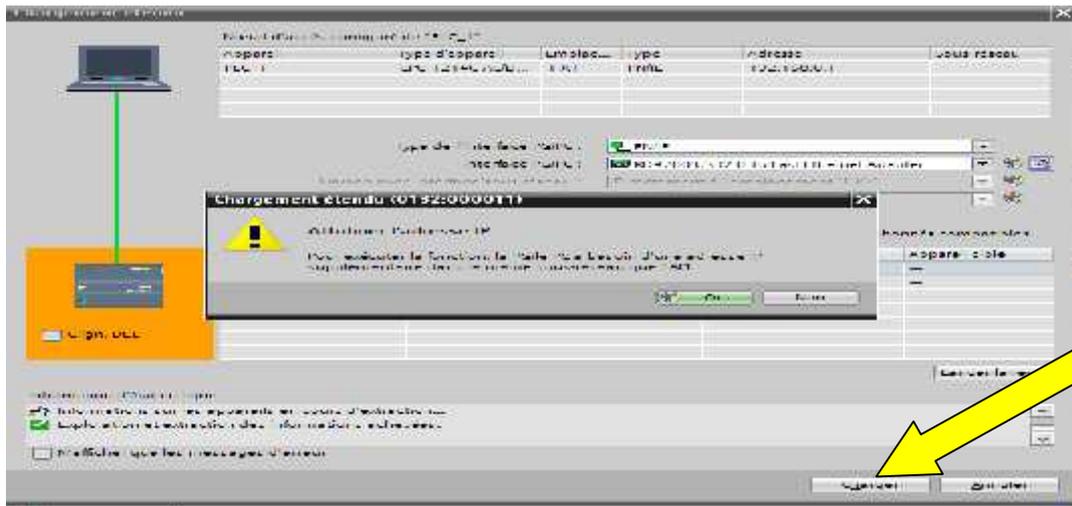
L'information (0 erreur ,0 avertissement) indique que le programme est juste.

- **Chargement**

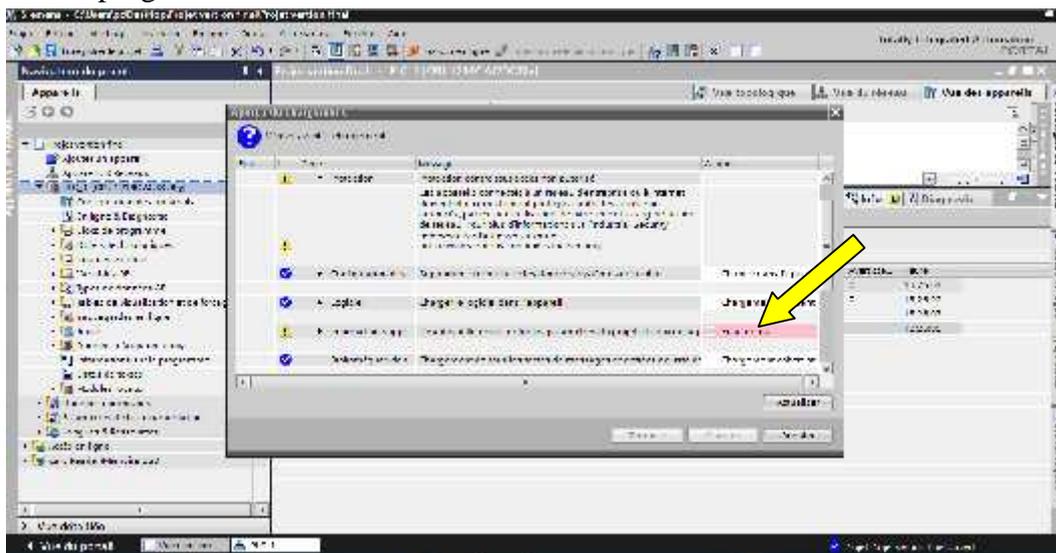
Le chargement c'est le transfert du programme du PC au processeur de l'automate afin de l'exécuter pour cela on doit établir une liaison parfaite entre logiciel le et l'automate CPU. C'est adire : les deux périphériques doivent avoir la même adresse IP.



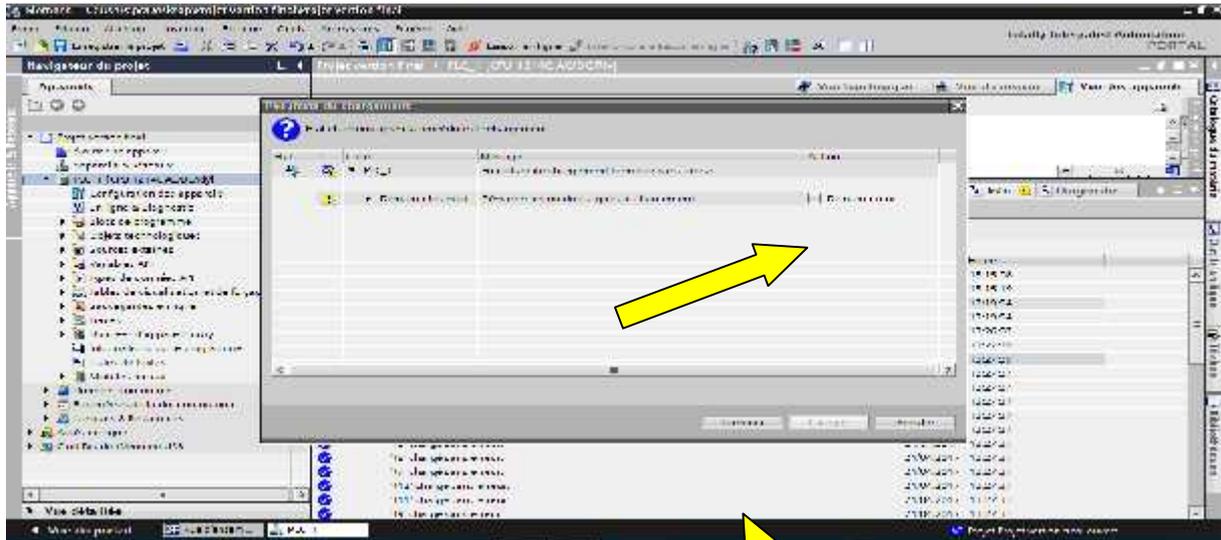
Si l'opération est faite avec succès, on clique sur charger.



Après que la nouvelle IP est ajoutée, le logiciel nous donne la main d'écarter tout programme installé dans l'automate.



La dernière étape du chargement.



On clique sur « **Terminer** ».

On appuis sur le bouton **Start** et le système commence à fonctionner. (Voir la vidéo).

- Pour nous permettre de contrôler et vérifier l'exécution de ce programme il est on a décidé de le partager en trois sous programme sous forme de testes.

Test n° 1
Distribution de la pièce

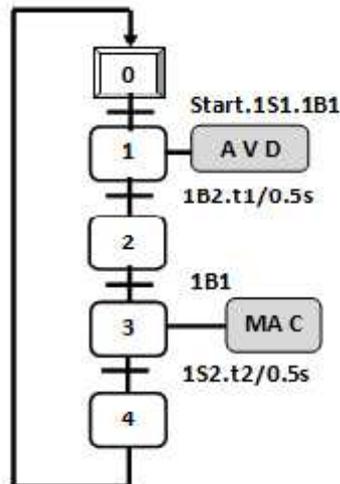
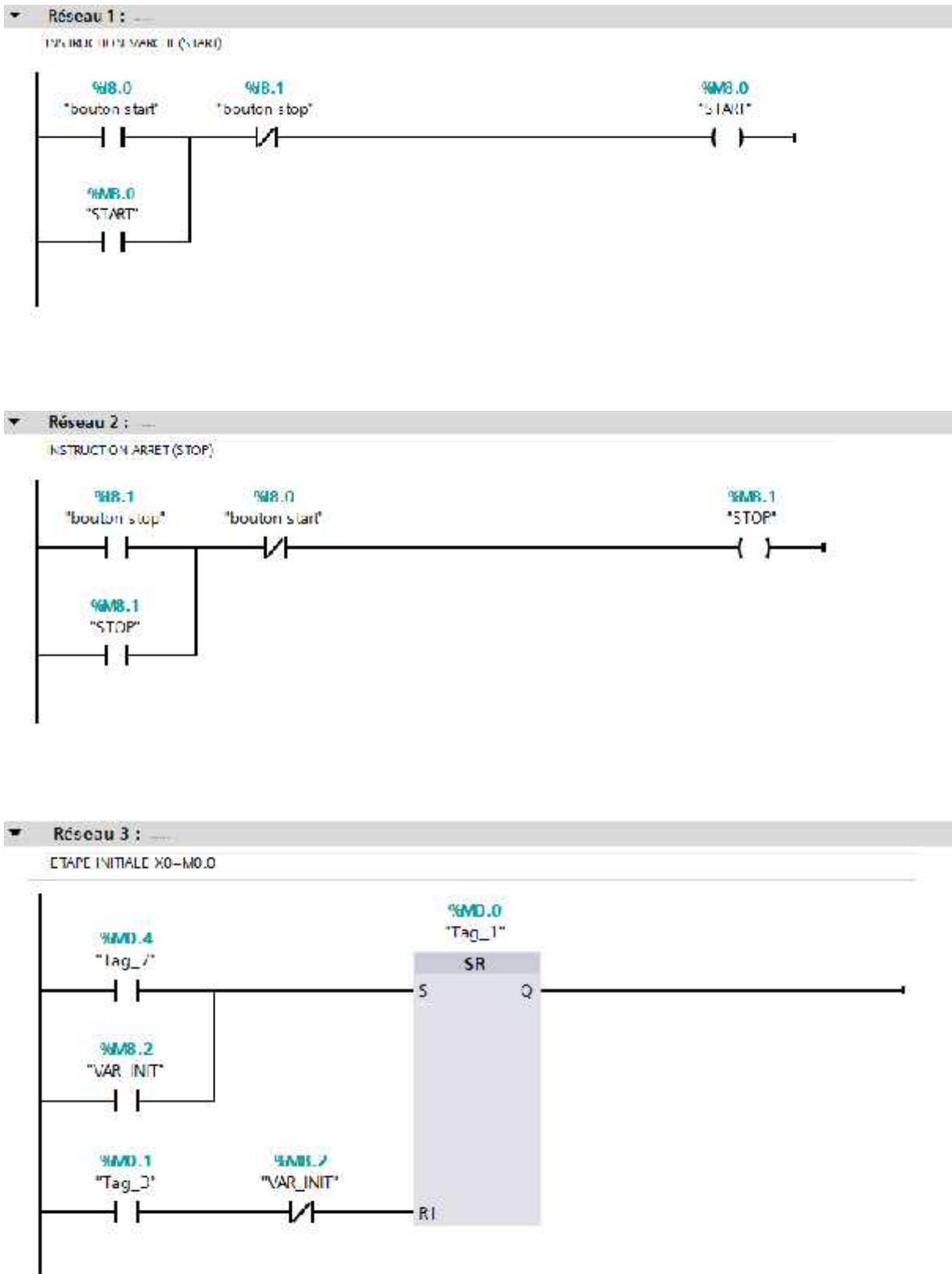


Tableau des équations d'états

Xn	CAXn	CDXn	$Xn = CA.Xn + CDXn.Xn$
X0	$1S2.t2.X3 + \text{init}$	$X1.\text{init}$	$X0 = 1S2.X3 + X1.X0 + \text{init}$
X1	$1S1.1B1.\text{start}.X0.\text{init}$	$X2 + \text{init}$	$X1 = (1S1.1B1.\text{Start}.X0 + X2.X1).\text{init}$
X2	$t1.1B2.X1.\text{init}$	$X3 + \text{init}$	$X2 = (1B2.X1 + X3.X2).\text{init}$
X3	$1B1.X2.\text{init}$	$X4 + \text{init}$	$X3 = (1B1.X2 + X4.X3).\text{init}$
X4	$1S2.t2.X3.\text{init}$	$X0 + \text{init}$	$X4 = (1S2.t2.X3 + X4.X0).\text{init}$

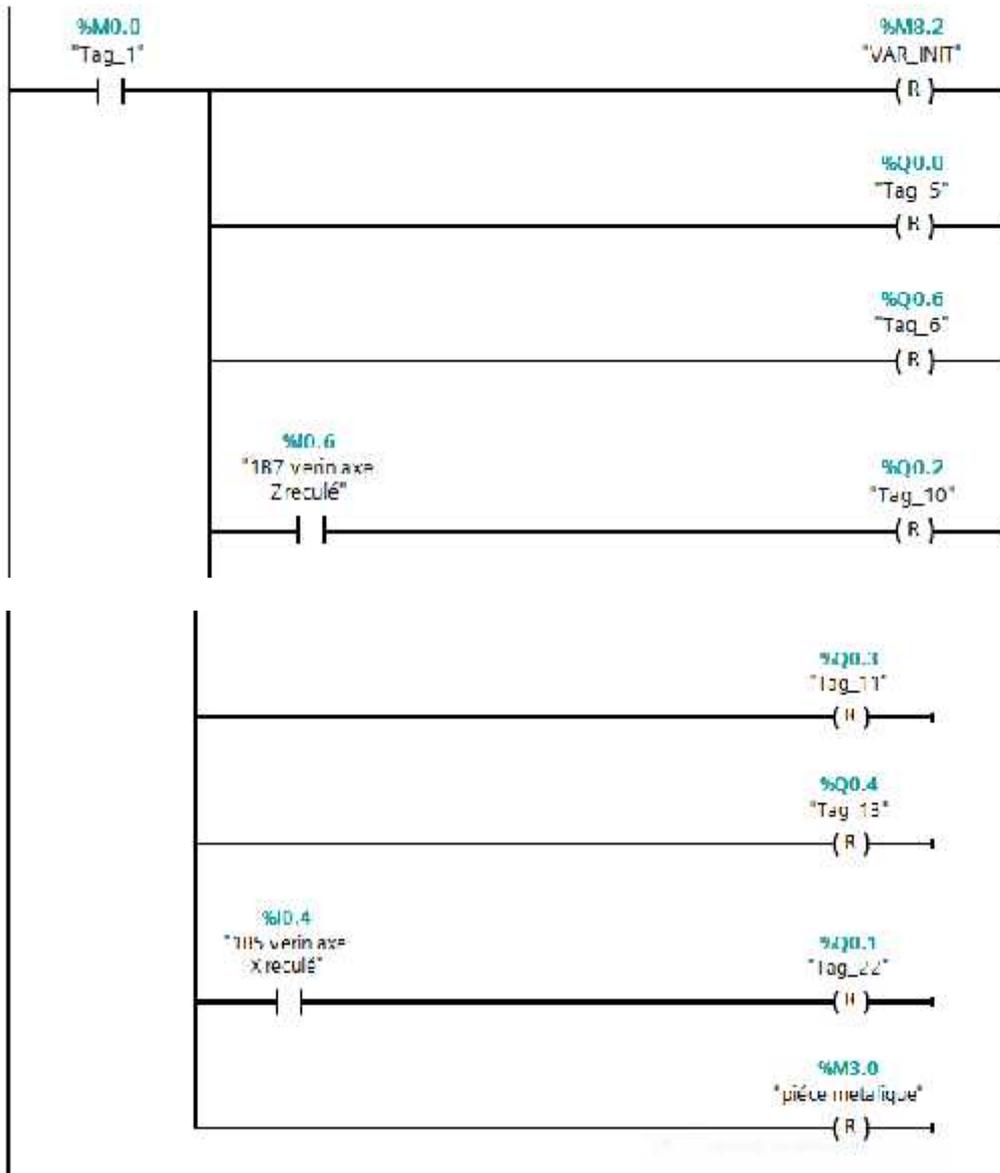
III.4 Le Programme

Le programme en langage contact est comme suit :



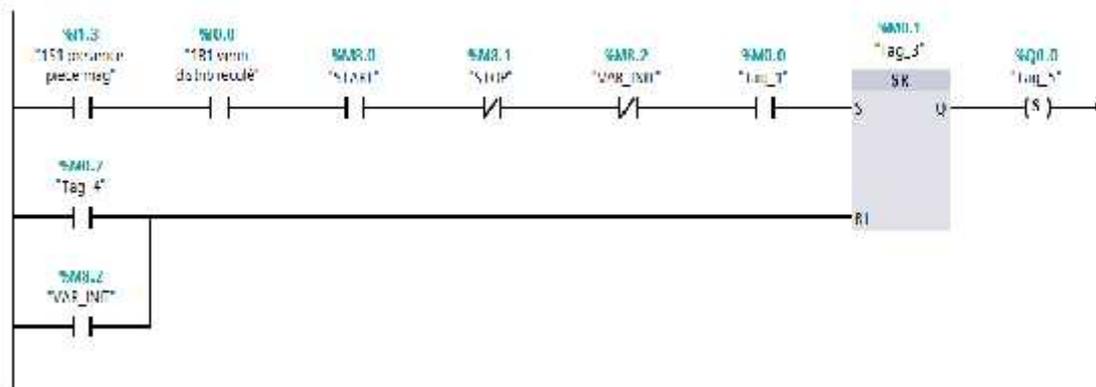
Réseau 4 :

LES ACTION DE L'ETAPE X0



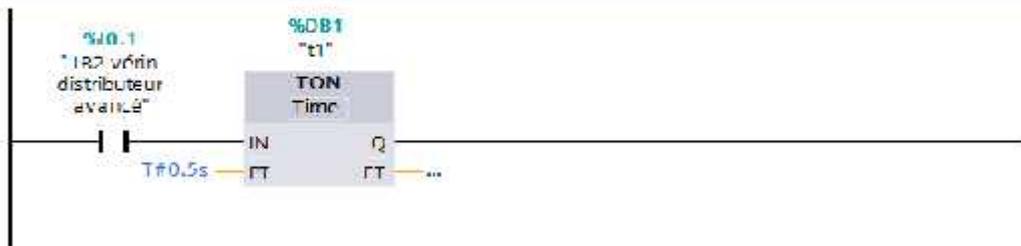
Réseau 5 :

ETAPE 1 AVANCE D STRIP...TF...R XI-U01



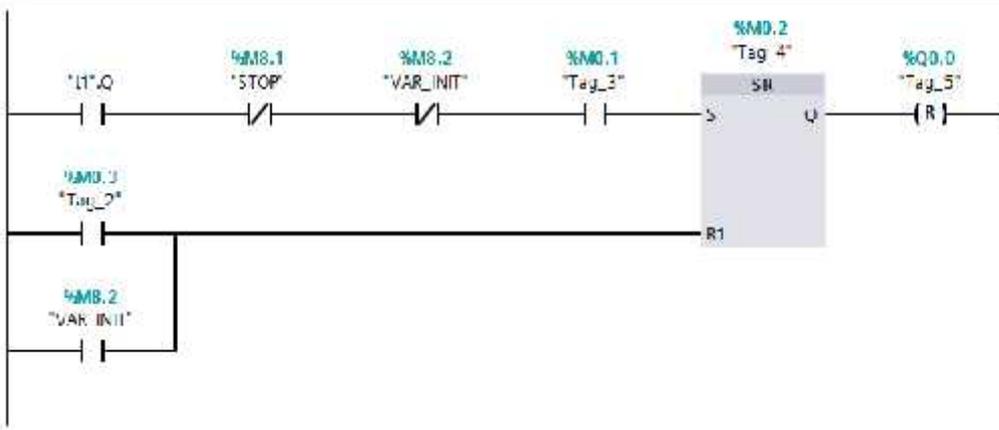
Réseau 6 :

Temporisation de recule (etendue)



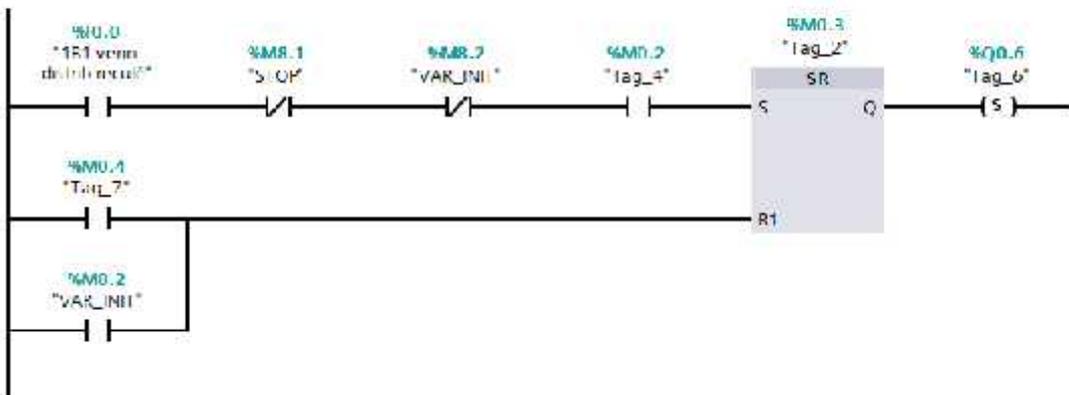
Réseau 7 :

ETAPE 2 RECULE DISTRIBUTEUR X2-M0.2



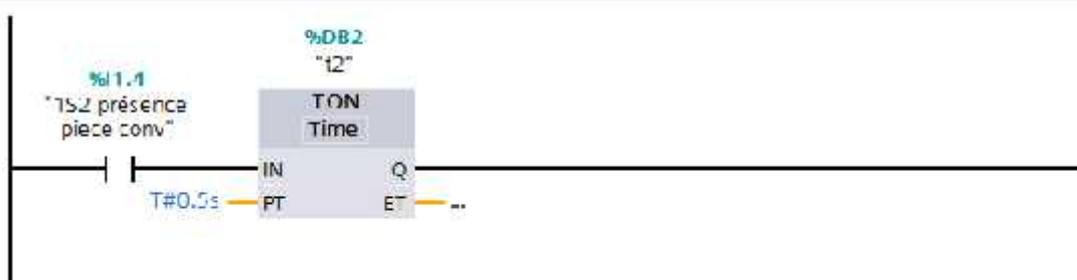
Réseau 8 :

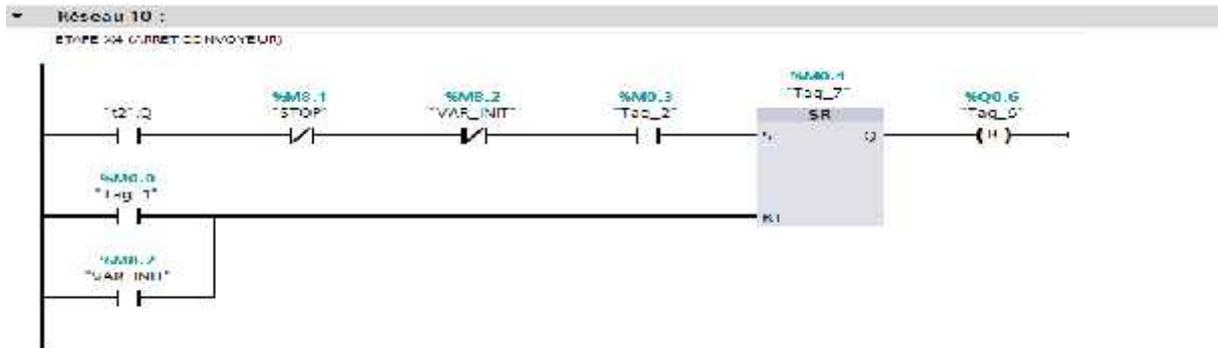
L'ARRET MARCHE CONVOYEUR



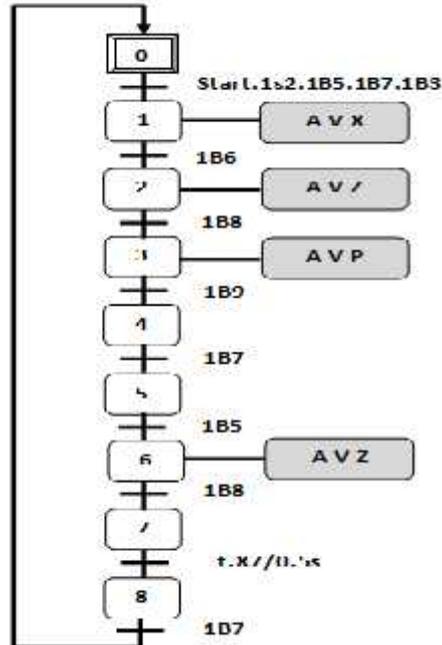
Réseau 9 :

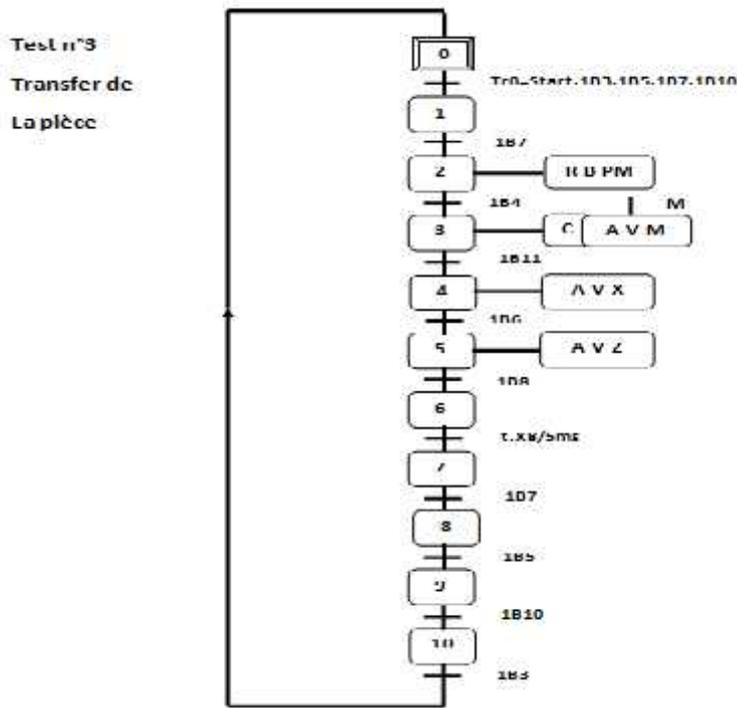
TEMPORISATION DE L'ARRET CONVOYEUR





Test n°2
Triage de la pièce





III.5 La supervision du processus avec SIMATIC Win CC

La supervision permet de visualiser et contrôler le système à distance.

Le superviseur peut être de type PC ou unité de visualisation spécialisée IHM.

Elle Permet la télésurveillance, le télédiagnostic, la télémaintenance à longue distance.

Définition

Win CC est le logiciel phare de Siemens permettant de créer des interfaces homme-machine sur pupitre tactile (IHM) ou sur écran. La dernière version de Win CC est intégrée à TIA Portal.

.Pour réalise cette tache on doit suivre les étapes suivantes :

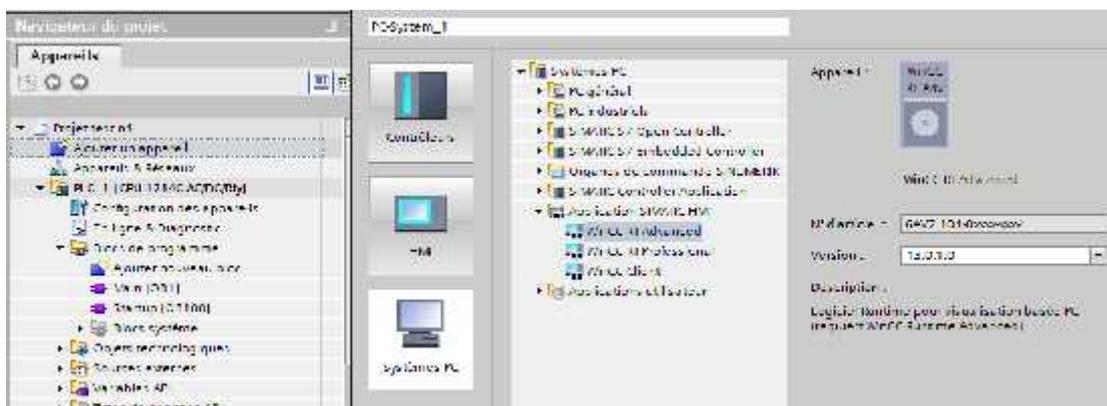
1- Création de l'interface homme machine

Plusieurs chemins nous permet de crée cette interface on prend la suivante :

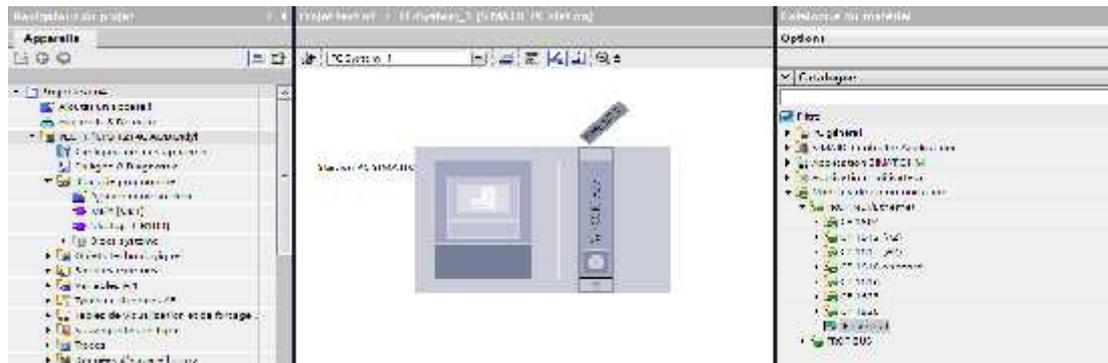
Dans la vue du navigateur du projet, un double clic sur « **ajouter un appareil** »

On sélectionne « systèmes PC », puis application « SIMATIC HMI » et on choisi

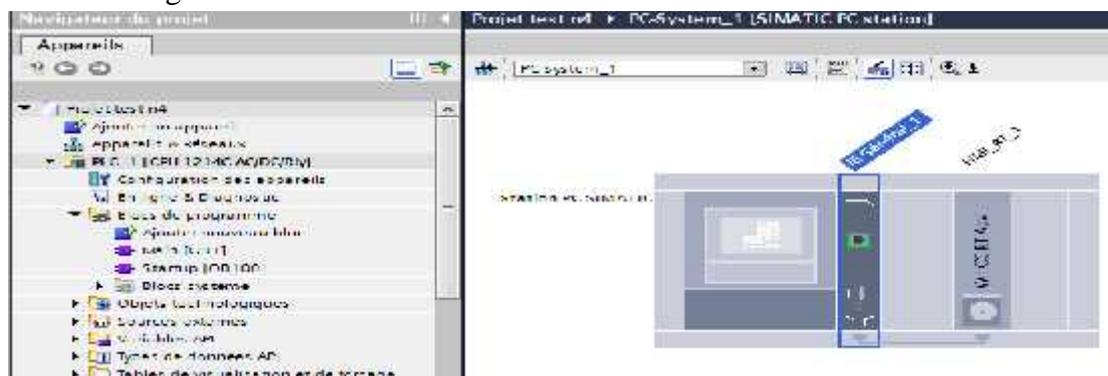
« Win CC RT Advanced ».



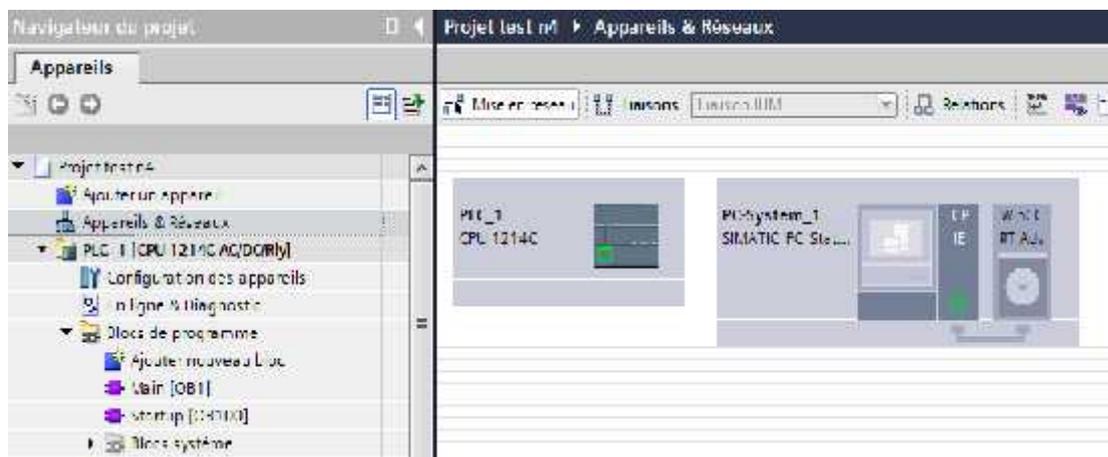
Puis la création du module de communication de cette interface avec l'automate. Dans la fenêtre « catalogue du matériels » on click sur « module de communication », puis « IE General »

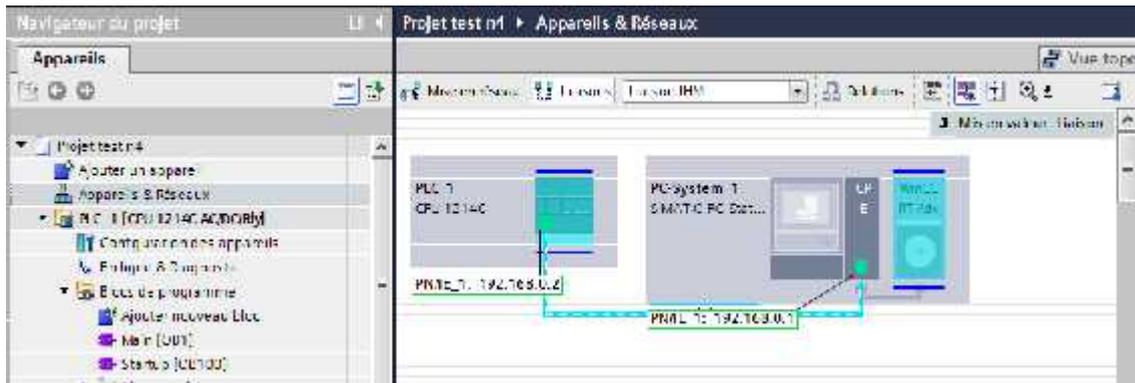


On aura la figure suivante



Après, dans la vue appareil et réseaux nous allons relier l'automate avec le module de communication du PC système.





On remarque que les deux appareils ont la même adresse IP on dit qu'ils sont sur le même réseau.

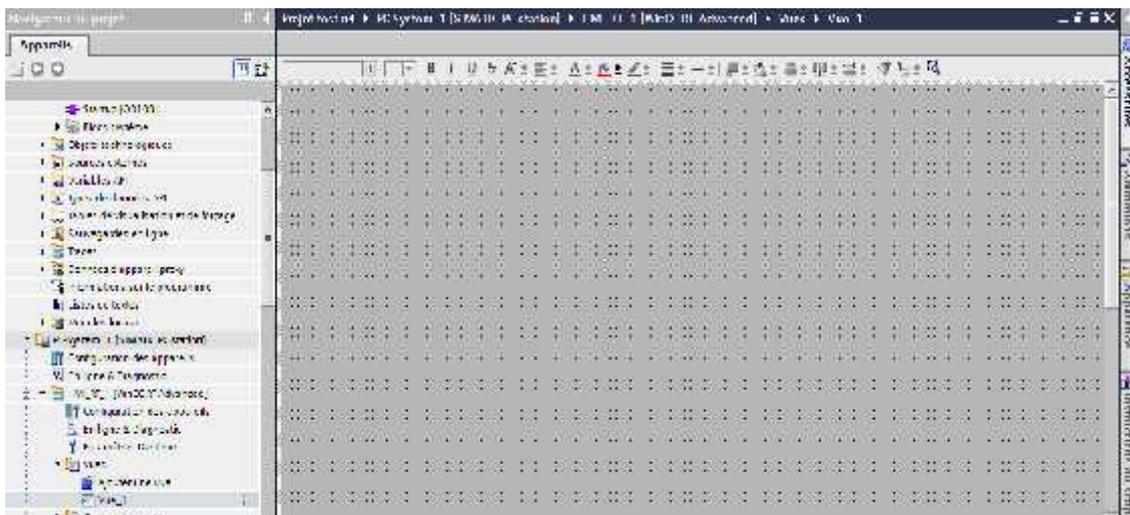
Après le câblage des deux interfaces, nous passons à la programmation des variables de commande tactile et les variables visuelles du système.

Premièrement nous devons ajouter ou créer la vue où on installe nos variables

Donc c'est la création de la vue tactile dans le PC system.

On clique sur « ajouter une vue » à la ligne « PC system ». On obtient la

« Vue 1 » mais avant de créer les variables tactiles (Win CC) on doit faire un changement dans le programme précédent. On doit changer les variables de commande d'être par des variables tactiles que nous allons déclarer ultérieurement dans la vue tactile



III.6 Câblage de l'automate

Les interfaces entrées/sorties tous ou rien sont câblés comme indique le schéma suivant :

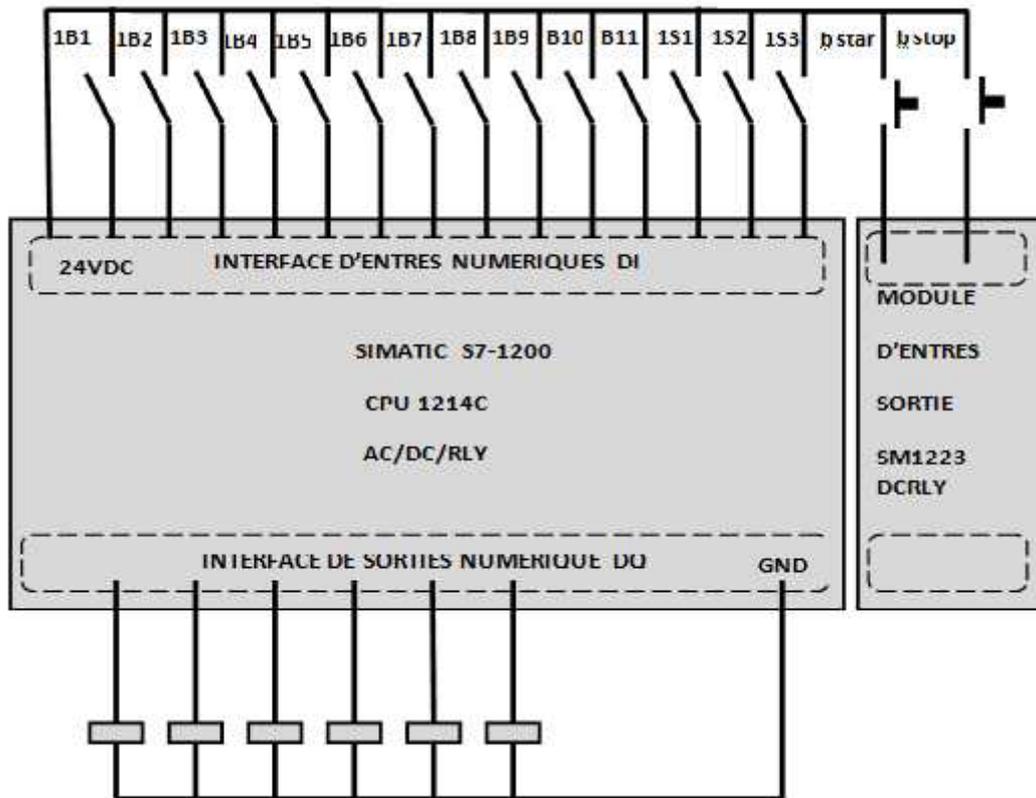


Figure III.4 câblage de l'automate

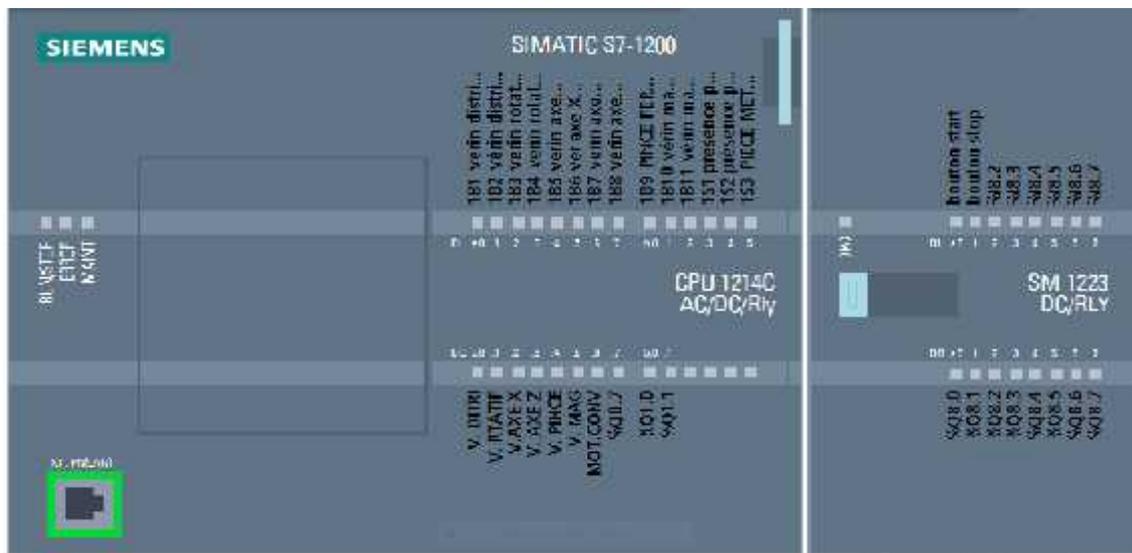


Figure III.5 vérification des entres /sorties sur automate

III.7 Conclusion

Dans ce chapitre on a réussi de rédiger un cahier de charge décrivant le fonctionnement de notre système et établir un GRAFCET convenable en appliquant tout les connaissances théoriques acquises lors de mes études. Il a été la démarche d'écrire un programme en ladder en exploitant le logiciel Tia portal pour charger ce programme dans un automate siemens s7-1200 afin d'exécuter toutes les actions qui sont déclarer dans notre GRAFCET et appliqué des modifications qu'on a vue nécessaires.

Conclusion Générale

Ce projet représente une étude d'une chaîne fonctionnelle automatisée. Cette chaîne fonctionnelle fait la distribution des pièces ensuite le triage et finalement le stockage dans un magasin déplaçable, le système est commandé par un automate programmable SIEMENS S7-1200, où cette dernière contrôle le robot pneumatique trainer-TK.

A travers notre projet de fin d'étude master a réussi de rédiger un cahier de charge décrivant le fonctionnement de notre système et établir un GRAFCET convenable en appliquant toutes les connaissances théoriques acquises lors de mes études. Il a été la démarche d'écrire un programme en ladder en exploitant le logiciel Tia portal pour charger ce programme dans un automate SIEMENS S7-1200 afin d'exécuter toutes les actions qui sont déclarées dans notre GRAFCET et appliqué des modifications qu'on a vu nécessaires.

Ce travail m'a permis d'apprendre les règles et techniques sur la modélisation graphique par le GRAFCET des systèmes industriels, et m'a rendu capable de bien maîtriser l'outil de programmation et de faire le diagnostic et les modifications nécessaires qui assurent un bon fonctionnement des systèmes de commandes par des automates avec les logiciels correspondants pour chaque constructeur.

Web bibliographie

- [1] Systemes automatise
<http://www.clg-daudet-carpentras.ac-aix-marseille.fr/spip/IMG/file/techno/niv3/.pdf>
- [2] Techno- logique système automatise (automatisme)
<http://www.techno-logique.com/AUT-systemes-automatiques.shtml>
- [3] Les vérins pneumatiques
http://www.cuvelier-ludovic.fr/docs/mei/pneu/05_Le%20v%C3%A9rin.pdf
- [4] Guide pneumatique
http://www.difluid.fr/documents/guide_pneumatique.pdf
- [5] Machines.pdf
http://lgt.garnier.free.fr/espace_ee_fichiers/machine_fichiers/machines.pdf
- [6] Organisation fonctionnelle d'une chaîne d'information.pdf
http://www.editions-ellipses.fr/PDF/9782340016743_extrait.pdf
- [7] Automate programmable
<http://www.institut-numerique.org/>
- [8] Les automates industriels programmables
<http://pamelard.electro.pagesperso-orange.fr/fichier%20pdf/automatisme/>
- [9] Signal analogique
https://fr.wikimini.org/wiki/Signal_analogique
- [10] Langages de programmation
- [11] GRAFCET - les normes du GRAFCET - AII GRAFCET - robert CIREDU
<http://robert.ciredu.free.fr/Ressources/AII/Rappels%20sur%20le%20GR>
- [12] Cours de GRAFCET les notions de bases
<http://go.microsoft.com/fwlink/?LinkId=69157>
- [13] Initiation au programme du SIMATIC S7-1200
http://w5.siemens.com/france/web/fr/ad/secteur/education_nationale/Docum
- [14] Tia portal
http://pierrard.be/cta/file_download/79/CTIA01+++Programmation+des+
- [15] PTP Pneumatic robot trainer (FPLC-PTP-ROB-TK)
<http://www.Festech.co.kr>