



Mémoire de fin d'étude

Pour l'obtention du diplôme d'Ingénieur

Filière : Automatique
Spécialité : Automatique

Intitulé :

**Développement d'un système variant
MPS sous L'outil Step7**

Présenté par :

Mr. RETIMI MOHAMED BACHIR

Mr. BENSID MOHAMED WALID

Le Jury :

| | | |
|-----------------------------------|---|--------------|
| Mr. GHOMRI Latéfa. | MCB | Président |
| Mr. HOUARI HABIBA. | MAA | Examineur |
| Mr. HASSAM Ahmed. | MCB | Examineur |
| Mr. MALIKI Fouad. | MCB | Encadrant |
| Mr. MKEDDER M ^{ed} Amin. | Ingénieur Principal en Recherche et Développement | Co-encadrant |

Année Universitaire : 2020- 2021

Remerciement



Louanges à Dieu le tout puissant, grâce à qui nous avons pu faire tout ce travail.

Nos remerciements s'adressent à toutes les personnes qui, d'une manière ou d'une autre nous ont permis de mener à son terme ce mémoire et plus particulièrement :

A nos encadrant, Mr MALIKI Fouad et Mr. MKEDDER Med EL Amin qui nous ont grandement aidées à définir avec circonspection le sujet de notre travail, et qui par leurs conseils et idées, nous ont permis d'améliorer la qualité du contenu. Leurs aides techniques ont été primordiaux pour avancer dans notre projet.

Comme on n'oublie pas Mr. MKEDDER Med EL Amin pour ses précieux conseils et remarques qui nous ont beaucoup aidés, et également pour son soutien moral tout au long de notre cursus vraiment c'était la personne la plus proche de nous.

Nos vifs remerciements vont aux membres de Jury, pour avoir accepté de juger notre travail.

A nos enseignants de l'ESSAT, à qui nous devons notre formation, qu'ils trouvent ici l'expression de nos sentiments les plus respectueux et de notre profonde gratitude pour leur gentillesse, leur disponibilité et leur soutien.

A nos chers parents, nos frères et sœurs et nos familles qui nous ont toujours soutenus le long de notre formation

A tous nos amis, pour leur soutien moral et aide précieuse.

Dédicaces



*La vie n'est qu'un éclair,
Et un jour de réussite est un jour très cher.*

Je dédie ce mémoire

A mon cher père

Et ma très chère mère

*Pour l'éducation et le grand amour dont ils m'ont entouré
depuis ma naissance.*

Et pour leur patience et leurs sacrifices.

*A mon grand-père Rétimi BACHIR, ceci est ma profonde
gratitude pour ton éternel amour, que ce travail soit
le meilleur cadeau que je puisse t'offrir.*

A mes chers frères ;

A mes chères sœurs ;

*A tous mes proches : mes tantes, mes cousines
Mohamed, Afdal, Kadi et Et surtout Mr : Amine.Mk
qui m'a aidé pendant mon travail, mes voisins et toute la
famille Rétimi et Hanichi....*

A tous ceux qui m'aiment ;

A tous mes ami (e) s ;

A tous ceux que j'aime ... ;

A tous mes collègues de promo

Automatique (2018/2021) ;

.....M^{ed} Bachir

Dédicaces



*La vie n'est qu'un éclair,
Et un jour de réussite est un jour très cher.*

Je dédie ce mémoire

A mon cher père

Et ma très chère mère

*Pour l'éducation et le grand amour dont ils m'ont entouré
depuis ma naissance.*

Et pour leur patience et leurs sacrifices.

A mes chers frères ;

A mes chères sœurs ;

A tous mes proches : mes tantes, mes cousines

*Et surtout Mr : Amine.Mk qui m'a aidé pendant mon
travail, mes voisins et toute la famille BENSID....*

A tous ceux qui m'aiment ;

A tous mes ami (e) s ;

A tous ceux que j'aime ... ;

A tous mes collègues de promo

Automatique (2018/2021) ;

.....Med Walid

GLOSSAIRE

Acronymes

API Automate Programmable Industriel

BCD Binary Coded Decimal

CAN convertisseur analogique-numérique

CPU Central Processing Unit

DB Bloc des données

E/S Entrées/Sorties

FB Bloc fonctionnel

FBD Fonction Bloc Diagram

GRAFCET Graphe Fonctionnel de Commande des Étapes et Transitions

IL instruction list

LD Ladder Diagram

MPI L'interface multipoint

OB Bloc d'organisation

S7 Step 7

SCL Screen Control Language

SIMATIC Siemens Automatique

TOR Tout Ou Rien

Table des matières

| | |
|--|-----------|
| Remerciement | I |
| Dédicaces | II |
| Glossaire | IV |
| Introduction générale | 1 |
| 1 L'industrie automatisé | 3 |
| 1.1 Introduction | 4 |
| 1.2 Historique [2] | 4 |
| 1.3 Les systèmes automatisés | 5 |
| 1.3.1 Définition [1] | 5 |
| 1.4 Structure des systèmes automatisés [3] | 6 |
| 1.4.1 Partie contrôle (PC) | 6 |
| 1.4.2 Partie opération (PO) | 7 |
| 1.5 But d'un système automatisé | 7 |
| 1.6 Les avantages et les inconvénients d'un système automatisé [1] | 7 |
| 1.6.1 Avantages | 7 |
| 1.6.2 Inconvénients | 8 |
| 1.7 Chaîne de fonctions d'un système automatisé | 8 |
| 1.8 Propriété d'une chaîne fonctionnelle [6] | 9 |
| 1.8.1 Chaîne d'action | 9 |
| 1.8.2 Chaîne d'acquisition | 10 |
| 1.8.3 Partie traitement | 10 |
| 1.9 Définitions des capteurs et caractéristiques générales | 10 |
| 1.9.1 Définitions [5] | 10 |
| 1.9.2 Capteurs passifs [6] | 11 |

| | | |
|----------|---|-----------|
| 1.9.3 | Capteurs actifs | 11 |
| 1.9.4 | Transmetteur | 12 |
| 1.9.5 | Capteurs intégrés [6] | 12 |
| 1.9.6 | Capteur intelligent | 12 |
| 1.10 | Les capteurs analogiques | 13 |
| 1.10.1 | Les capteurs potentiométriques [7] | 13 |
| 1.11 | Les capteurs numériques [7] | 14 |
| 1.11.1 | Les codeurs rotatifs | 14 |
| 1.11.2 | Le codeur incrémental [7] | 15 |
| 1.11.3 | Le codeur absolu | 16 |
| 1.12 | Les capteurs TOR [7] | 17 |
| 1.12.1 | Les capteurs mécaniques ou interrupteurs de position | 17 |
| 1.12.2 | Les capteurs de proximité | 18 |
| 1.12.3 | Les capteurs de proximité inductifs | 19 |
| 1.12.4 | Les capteurs de proximité capacitifs | 20 |
| 1.13 | Les capteurs photoélectriques [7] | 20 |
| 1.14 | Choix d'un capteur | 20 |
| 1.15 | conclusion | 21 |
| 2 | Un Aperçu sur les API's & présentation de la station tri | 22 |
| 2.1 | Introduction [15] | 23 |
| 2.2 | Automates Programmables Industriels | 23 |
| 2.3 | Interfaces d'entrées/Sorties | 27 |
| 2.3.1 | Types des signaux d'Entrées/Sorties | 27 |
| 2.3.2 | Modules d'Entrées TOR | 27 |
| 2.3.3 | Modules de sorties TOR | 29 |
| 2.3.4 | Cartes d'entrées analogiques [12] | 33 |
| 2.3.5 | Cartes de sorties analogiques [12] | 36 |
| 2.3.6 | Cartes d'entrées/sorties spéciaux [12] | 37 |
| 2.4 | Avantages des automates fixes | 39 |
| 2.5 | Inconvénients des automates fixes | 39 |
| 2.6 | Les langages de Programmation des APIs | 39 |
| 2.6.1 | Langage GRAFCET | 40 |
| 2.6.2 | Langage CONTACT (LADDER) | 43 |

| | | |
|----------|--|-----------|
| 2.6.3 | Langage LOGIGRAMME | 47 |
| 2.6.4 | Langage LIST [9] | 51 |
| 2.6.5 | Langage SCL [14] | 53 |
| 2.7 | Description de la station tri | 54 |
| 2.8 | Les composants | 55 |
| 2.8.1 | Module de Convoyeur | 55 |
| 2.8.2 | Les entrées et les sorties de la station tri | 56 |
| 2.8.3 | Module de barrage (stoppeur) | 58 |
| 2.8.4 | Module de dérivation | 58 |
| 2.8.5 | Pupitre de commande | 59 |
| 2.9 | Conclusion | 59 |
| 3 | L'automatisation de notre station de tri | 60 |
| 3.1 | Introduction | 61 |
| 3.2 | Cahier des charges | 61 |
| 3.3 | Problématique | 61 |
| 3.3.1 | Solutions | 62 |
| 3.4 | SIMATIC STEP7 | 62 |
| 3.4.1 | Définition logiciel STEP7 | 62 |
| 3.4.2 | Les étapes fondamentales d'un projet d'automatisation sous Step7 : | 62 |
| 3.5 | Table des Mnémoniques | 69 |
| 3.6 | Programme de la station tri | 70 |
| 3.6.1 | Les blocs de programmation | 70 |
| 3.7 | Compilation et simulation du programme | 73 |
| 3.7.1 | L'EasyPort USB | 73 |
| 3.7.2 | Test du programme dans S7-PLCSIM | 76 |
| 3.7.3 | Visualisation | 77 |
| 3.8 | Conclusion | 79 |
| | Conclusion Générale | 81 |
| | références bibliographiques | 83 |
| | Résumé | 84 |

Table des figures

| | | |
|------|--|----|
| 1.1 | Processus d'un système automatisé.[1] | 6 |
| 1.2 | Organisation générale d'une chaîne fonctionnelle. [6] | 9 |
| 1.3 | Capteur potentiométrique.[7] | 13 |
| 1.4 | Codeur absolu.[7] | 14 |
| 1.5 | Codeur incrémental.[7] | 15 |
| 1.6 | Principe d'un codeur incrémental.[7] | 16 |
| 1.7 | Principe d'un capteur absolu.[7] | 17 |
| 1.8 | Capteur électromécanique.[7] | 18 |
| 1.9 | Capteurs de proximité. [7] | 18 |
| 1.10 | Détecteur sur vérin pneumatique. [7] | 19 |
| 1.11 | Capteurs de proximité inductifs.[7] | 19 |
| 1.12 | Capteurs de proximité capacitifs. [7] | 20 |
| 2.1 | Système basé sur API. [12] | 24 |
| 2.2 | Automate programmable industriel.[12] | 24 |
| 2.3 | Architecture matérielle d'un API.[13] | 25 |
| 2.4 | Schéma fonctionnel d'une entrée TOR de type AC.[12] | 27 |
| 2.5 | Schéma fonctionnel d'une entrée TOR de type DC.[12] | 28 |
| 2.6 | Connexion des entrées TOR de type TTL. [12] | 28 |
| 2.7 | Connexion des entrées TOR de type BCD.[12] | 29 |
| 2.8 | Schéma fonctionnel d'une sortie TOR de type AC.[12] | 30 |
| 2.9 | Connexion des sorties TOR de type DC. [12] | 30 |
| 2.10 | Connexion des sorties TOR de type TTL.[12] | 31 |
| 2.11 | Connexion des sorties TOR de type BCD.[12] | 32 |
| 2.12 | Connexion des sorties TOR de type relais.[12] | 32 |
| 2.13 | Principe de fonctionnement d'une carte Bypass.[12] | 33 |
| 2.14 | Principe de fonctionnement d'une carte d'entrée analogique. [12] | 34 |
| 2.15 | Connexion d'une carte d'entrée analogique.[12] | 35 |

| | | |
|------|---|----|
| 2.16 | Sortie analogique.[12] | 36 |
| 2.17 | Cartes d'action directe.[12] | 38 |
| 2.18 | Cartes intelligentes.[12] | 38 |
| 2.19 | Principe de grafcet.[10] | 41 |
| 2.20 | Etape initiale.[10] | 41 |
| 2.21 | Franchissement d'une transition.[10] | 42 |
| 2.22 | Evolution des étapes active.[10] | 43 |
| 2.23 | Bloc temporisation.[9] | 45 |
| 2.24 | Fonction comptage/décomptage.[9] | 45 |
| 2.25 | Monostable.[9] | 45 |
| 2.26 | Registre télémécanique. [9] | 46 |
| 2.27 | Générateur d'impulsions SIEMENS. [9] | 46 |
| 2.28 | Exemple d'un réseau Ladder. [9] | 47 |
| 2.29 | Fonction ET. [9] | 48 |
| 2.30 | Fonction OU. [9] | 48 |
| 2.31 | Contact à front montant/descendant. [9] | 48 |
| 2.32 | Opération sortie. [9] | 48 |
| 2.33 | Opération SET. [9] | 49 |
| 2.34 | Opération RESET. [9] | 49 |
| 2.35 | Opération de transfert. [9] | 49 |
| 2.36 | Bloc de conversion. [9] | 50 |
| 2.37 | Type de temporisation TON. [9] | 50 |
| 2.38 | Type de compteur CTU. [9] | 50 |
| 2.39 | Bloc PLS. [9] | 51 |
| 2.40 | Vue de la station de tri. [11] | 55 |
| 2.41 | Convoyeur et les glissières de la station. | 56 |
| 2.42 | Vue détail de convoyeur et de la station de tri. [11] | 56 |
| 2.43 | Module de barrage de la station. | 58 |
| 2.44 | Module de dérivation de la station. | 58 |
| 2.45 | Pupitres des commandes (station tri). | 59 |
| 3.1 | Fenêtre SIMATIC Manager. | 63 |
| 3.2 | Fenêtre de démarrage de Step7. | 63 |
| 3.3 | Insertion d'une nouvelle station. | 64 |

| | | |
|------|--|----|
| 3.4 | Ouverture de la fenêtre de configuration matérielle (HW Config). | 65 |
| 3.5 | Insertion du rack. | 65 |
| 3.6 | Implémentation une alimentation : | 66 |
| 3.7 | Emplacement notre CPU. | 67 |
| 3.8 | Confirmation de communication via câble Ethernet. | 67 |
| 3.9 | Configuration réseau dans STEP 7. | 68 |
| 3.10 | Hierarchie du programme STEP7. | 69 |
| 3.11 | Variables utilisées dans notre programme. | 70 |
| 3.12 | Ajouter un bloc FB1. | 71 |
| 3.13 | S7-GRAPH de la station trie. | 71 |
| 3.14 | Grafcet de commande. | 72 |
| 3.15 | Liaison du bloc FB par OB1 via DB1. | 73 |
| 3.16 | L'EasyPort USB. | 74 |
| 3.17 | Présentation de l'EasyPort USB [17]. | 75 |
| 3.18 | Interface EzOPC. | 76 |
| 3.19 | S7-PLCSIM. | 77 |
| 3.20 | Chargement des blocs. | 77 |
| 3.21 | Simulation de programme [FB1] (graphe de commande). | 78 |
| 3.22 | Simulation de programme [FB1]. | 78 |

Liste des tableaux

| | | |
|-----|--|----|
| 1.1 | Quelques capteurs passifs. | 11 |
| 1.2 | Quelques capteurs actifs. | 11 |
| 2.1 | Exemples des entrées analogiques. | 34 |
| 2.2 | Valeurs électriques standard des entrées analogiques.[12] | 35 |
| 2.3 | Valeurs électriques standard des sorties analogiques.[12] | 36 |
| 2.4 | Les principaux éléments (contacte et bobines) d'un réseau LD.[8] | 44 |
| 2.5 | Entrées / Sorties de la station trie. | 57 |
| 2.6 | Entrées/sorties (Partie Commande). | 59 |

INTRODUCTION GÉNÉRALE

Il est tout à fait banal pour les ingénieurs d'aspirer à des systèmes automatisés capables d'effectuer les tâches les plus répétitives. Parfois, cette automatisation est si rapide et précise qu'elle effectue des actions impossibles pour un être humain. L'automatisation est donc aussi synonyme de productivité et de sécurité.

Et Parmi leurs dimensions, nous citons :

- Relié à un ordinateur de contrôle, il produit un effet efficace d'intelligence et d'indépendance.
- Aussi parce qu'il est programmable, il peut nous fournir des commandes répétées.
- Ainsi il offre la possibilité de contrôler plusieurs machines en même temps avec les mêmes instructions.

Apparus aux États-Unis, les automates programmables sont des dispositifs conçus spécifiquement pour contrôler et faciliter les opérations industrielles d'une machine à tous égards.

C'est une mémoire programmable utilisant un langage spécifique.

Notre projet a été réalisé dans le laboratoire de recherche productique MELT de l'Université Sur un Matériel pédagogique FESTO de Tlemcen

Nous utilisons le système MPS500, puis nous proposerons une solution pour station de trie, nous l'exécuterons en fonction de celle-ci et nous terminerons la procédure par des tests fonctionnels appropriés.

A cet effet, ce mémoire comportera donc trois chapitres : le premier donne un aperçu sur l'automatisation industriels, le second chapitre abordera une présentation de la station de tri et description des différentes fonctions, ainsi que les automates programmables, et pour terminer, le dernier chapitre exposera l'automatisation de cette station par un cahier de charge bien définie et bien déterminé ainsi que la modélisation de ce dernier par l'outil GRAFCET, aussi ce chapitre est consacré à la programmation de l'API via l'outil SIMATIC STEP7 et la simulation par l'automate virtuel PLC Sim.

CHAPITRE

1

L'INDUSTRIE AUTOMATISÉ

1.1 Introduction

Depuis l'époque de la révolution industrielle britannique, elle s'est engagée à développer des technologies complexes pour aider les gens à effectuer différentes tâches de production. Le processus de production industrielle consiste en une série de machines à travers lesquelles la combinaison de matières premières est transférée et transformée en produit final. Ici, le terme « Machine » peut être n'importe quoi, comme des moteurs, des forêts, des bandes transporteuses, etc. Équipements appartenant à des équipements électromécaniques ou à des machines chimiques (tels que des fours, des séchoirs, des systèmes de combustion chimique, etc.).

Aujourd'hui, l'automatisation industrielle a pris en charge le processus de production dans les industries et il est très difficile d'imaginer une ligne de production sans systèmes d'automatisation. Plusieurs facteurs conduisent à la mise en œuvre d'un système d'automatisation dans la production industrielle, comme l'exigence de produits de haute qualité, les attentes élevées en matière de fiabilité des produits, la production en grande quantité, etc... [1]

1.2 Historique [2]

Les automates programmables (PLC) sont apparus à la fin des années 1960 à la demande de l'industrie automobile américaine, exigeant que son système de contrôle ait une plus grande adaptabilité. Ensuite, le coût de l'équipement électronique permet de remplacer la technologie avantageusement utilisée. Les technologies précédemment utilisées étaient des systèmes câblés basés sur des relais électromécaniques et des systèmes pneumatiques et hydrauliques. Avec l'avènement des microprocesseurs et des systèmes de programmation, les restrictions imposées par l'utilisation de systèmes filaires sont devenues déraisonnables. Parmi ces lacunes, on peut énumérer :

- Les câbles sont encombrants et coûteux.
- Aucune flexibilité.
- Incapable de communiquer.

1.3 Les systèmes automatisés

1.3.1 Définition [1]

L'automatisation industrielle consiste à utiliser la programmation logique numérique pour faire fonctionner des machines et d'autres équipements industriels, qui réduit l'intervention manuelle dans le processus de prise de décision et utilise des équipements mécaniques pour le contrôle manuel.

La définition ci-dessus n'est bien sûr pas facile à comprendre, mais essayons de comprendre ce qu'est l'automatisation industrielle à travers un petit exemple.

- **Exemple :**

Considérons un processus de production industrielle manuel, au cours duquel l'opérateur observe la température du four. Supposons que la tâche consiste à atteindre une certaine température et à la maintenir pendant environ 30 minutes. Par conséquent, l'opérateur doit d'abord ajuster la quantité de combustible fournie au four en commandant la soupape pour augmenter la température jusqu'à la quantité requise. Après avoir atteint la température requise, la vanne doit être constamment ajustée pour maintenir la température, c'est-à-dire augmenter ou diminuer la quantité de carburant en fonction de la température dans les 30 prochaines minutes.

Aujourd'hui, avec l'automatisation industrielle, l'ensemble du processus peut être achevé sans l'aide d'un opérateur. Tout d'abord, placez le capteur de température près du four et transmettez la température à l'ordinateur. Ensuite, une électrovanne, également contrôlée par un ordinateur, est utilisée pour alimenter le four en combustible. En fonction de la température lue par le capteur, l'ordinateur ouvrira la vanne pour injecter plus de carburant. Une fois la température requise atteinte, la vanne se ferme. Cependant, en fonction de la lecture de la température, l'ordinateur peut ouvrir ou fermer la vanne et même mettre une petite quantité de carburant. Une fois que le minuteur de l'ordinateur affiche 30 minutes, l'ordinateur peut arrêter complètement le système.

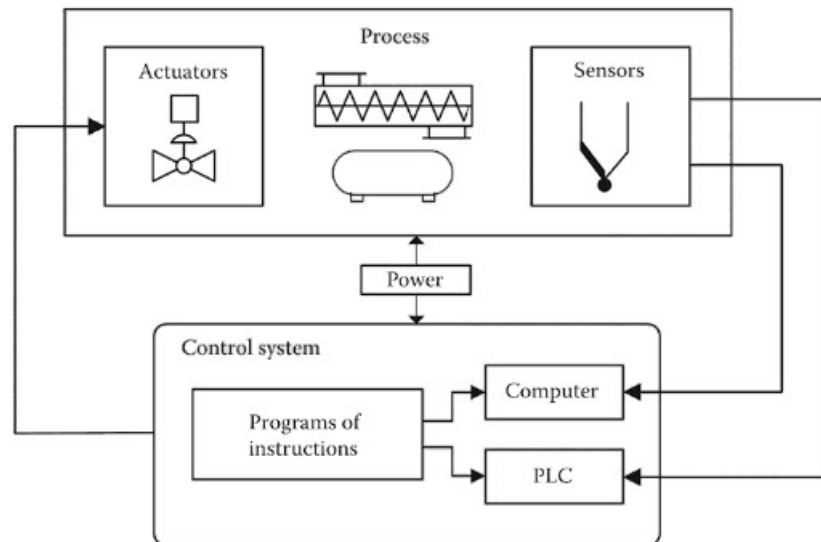


FIGURE 1.1 – Processus d'un système automatisé.[1]

L'exemple ci-dessus peut sembler vague, mais il aide à comprendre comment mettre en œuvre un système d'automatisation industrielle typique. Dans l'exemple ci-dessus, il n'y a absolument aucune intervention humaine et toute la tâche est complètement accomplie par le système automatisé. [1]

1.4 Structure des systèmes automatisés [3]

Le système d'automatisation est essentiellement composé de deux parties principales par rapport aux autres parties pour assurer le fonctionnement du système et l'interaction entre les parties.

1.4.1 Partie contrôle (PC)

C'est la partie opérationnelle de l'architecture avec des composants mécaniques, pneumatiques et hydrauliques, qui peuvent se résumer à un actionneur et répondre au signal électrique envoyé par la partie commande, puis exécuter physiquement le processus industriel. Il existe également des éléments de détection physique, tels que des capteurs, qui permettent à la partie commande de notifier passivement l'état du système à un instant donné (la partie commande est alors chargée de vérifier l'état). Actif (déclencher un signal lors d'un événement physique spécifique).

1.4.2 Partie opération (PO)

Il délivre des commandes de fonctionnement à la partie opérateur. Donc elle :

- * Envoyer des instructions d'opération à la partie exécution. Ces commandes sont transmises au pré-exécuteur.
- * Recevoir des informations sur l'état du site chirurgical envoyées par le capteur.
- * Recevoir les instructions d'utilisation de la console.
- * Envoyer des signaux de signalisation.
- * Assurez-vous que les informations sont traitées selon la logique donnée (programme) pour passer une commande.

1.5 But d'un système automatisé

Chaque système a un objectif. Dans ce cas, son rôle est :

- a. Réduire les coûts et le temps de production, simplifier le travail manuel
- b. Éliminer les tâches douloureuses et répétitives
- c. Améliorer la productivité et améliorer la qualité
- d. Réduire les risques de travail et améliorer la sécurité
- e. Économiser les matières premières, économiser de l'énergie
- f. Maintenir la qualité

1.6 Les avantages et les inconvénients d'un système automatisé [1]

1.6.1 Avantages

- les tâches effectuées par les opérateurs humains impliquant un travail physique fastidieux peuvent être facilement remplacées.
- Les opérateurs humains peuvent éviter de travailler dans des environnements de production dangereux avec des températures extrêmes, de la pollution, des éléments intoxicants ou des substances radioactives.
- Les tâches qui sont difficiles pour un opérateur humain typique peuvent être facilement effectuées. Ces tâches comprennent le levage de charges lourdes et volumineuses, le travail

avec des objets extrêmement petits, etc.

- La production est toujours plus rapide et le coût du produit est nettement inférieur (par rapport au même produit fabriqué avec une opération manuelle).
- Plusieurs contrôles de qualité peuvent être intégrés dans le processus de production pour assurer la cohérence et l'uniformité.
- L'économie de l'industrie peut être considérablement améliorée, ce qui a un impact direct sur le niveau de vie.

1.6.2 Inconvénients

- Perte d'emplois. Comme la majorité du travail est effectué par des machines, le besoin en travail manuel est très réduit.
- Toutes les tâches souhaitées ne peuvent être automatisées à l'aide de la technologie actuelle. Par exemple, il est préférable de laisser l'assemblage manuel des produits de forme et de taille irrégulières. (Cette tendance semble changer avec les ordinateurs et les algorithmes avancés).
- Il est possible d'utiliser l'automatisation pour certains processus, par exemple la production en grande quantité, les produits reproductibles et cohérents.
- Le coût initial de la mise en œuvre d'un système d'automatisation est très élevé. Un ensemble de personnel qualifié est toujours nécessaire pour la maintenance et l'entretien.

1.7 Chaîne de fonctions d'un système automatisé

La chaîne fonctionnelle constitue l'unité de base de la conception, étudiez le fonctionnement du système automatisé. Il se caractérise par la disposition des fonctions de chaque membre. Forme de chaîne, qui combine tous les éléments de la partie exécution. Partie de contrôle du système qui facilite l'exécution des fonctions chirurgicales. Parce que tous ces éléments sont fonctionnellement liés les uns aux autres, ce regroupement est appelé une chaîne fonctionnelle.

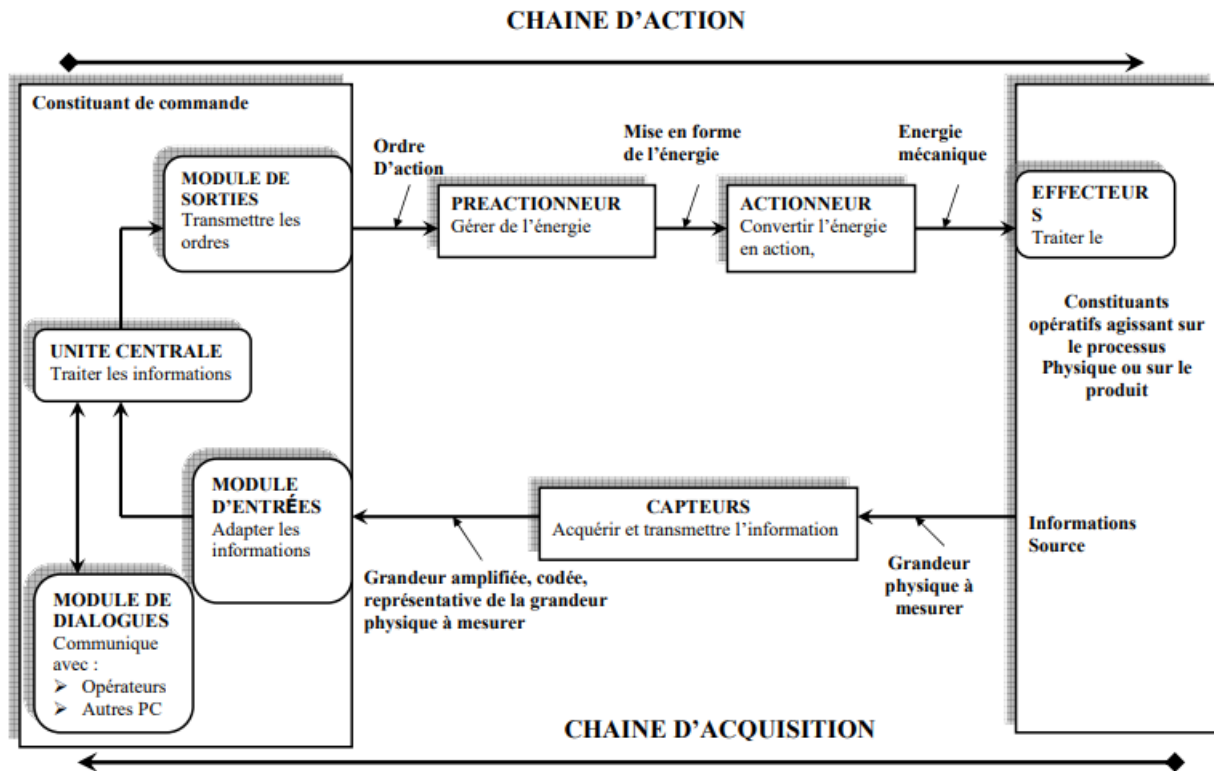


FIGURE 1.2 – Organisation générale d'une chaîne fonctionnelle. [6]

1.8 Propriété d'une chaîne fonctionnelle [6]

Elle contient généralement 3 parties :

1.8.1 Chaîne d'action

Il assure l'animation de la partie opération nécessaire au fonctionnement du matériau de travail à partir du développement de la commande au sein des éléments constitutifs :

- **Pré-actionneur :**

L'interface entre l'unité de contrôle (PC) et l'unité d'opérative (PO). Sa fonction est de transférer les commandes du PC au PO. Généralement, ils sont utilisés pour contrôler la puissance sur la base de signaux de contrôle de faible puissance.

- **Actionneur :**

Ils convertissent l'énergie d'entrée en énergie de sortie appropriée pour que l'actionneur exécute

la tâche de fonctionnement.

- **Effecteur :**

C'est l'élément terminal de la chaîne fonctionnelle. Il agit directement sur le produit et réalise la valeur ajoutée recherchée par l'opération ou la tâche d'opération. Il est généralement connecté à l'actionneur via une chaîne d'alimentation. La chaîne de puissance est une structure composée d'éléments de transmission, de conversion et de guidage, qui permet de rendre l'énergie de sortie de l'actionneur disponible pour une utilisation par l'actionneur.

1.8.2 Chaîne d'acquisition

Elle permet de développer et de transmettre des informations qui peuvent être utilisées par l'unité centrale, ainsi de remplir la fonction de "collecte d'informations". Elle se compose généralement par des capteurs qui font la fonction d'obtention des informations. Cela comprend la mesure des phénomènes physiques et la fourniture d'images informatives de ceux-ci.

1.8.3 Partie traitement

La partie de traitement de la chaîne fonctionnelle, utilisée pour traiter les informations dans la chaîne d'achat pour préparer la commande de la chaîne d'action.

1.9 Définitions des capteurs et caractéristiques générales

1.9.1 Définitions [5]

Un capteur est un dispositif qui transforme l'état d'une grandeur physique observée en une grandeur utilisable, exemple : une tension électrique, une hauteur de mercure, une intensité, la déviation d'une aiguille. . . . On fait souvent (à tort) la confusion entre capteur et transducteur : le capteur est au minimum constitué d'un transducteur.

Le capteur se distingue de l'instrument de mesure par le fait qu'il ne s'agit que d'une simple interface entre un processus physique et une information manipulable. Par opposition, l'instrument de mesure est un appareil autonome se suffisant à lui-même. Il dispose donc d'un affichage

ou d'un système de stockage des données. Ce qui n'est pas forcément le cas du capteur. Les capteurs sont les éléments de base des systèmes d'acquisition de données. Leur mise en œuvre est du domaine de l'instrumentation.

1.9.2 Capteurs passifs [6]

Il s'agit d'impédances (très souvent des résistances) dont l'un des paramètres déterminants est sensible au mesurande.

| Mesurande | Caractéristique électrique sensible | Types de matériaux utilisés |
|------------------------|-------------------------------------|--|
| Température | Résistivité | Métaux : platine, nickel, cuivre. |
| Très basse température | Constante diélectrique | Semi-conducteurs. Verres. |
| Flux lumineux | Résistivité | Semi-conducteurs. |
| Position (aimant) | Résistivité | Matériaux magnéto résistants : bismuth |

TABLE 1.1 – Quelques capteurs passifs.

1.9.3 Capteurs actifs

Ils fonctionnent en générateurs en convertissant la forme d'énergie propre au mesurande en énergie électrique.[6]

| Mesurande | Effet utilisé | Grandeur de sortie |
|---------------|-----------------------------|--------------------|
| Température | Thermoélectricité | Tension. |
| Flux lumineux | Pyroélectricité | Charge. |
| Force | Piézoélectricité | Charge. |
| Vitesse | Induction électromagnétique | Tension |

TABLE 1.2 – Quelques capteurs actifs.

1.9.4 Transmetteur

Le transmetteur est un dispositif qui convertit le signal de sortie du capteur en un signal de mesure standardisé. Il fait le lien entre le capteur et le système de contrôle et de commande.

Ces principales fonctions sont :

- . L'alimentation du capteur.
- . Linéarisation du signal.
- . Décalage du zéro.
- . Mise à l'échelle (amplification).

1.9.5 Capteurs intégrés [6]

Un tel capteur intègre sur le même substrat de silicium (donc dans un seul composant électronique) :

- . Le capteur.
- . Le conditionnement du signal (le transmetteur).

Ce principe réduit l'encombrement de la chaîne de mesure, facilite la mise en œuvre du capteur et favorise la normalisation des capteurs.

1.9.6 Capteur intelligent

Avec le développement de la technologie LSI (large scale integration) et VLSI (very large scale integration), les capteurs peuvent être intégrés aux circuits de traitement du signal sur la même puce, pour réaliser l'ensemble des fonctions. On les appelle « capteurs intelligents ».

Il intègre :

- * Le capteur.
- * Une conversion analogique numérique.
- * La chaîne de mesure et de traitement numérique du signal pilotée par un microprocesseur.
- * Une mémoire.
- * Une interface de communication numérique standardisée avec un ordinateur ou ordinateur via un bus partagé entre plusieurs capteurs intelligents.

Ces capteurs intelligents peuvent ainsi, outre la prise du signal, assurer des fonctions multiples :

- * Amélioration du rapport signal/bruit par adaptation d'impédance et amplification.
- * Prétraitement du signal, tel que compensation en température et en variation d'alimentation, remise à zéro automatique, filtrage de signaux parasites, corrections de non linéarité.

- * Traitement du signal, tel que codage et modulation des signaux de sortie, moyennage redondance pour acquérir le même signal avec plusieurs capteurs, alarmes intégrées pour signaler des défauts de fonctionnement des capteurs.
- * Logique et décision.
- * Réduction de consommation, par exemple en faisant travailler les capteurs en temps partagé.[6]

1.10 Les capteurs analogiques

La sortie est une grandeur physique dont la valeur est proportionnelle à la grandeur physique mesurée par le capteur. La sortie peut prendre une infinité de valeurs continues. Le signal des capteurs analogiques peut être du type : sortie tension ou sortie courant.

1.10.1 Les capteurs potentiométriques [7]

Les capteurs potentiométriques servent à détecter une position ou un déplacement rectiligne ou angulaire.

La rotation de son axe est liée à la variation de la résistance comprise entre le curseur et l'une de butées par rapport à sa résistance totale : on peut transmettre à distance, un signal électrique de tension proportionnel à la position de l'axe.



FIGURE 1.3 – Capteur potentiométrique.[7]

1.11 Les capteurs numériques [7]

1.11.1 Les codeurs rotatifs

Le contrôle du déplacement, de la position et de la vitesse est un problème habituellement rencontré sur les systèmes automatisés.

Les systèmes de détection conventionnels, interrupteurs de position, détecteur inductif ou photoélectrique trouvent rapidement leurs limites dès lors que le nombre de positions à contrôler devient trop important.

Les codeurs rotatifs permettent au système de traitement de maîtriser le positionnement d'un mobile avec une grande précision et sans répartir sur le système technique un grand nombre de détecteurs de position.

Un codeur optique est un capteur angulaire de position, lié mécaniquement à un arbre qui l'entraîne, son axe fait tourner un disque qui comporte une succession de zones opaques et transparentes. La lumière émise par des diodes électroluminescentes arrive sur des photodiodes chaque fois qu'elle traverse les zones transparentes du disque. Les photodiodes génèrent alors un signal électrique.

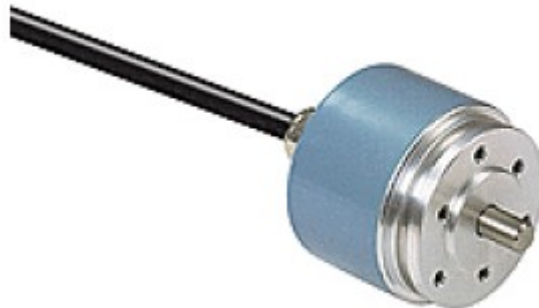


FIGURE 1.4 – Codeur absolu.[7]



FIGURE 1.5 – Codeur incrémental.[7]

1.11.2 Le codeur incrémental [7]

Le codeur incrémental ou relatif est également appelé générateur d'impulsions. Une ou deux pistes extérieures divisées en N intervalles d'angles égaux alternativement opaques et transparents. Pour un tour complet de l'axe codeur, le faisceau lumineux est interrompu N fois et délivre N signaux carrés A et B en quadrature. Le nombre de points par tour du capteur se nomme la résolution. Le déphasage de 90° électrique des signaux A et B permet de déterminer le sens de rotation :

- Dans un sens pendant le front montant du signal (A), le signal (B) est à zéro.
- Dans l'autre sens pendant le front montant du signal (A), le signal (B) est à un.
- La piste intérieure ($Z = \text{top zéro}$) comporte une seule fenêtre transparente et délivre un seul signal par tour.
- Ce signal (Z) d'une durée 90° électrique détermine une position de référence et permet la réinitialisation à chaque tour.
- Le comptage décomptage des impulsions par l'unité de traitement permet de définir la position du mobile.

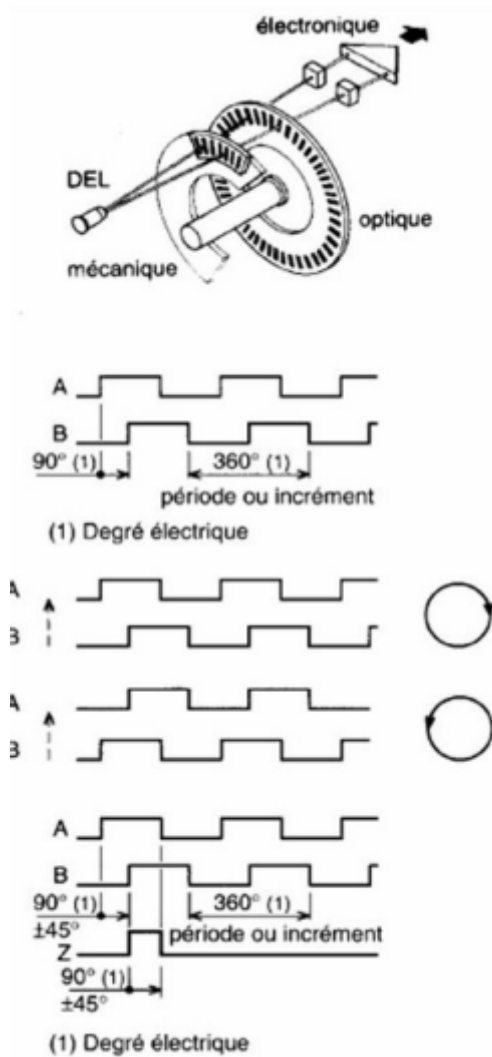


FIGURE 1.6 – Principe d'un codeur incrémental.[7]

1.11.3 Le codeur absolu

Ce concept a été développé pour pallier les contraintes développées ci-dessous et générées par le codeur incrémental (appelé aussi relatif) :

- En cas d'absence prolongée du réseau (en cas d'absence de sauvegarde côté unité de traitement) l'information de position peut être perdue.
- En cas d'une modification de position (déplacement manuel hors tension du mobile contrôlé en position), il y a perte de la position du mobile.
- En cas d'impossibilité de recalage par le "Top zéro" dans le cas de mouvement de type oscillant ne décrivant jamais un tour complet.
- Le disque rotatif du codeur absolu comporte un nombre (n) de pistes.
- Chaque piste a son propre système de lecture (diode émettrice et diode réceptrice).
- A chaque position angulaire de l'axe codeur correspond un code binaire.

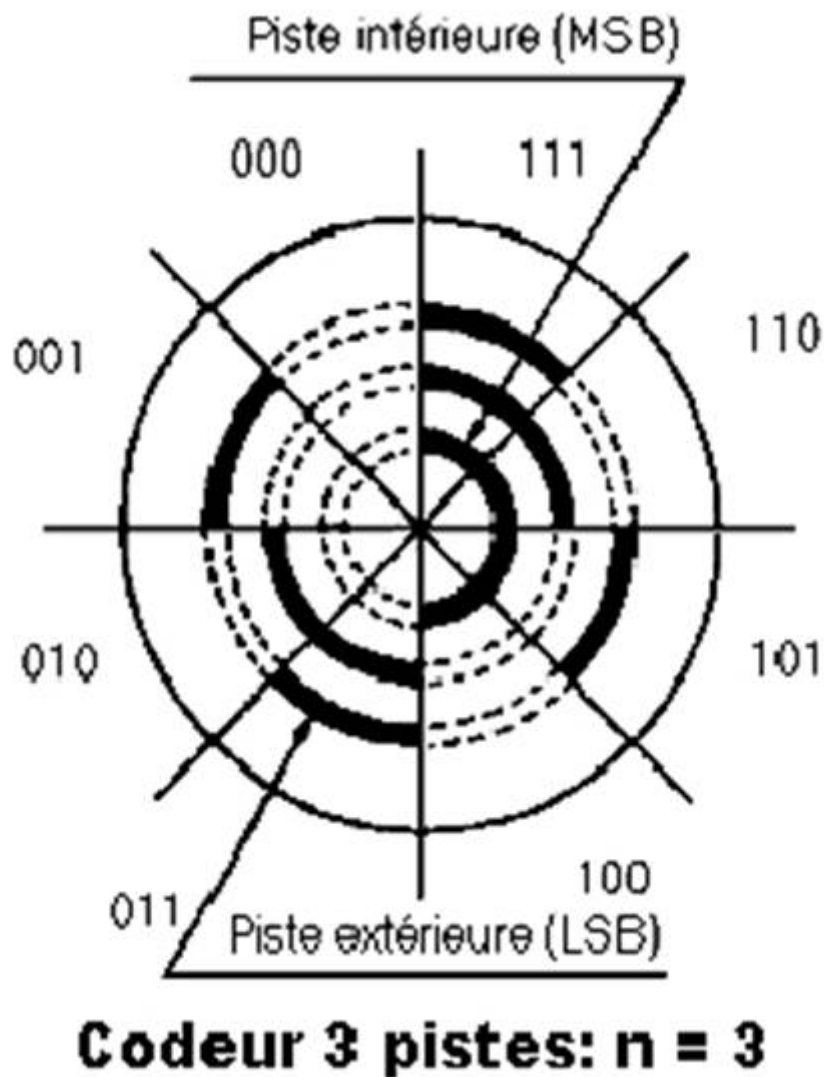


FIGURE 1.7 – Principe d'un capteur absolu.[7]

1.12 Les capteurs TOR [7]

1.12.1 Les capteurs mécaniques ou interrupteurs de position

Les capteurs mécaniques ou interrupteurs de position sont en contact direct avec la pièce en mouvement qu'il faut détecter. L'action mécanique sur la partie mobile du capteur permet d'établir ou d'interrompre un contact électrique.

Ils transmettent au système de traitement les informations de présence, d'absence, de passage, de positionnement ou de fin de course.



FIGURE 1.8 – Capteur électromécanique.[7]

1.12.2 Les capteurs de proximité

Le capteur de proximité permet de détecter tout objet magnétique qui se trouve à proximité de la tête de détection. Un détecteur magnétique se compose essentiellement de 2 lames conductrices. Le passage d'un matériau aimanté entraîne la déformation de ces lames, celles-ci entrent en contact et permettent le passage du courant.



FIGURE 1.9 – Capteurs de proximité. [7]

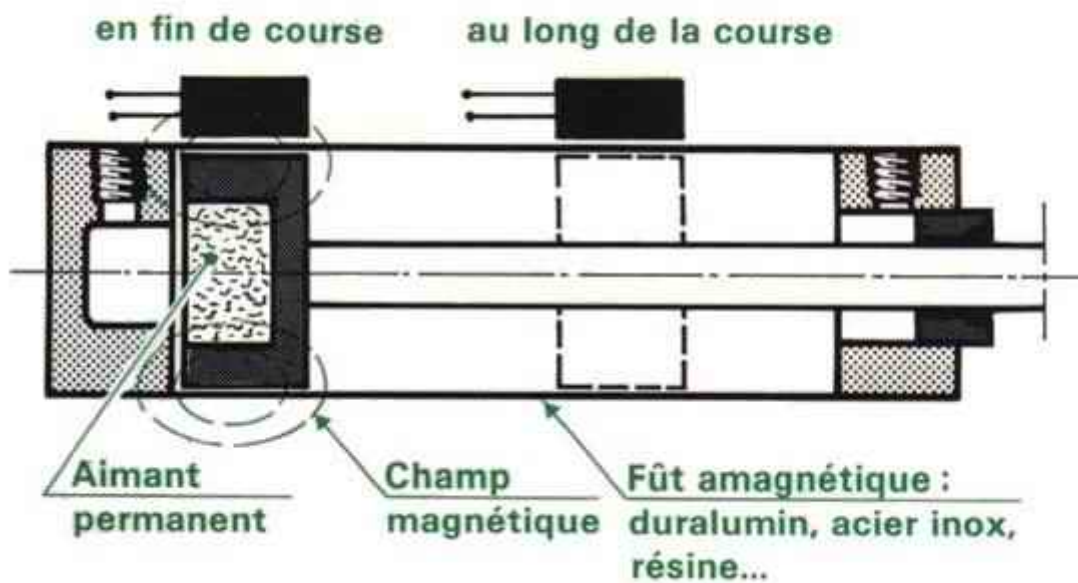


FIGURE 1.10 – Détecteur sur vérin pneumatique. [7]

1.12.3 Les capteurs de proximité inductifs

Les capteurs de proximité inductifs permettent de détecter tout objet métallique qui se trouve à proximité de la tête de détection.



FIGURE 1.11 – Capteurs de proximité inductifs.[7]

Un capteur inductif se compose essentiellement d'un oscillateur. Le champ électromagnétique est créé à l'avant de la face sensible. Toute pièce métallique pénétrant dans ce champ devient le siège de courants de Foucault qui provoquent l'arrêt des oscillations.

1.12.4 Les capteurs de proximité capacitifs

Les capteurs de proximité capacitifs permettent de détecter tout objet qui se trouve à proximité de la tête de détection.



FIGURE 1.12 – Capteurs de proximité capacitifs. [7]

Un capteur capacitif se compose essentiellement d'un oscillateur dont le condensateur constitue la face sensible. Lorsqu'un matériau conducteur ou isolant de permittivité supérieure à 1 est placé dans ce champ, il modifie la capacité et provoque l'arrêt des oscillations. Comme pour le capteur inductif, c'est l'arrêt des oscillations qui est détecté. L'avantage par rapport au capteur inductif c'est qu'il peut détecter à courte distance la présence de tous types d'objets.

1.13 Les capteurs photoélectriques [7]

Un détecteur photoélectrique réalise la détection d'une cible, qui peut être un objet ou une personne, au moyen d'un faisceau lumineux. Les détecteurs photoélectriques se composent essentiellement d'un émetteur de lumière associé à un récepteur photosensible. La détection est effective quand l'objet pénètre dans le faisceau lumineux et modifie suffisamment la quantité de lumière reçue par le récepteur pour provoquer un changement d'état de la sortie.

1.14 Choix d'un capteur

Il existe différentes familles de détecteurs de présence ; le choix se déroule en deux temps. Le logigramme ci-dessous illustre cette démarche qui conduit à sélectionner une famille de détecteurs sur la base de critères simples.

- Phase 1 : elle consiste à déterminer la famille de détecteurs la mieux adaptée à l'application en répondant aux questions suivantes :
 - * Nature de l'objet à détecter : solide, liquide, gazeux, métallique ou non.

- * Contact possible avec l'objet.
- * Distance objet/détecteur, masse de l'objet.
- * Vitesse de défilement.
- * Cadences de manœuvres.
- * Espace d'intégration du détecteur dans la machine.
- Phase 2 : elle vise à déterminer le type et la référence du détecteur recherché. Cette deuxième phase tient compte :
 - * L'environnement : température, humidité, poussières, projections diverses, etc...
 - * La source d'alimentation : alternative ou continue.
 - * Le signal de sortie : électromécanique, statique.
 - * Le type de raccordement : câble, bornier, connecteur.[7]

1.15 conclusion

Dans ce chapitre, nous avons présenté la Généralité sur les systèmes automatisé ainsi nous avons vu une vue théorique sur les capteurs et leurs types.

CHAPITRE

2

UN APERÇU SUR LES API'S &
PRÉSENTATION DE LA STATION TRI

2.1 Introduction [15]

La première apparition des automates programmables industriel remonte à 1969. Son apparition a conduit à la révolution industrielle dans le monde. Tous les systèmes automatisés sont devenus des API contrôlées et gérées, ce qui se traduit par un accès accru à la production avec des produits de haute qualité. C'est la demande et le but de tous. Il est utilisé pour plusieurs raisons :

- Progrès scientifique et technique.
- Complexité accrue dans divers processus industriels.
- Nécessaire pour la vitesse et la précision dans les performances.
- Besoin de plus de contrôle dans le même processus.

Il existe plusieurs types d'API, et chacune est différente de l'autre. Le type approprié est sélectionné parmi plusieurs catégories si nécessaire :

- Quantité et type d'entrée / sortie.
- Type de processeur.
- Fonctions ou modules spéciaux.
- Fonction de communication.

Dans ce chapitre, nous allons regarder tout ce que l'automatisation industrielle dans du contrôle automatique des industriels à l'automatisation de ces systèmes, nous représentons que nous avons étudié. Ensuite, nous représentons une étude approfondie des contrôleurs logiques programmables industriels, les outils et langages de programmation. Puis nous définissons ce système, son architecture, sa structure et son principe de fonctionnement. Ce chapitre regroupe toutes ces informations de manière générale.

2.2 Automates Programmables Industriels

a. Définition :

Les API's ont de nombreuses définitions. Cependant, ils peuvent être considérés en termes simples comme des ordinateurs industriels avec des interfaces d'E / S et des fonctions spécialement conçues. La figure 2.1 montre le schéma conceptuel d'un système basé sur API :

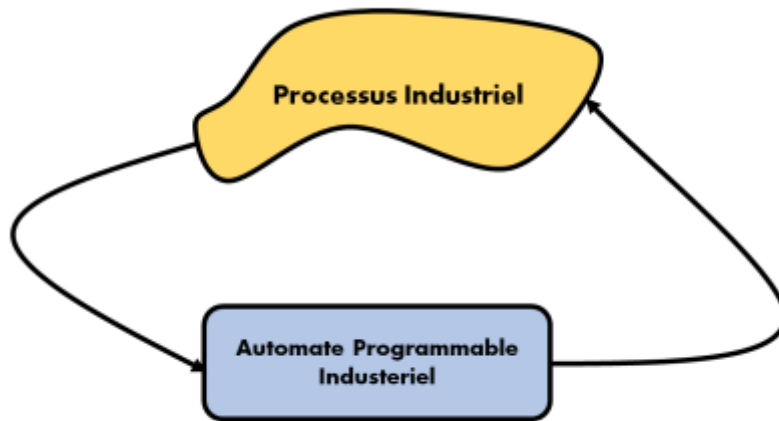


FIGURE 2.1 – Système basé sur API. [12]

De même, il peut être défini comme une forme spéciale de microprocesseur, qui utilise Mémoire programmable pour stocker des instructions et exécuter des fonctions Tels que la logique, le tri, la synchronisation, le comptage et la résolution Opérations arithmétiques et fonctions de communication pour le contrôle Machines et procédés industriels. Les concepteurs de l'API ont préprogrammé le contrôleur afin que le programme de contrôle puisse être saisi dans un langage simple plutôt qu'intuitif. Entrée (capteur, interrupteur, etc.) et sortie (moteur, vanne, etc.) Chacun des systèmes est connecté à l'automate et chaque automate est identifié par sa propre adresse. [12]

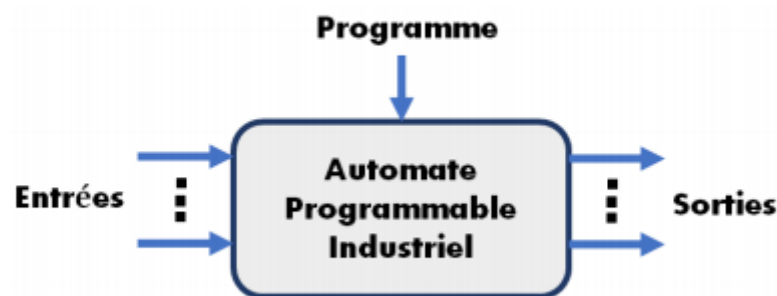


FIGURE 2.2 – Automate programmable industriel.[12]

Ensuite, l'opérateur introduit la séquence d'instructions (programme) dans la mémoire de l'automate. Ce dernier surveille alors l'entrée et la sortie en fonction du programme et exécute

les règles de contrôle qui lui ont été programmées.[12]

b. Architecture des APIs :

Le terme d'architecture d'automate fait référence à la spécification de conception des différents composants matériels et logiciels de l'automate et à la manière dont ils interagissent les uns avec les autres pour former le système global de l'automate. L'architecture d'un automate est basée sur les mêmes principes que ceux utilisés dans l'architecture standard des ordinateurs. Cependant, l'architecture d'un automate diffère parce que la conception est basée sur la fourniture d'une haute fiabilité, l'immunité à un environnement industriel difficile, la facilité de maintenance et l'accès à de grandes quantités d'entrées et de sorties périphériques.

Si un système PLC est décrit comme ayant une architecture fermée, cela fait référence à un système propriétaire de composants matériels et logiciels qui ne peuvent pas (ou difficilement) être connectés à des composants et logiciels d'autres fabricants. En revanche, si un système PLC est décrit comme ayant une architecture ouverte, cela signifie que le système PLC possède des composants prêts à l'emploi qui adhèrent à une norme commune et sont facilement connectés aux composants matériels et logiciels d'autres fabricants.[13]

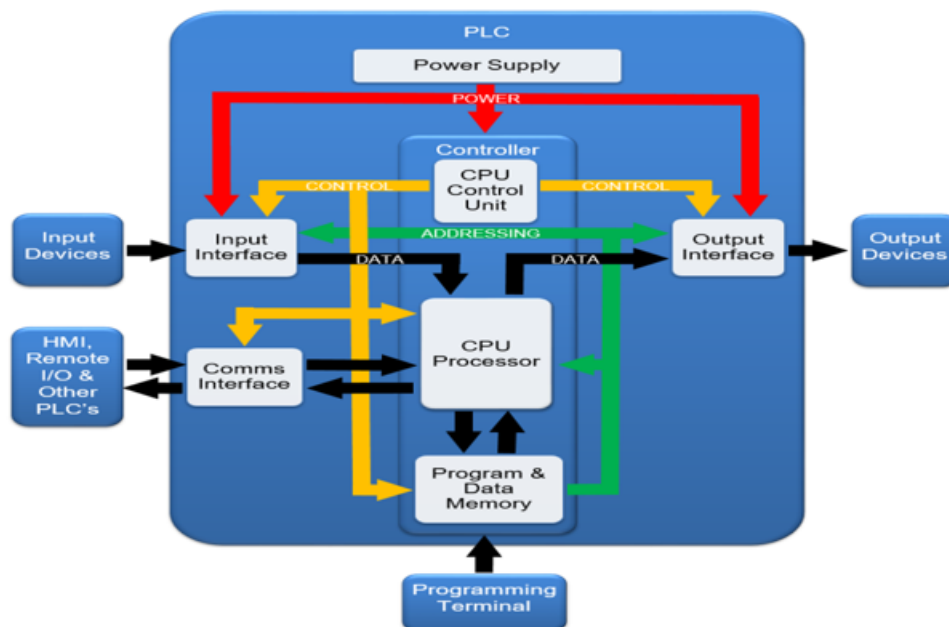


FIGURE 2.3 – Architecture matérielle d'un API.[13]

Un A.P.I. se compose :

i. Une unité de traitement ou processeur : Le processeur gère l'ensemble des échanges informationnels en assurant :

- La lecture des informations d'entrée.
- L'exécution des instructions du programme mis en mémoire.
- La commande ou l'écriture des sorties.

ii. Une mémoire programme : La mémoire programme de type RAM contient les instructions à exécuter par le processeur afin de déterminer les ordres à envoyer aux pré-actionneurs reliés à l'interface de sortie en fonction des informations recueillies par les capteurs reliés à l'interface d'entrée.

iii. Une mémoire de données : La mémoire de donnée permet le stockage de :

- L'image des entrées reliées à l'interface d'entrée.
- L'état des sorties élaborées par le processeur.
- Les valeurs internes utilisées par le programme (résultats de calculs, états intermédiaires, ...).
- Les états Forcés ou non des E/S.

iv. Une interface d'entrée : L'interface d'entrée permet la connexion à l'API d'une multitude de capteurs pouvant être :

- TOR (logiques ou Tout Ou Rien).
- Numériques.
- Analogiques.

Ces différentes entrées sont mises en forme par l'interface d'entrée avant d'être stockées dans la mémoire de données.

v. Une interface de sortie : L'interface de sortie permet la connexion à l'API d'une multitude de pré-actionneurs pouvant être :

- TOR (logiques ou Tout Ou Rien).
- Numériques.
- Analogiques.

vi. Un module d'alimentation : Le module d'alimentation transforme l'énergie externe provenant du réseau en en la mettant en forme afin de fournir aux différents modules de l'API les niveaux de tension nécessaires à leur bon fonctionnement. Plusieurs niveaux de tension peuvent être utilisés par les circuits internes (3v, 5v, 12v, 24v...) il sera dimensionné en fonction des consommations des différentes parties.

2.3 Interfaces d'entrées/Sorties

2.3.1 Types des signaux d'Entrées/Sorties

Généralement, Il existe trois types des signaux d'entrées/sorties : Tout ou Rien (TOR), numériques et analogiques, et :

- * Essentiellement juste un signal marche / arrêt (0/24V... etc.).
- * Une séquence d'impulsions.
- * Un signal dont la valeur est liée à la valeur de la quantité mesurée (0-10V, 4-20mA, ...etc.).[12]

2.3.2 Modules d'Entrées TOR

Les modules d'entrées TOR connectent des dispositifs d'entrées (capteurs) à l'automate programmable sous forme d'un signal logique 0 ou bien 1. Ce type d'entrée n'ayant que deux états. (ON / OFF, OUVERT / FERMÉ, VRAI / FAUX, etc.). En réalité, il y a deux formes d'entrées TOR ; soit sous forme des bits simples, qui contrôlent une entrée codée sur un bit, soit sous forme des bits multiples, qui contrôlent de nombreuses entrées. Ce dernier est appelé modules multi bits, recevant plusieurs entrées, tels que les commutateurs à molette utilisés dans les interfaces de registre (BCD), où le transfert de données qui est représenté par le mot (Word).

i. Types des entrées TOR :

a. Entrées TOR de type AC :

Le schéma fonctionnel d'un circuit d'interface d'une entrée TOR de type AC est présenté par la figure (4). Les circuits d'entrées varient considérablement selon les fabricants d'API, mais en général, les interfaces TOR de type AC fonctionnent de la même manière que le circuit présenté dans la figure (2.4) :

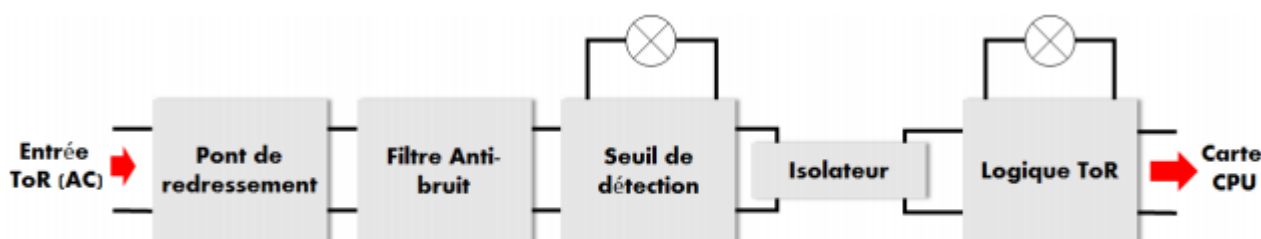


FIGURE 2.4 – Schéma fonctionnel d'une entrée TOR de type AC.[12]

b. Entrées TOR de type DC :

Le schéma fonctionnel d'un circuit d'interface d'une entrée TOR de type DC est présenté par la figure (2.5). En pratique la majorité des entrées TOR est de type DC, ce type de carte travail avec les types de connexions source ou récepteur.

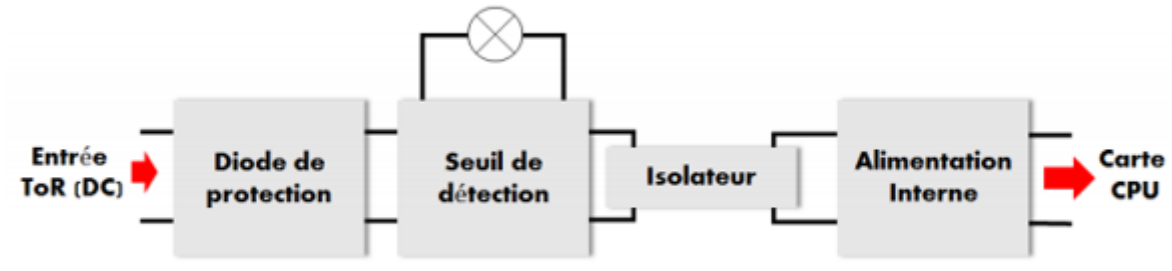


FIGURE 2.5 – Schéma fonctionnel d'une entrée TOR de type DC.[12]

c. Entrées TOR de type TTL :

Les cartes d'entrées de type TTL permettent aux contrôleurs (CPU) d'accepter uniquement des signaux provenant des capteurs compatibles TTL (5VDC), tels que les commandes à semi-conducteurs et des instruments de détection (capteurs photoélectriques...etc.). La figure (2.6) montre la connexion électrique d'un capteur TOR de type TTL :

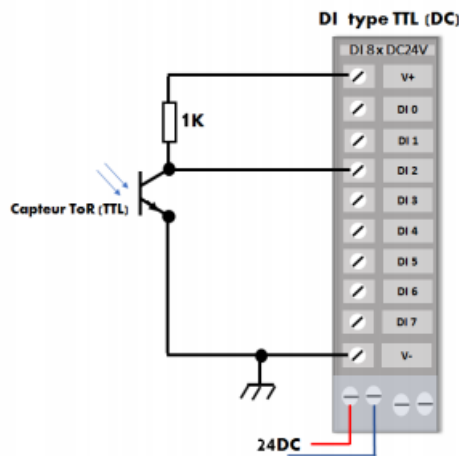


FIGURE 2.6 – Connexion des entrées TOR de type TTL. [12]

d. Entrées TOR de type BCD (multi bits) :

Les cartes d'entrées multi bits à registre/BCD améliorent les méthodes d'interfaçage des entrées avec l'automate programmable grâce à l'utilisation d'interrupteurs à molette standard. Ce registre, ou configuration BCD, permet d'entrer des groupes de bits, c'est-à-dire des bits sous forme parallèle.[12]

La figure (2.7) schématisée la connexion électrique d'un capteur TOR de type BCD :

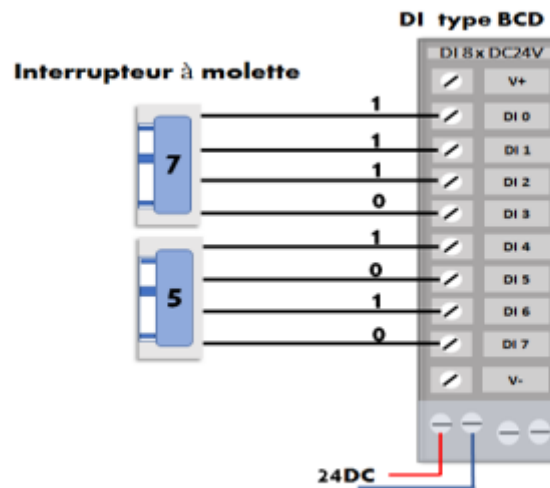


FIGURE 2.7 – Connexion des entrées TOR de type BCD.[12]

2.3.3 Modules de sorties TOR

Comme les modules d'entrées TOR, les modules de sorties TOR connecte des dispositifs de sorties (actionneurs ou prés-actionneurs) de champ à l'automate programmable. Ce type de sortie n'ayant que deux états (ON / OFF, OUVRIR / FERMER, VRAI / FAUX, etc.).

ii. Types des sorties TOR :

a. Sorties TOR de type AC :

Les cartes de sorties AC, comme les cartes d'entrées, varient considérablement selon les fabricants d'API, mais le schéma fonctionnel présenté à la figure (2.8) décrit leur configuration générale. Cette configuration montre les principales sections d'une carte de sortie AC, ainsi que son fonctionnement. [12]

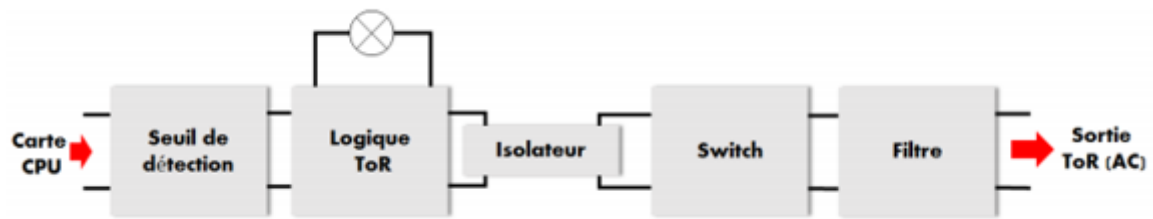


FIGURE 2.8 – Schéma fonctionnel d'une sortie TOR de type AC.[12]

b. Sorties TOR de type DC :

Les cartes de sorties DC contrôle les charges de type DC d'un fonctionnement TOR. Cependant, le circuit qui affecte la sortie est variable en fonction de la puissance, de la rapidité, du rendement et du prix, tel que le contacte d'un relais, triac, transistor, IGBT...etc.[12]

La figure (2.9) montre les schémas de connexion des sorties TOR de type DC :

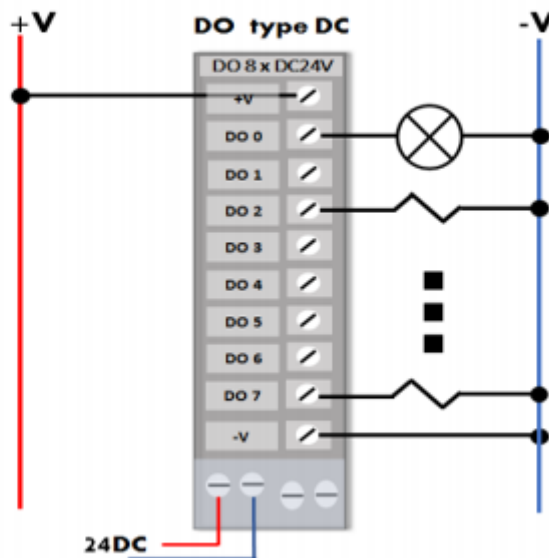


FIGURE 2.9 – Connexion des sorties TOR de type DC. [12]

Comme dans les cartes d'entrées DC, les cartes de sortie DC peuvent avoir des configurations comme source de courant ou récepteur de courant.

c. Sorties TOR de type TTL :[12]

Ce type des cartes de sortie TTL permet à un API de piloter des périphériques de sortie compatibles TTL, tels que des afficheurs sept segments, des circuits intégrés et des périphériques travaillant avec une tension de 5V DC.

La figure (2.10) montre les schémas de connexion des sorties TOR de type TTL :

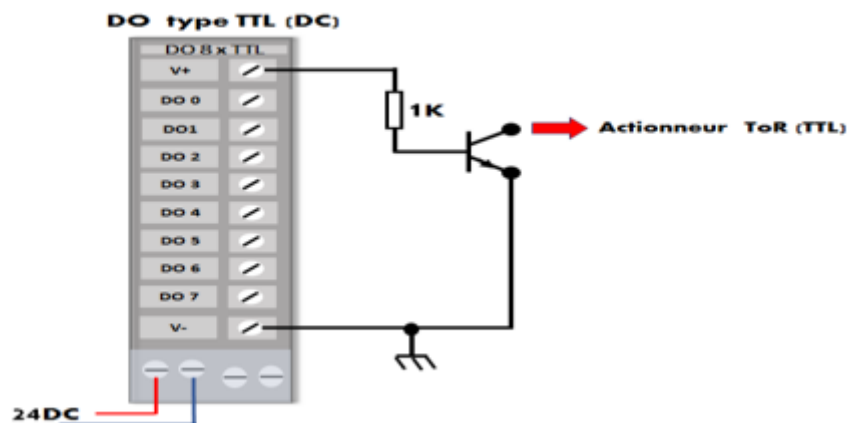


FIGURE 2.10 – Connexion des sorties TOR de type TTL.[12]

d. Connexion des sorties TOR de type TTL :[12]

Les cartes de sortie de type BCD multi bits établissent une communication parallèle entre le processeur (CPU) et un périphérique de sortie, tel qu'un afficheur sept segments ou un afficheur alphanumérique BCD...etc. Cependant, ce type de carte de sortie peuvent également piloter que des petites charges DC avec des exigences de courant faibles (<0,5ampères)

La figure (2.11) montre les schémas de connexion des sorties TOR de type BCD :

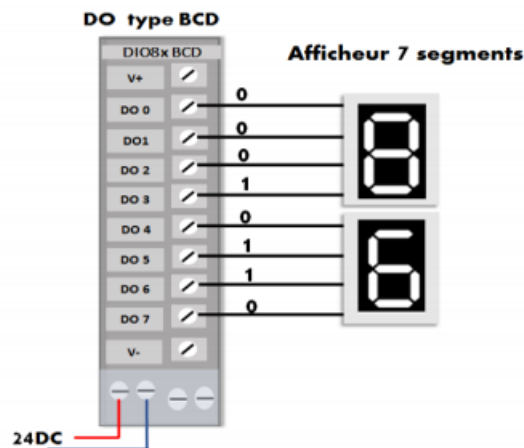


FIGURE 2.11 – Connexion des sorties TOR de type BCD.[12]

e. Sorties TOR de type contact de relais : [12]

Les cartes de sortie de type contact permettent aux périphériques de sortie d'être commutés par des contacts de relais normalement ouverts ou normalement fermés. Ce type de sortie fournissent une isolation électrique entre la partie puissance (charge) et la partie commande (CPU).

La figure (2.12) montre le schéma fonctionnel des sorties TOR de type relais :

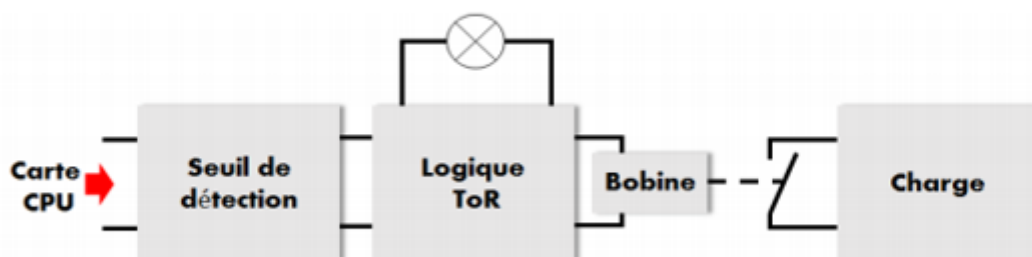


FIGURE 2.12 – Connexion des sorties TOR de type relais.[12]

f. Bypass des sorties TOR : [12]

Les cartes Bypass sont des cartes de contournement de contrôle, ce type de carte donne la possibilité de commander la sortie TOR par deux méthodes, soit par une commande venue de l'API ou bien par une commande manuelle. Aussi, ces cartes permettent une flexibilité lors du

démarrage et des défaillances de sortie. En incorporant un sélecteur qui permet de mettre la sortie TOR sous tension quel que soit l'état de sa carte de sortie (en position ON), comme il met la sortie en position OFF, c'est à dire désactivé.

La figure (2.13) montre le schéma électrique des cartes Bypass :

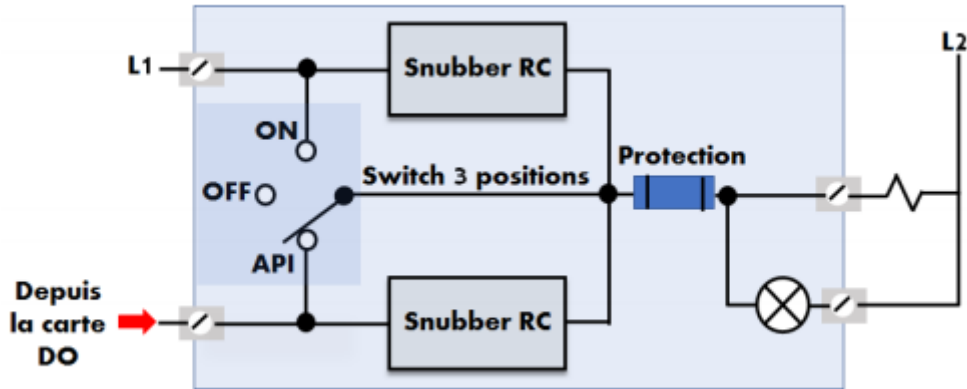


FIGURE 2.13 – Principe de fonctionnement d'une carte Bypass. [12]

2.3.4 Cartes d'entrées analogiques [12]

Les cartes d'entrées analogiques sont utilisées dans les applications où le signal est continu. Contrairement aux signaux TOR, qui ne possèdent que deux états (ON et OFF), les signaux analogiques ont un nombre infini d'états. Par exemple, la température, donne un signal analogique car il varie continuellement par la variation de la température.

Le tableau (2.1) présente quelques types des capteurs analogiques :

| |
|-----------------------------|
| Capteurs analogiques |
| Capteur de débit |
| Capteurs d'humidité |
| Capteur de pression |
| Capteur de vibrations |
| Capteur de température |
| Capteur de niveau |
| Capteur de vitesse |
| Etc... |

TABLE 2.1 – Exemples des entrées analogiques.

Ce type de carte permet de convertir les signaux analogiques, venus depuis les capteurs analogiques, en des signaux numériques exploitables par la carte CPU de l'automate. La figure (2.14) illustre la séquence numérique survenant lors de la lecture d'un signal d'entrée analogique. Le carte (AI) transforme le signal analogique, par l'intermédiaire d'un convertisseur analogique-numérique (CAN), en un mot de 12 bits qui sera stocké dans la mémoire image d'entrée. La valeur analogique stockée dans la mémoire image d'entrée sera au format BCD ou binaire.

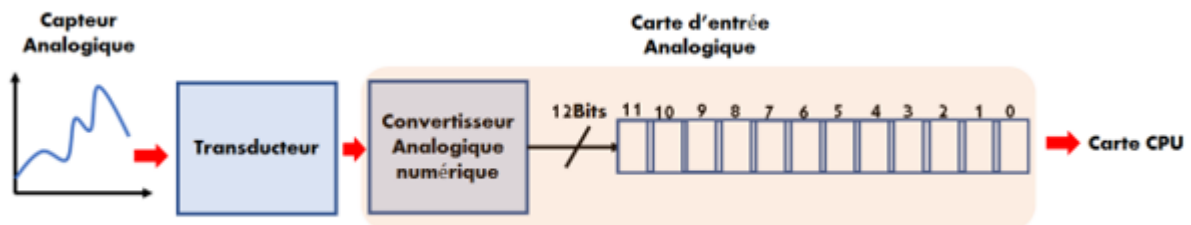


FIGURE 2.14 – Principe de fonctionnement d'une carte d'entrée analogique. [12]

Le rôle du transducteur est de transformer le signal d'entrée en un signal électrique normalisé que l'entrée analogique peut reconnaître (exemple 0-100 bar en 4-20mA). Notant que le transducteur génère un signal électrique de très faible niveau (courant ou tension), donc on doit amplifier ce signal par l'utilisation des transmetteurs, qui à leurs tours envoient le signal à la carte d'entrée analogique (figure 2.15) :

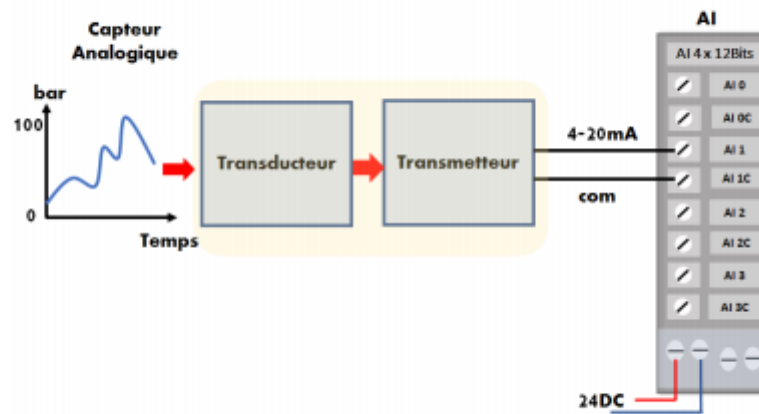


FIGURE 2.15 – Connexion d’une carte d’entrée analogique.[12]

En raison des nombreux types de transducteurs disponibles au marché, les cartes d’entrées analogiques ont plusieurs caractéristiques électriques standards. Le tableau (2.2) répertorie les valeurs électriques standard de courant et de tension pour les cartes d’entrées.

| Valeurs d’entrées électriques |
|-------------------------------|
| + 0 à +1 V(DC) |
| + 0 à +5 V(DC) |
| + 0 à +10 V(DC) |
| + 1 à +5 V(DC) |
| ±5 V(DC) |
| ±10 V(DC) |

TABLE 2.2 – Valeurs électriques standard des entrées analogiques.[12]

iii. Connexions électriques des entrées analogiques : [12]

Les cartes d’entrées analogiques fournissent généralement une impédance d’entrée élevée (dans la gamme des mégaohms) pour les signaux d’entrées de type tension. Pour les cartes d’entrées de type courant fournissent une impédance d’entrée faible (entre 250 et 500 ohms). Ce qui entraîne de prendre en considération le bon choix des transducteurs-transmetteur en point de vue courant et tension.

Les cartes d’entrées analogiques peuvent recevoir des entrées asymétriques ou différentielles,

dans le premier cas, les communs sont reliés ensemble (un seul commun), par contre le dans le cas différentielle chaque entrée est indépendante des autres (chacune son commun). Selon le fabricant, une carte d'entrée peut être configurée en mode asymétrique ou différentiel à l'aide d'un commutateur.

2.3.5 Cartes de sorties analogiques [12]

Les cartes de sorties analogiques sont utilisées dans les applications nécessitant le contrôle des actionneurs répondant à des niveaux de tension ou de courant d'une façon continue. Comme les entrées analogiques, les cartes de sorties analogiques sont généralement connectées à des dispositifs de contrôle via des transducteurs (voir Figure 2.16). Ces transducteurs amplifient, réduisent ou modifient le signal de tension en un signal analogique qui, à son tour de contrôler l'actionneur de sortie.

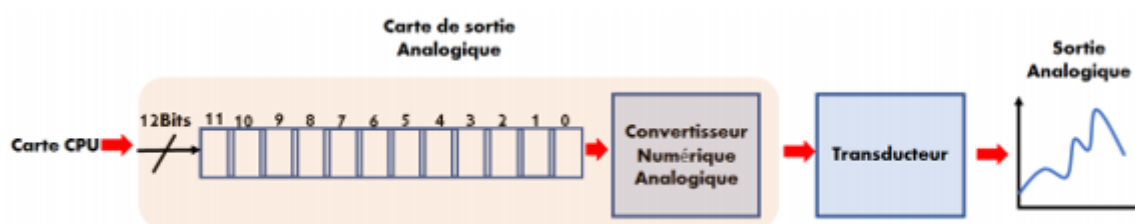


FIGURE 2.16 – Sortie analogique.[12]

Le tableau (2.3) répertorie certaines des caractéristiques nominales standard utilisées dans les automates programmables dotés de capacités de sortie analogique.

| Valeurs des sorties électriques |
|---------------------------------|
| + 0 à +5 V(DC) |
| + 0 à +10 V(DC) |
| ±2.5 V(DC) |
| ±5 V(DC) |
| ±10 V(DC) |

TABLE 2.3 – Valeurs électriques standard des sorties analogiques.[12]

iv. Connexions électriques des sorties analogiques :[12]

Les cartes de sorties analogiques sont disponibles dans des configurations allant de 2 à 8 sorties par carte, mais en moyenne, la plupart des cartes disposent de 4 à 8 canaux de sorties analogiques. Ces canaux peuvent être configurés comme sorties asymétriques ou différentielles.

2.3.6 Cartes d'entrées/sorties spéciaux [12]

Les cartes spéciales des d'E/S assurent la liaison entre les automates programmables et les appareils nécessitant des types de signaux particuliers. Ces signaux spéciaux, qui diffèrent des signaux analogiques et numériques standards, ne sont pas très courants et ne surviennent que dans 5 à 10% des applications d'automates. Cependant, sans ce type de cartes, les processeurs (CPU) ne seraient pas en mesure d'interpréter ces signaux et de mettre en œuvre des programmes de contrôle, on citant par exemple, les cartes de régulation PID, carte de comptage rapide, les cartes de position, les cartes ASCII, les cartes réseaux pour les E/S distants, les cartes de traitement basé sur la logique floue...etc...

Les cartes spéciaux d'E/S peuvent être divisées en deux catégories :

- Les cartes d'action directe.
- Les cartes intelligentes.

a. Les cartes d'actions directes :

Les cartes d'E/S à action directe sont des cartes qui se connectent directement aux capteurs d'entrée ou avec les actionneurs de sortie. Ces cartes prétraitent les signaux d'entrée ou de sortie et transmettent ces informations prétraitées directement au processeur de l'API (voir Figure 2.17). Ce type des cartes d'E/S comprennent des cartes qui prétraitent généralement les signaux de faible tension comme dans le cas des thermocouples et rapides comme la grandeur de vitesse, que les autres cartes d'E/S standard ne peuvent pas le lire.

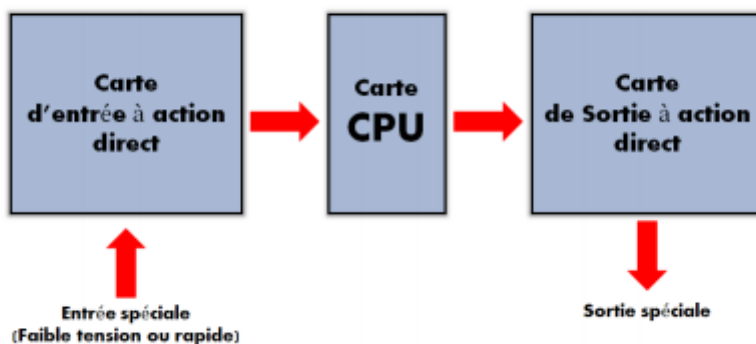


FIGURE 2.17 – Cartes d'action directe.[12]

b. Cartes intelligentes :

Des cartes d'E/S intelligentes à fonctions spéciales incorporent des microprocesseurs intégrés pour ajouter le facteur d'intelligence. Ces cartes intelligentes peuvent effectuer des tâches de traitement complètement indépendantes du processeur CPU et la fonction scan l'API (voir Figure 2.18). Ils peuvent aussi avoir des commandes numériques, ainsi que analogiques.

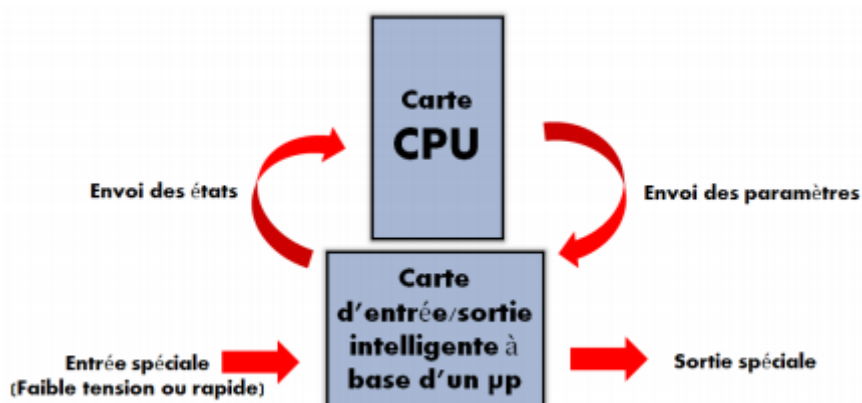


FIGURE 2.18 – Cartes intelligentes.[12]

2.4 Avantages des automates fixes

Les automates fixes présentent de grands avantages par rapport aux autres types d'automates. Ils ont été spécialement conçus pour répondre aux besoins des petits projets d'automatisation bas de gamme. Les avantages d'un automate fixe sont les suivants :

- * De petite taille, ils ne prennent pas beaucoup de place dans un boîtier.
- * Rapide et facile à monter.
- * Leur coût est faible, ce qui en fait une solution économique pour les applications de base.

[13]

2.5 Inconvénients des automates fixes

Si vous envisagez de choisir un automate fixe pour automatiser votre prochaine application, il est prudent de prendre en compte certains de ses inconvénients pour vous assurer que vous choisissiez le type d'automate qui convient à votre application. Les inconvénients d'un automate fixe sont les suivants :

- * La puissance de traitement de l'unité centrale est faible et la mémoire est petite, de sorte que les tâches complexes peuvent être difficiles à réaliser.
- * Inflexible car le nombre d'interfaces d'entrée, de sortie et de communication est fixe.
- * Ne convient que pour les applications de base avec un petit nombre d'entrées et de sorties. [13]

2.6 Les langages de Programmation des APIs

Les langages de programmation utilisés pour les automates programmables évoluent depuis la création des APIs à la fin des années 1960. Dans ce contexte, la commission internationale d'électrotechnique (CIE) définit cinq langages de programmation des automates programmables (IEC 61131-3), celles-ci consistent en deux langages textuels, (IL : liste d'instructions) et (SCL : Structured Control Language), et trois langages graphiques, (LD : schéma à contacts), (FBD : schéma fonctionnel) et (SFC : Sequential Function Chart) sont définis pour structurer l'organisation interne des programmes et des blocs fonctionnels.[12]

2.6.1 Langage GRAFCET

- **Définition : (Graphe Fonctionnel de Commande Étapes-Transitions)**

Le grafcet est un diagramme fonctionnel ; il représente par un graphe le fonctionnement de la partie opérative, donc les actions effectuées par le système. Il nous servira ensuite à décrire le fonctionnement de la partie commande, c'est-à-dire la technologie employée pour commander les actionneurs.[9]

- **Domaine d'application :** Le diagramme fonctionnel est indépendant des techniques séquentielles "tout ou rien", pneumatique, électrique ou électronique, câblées ou programmées, pouvant être utilisées pour réaliser l'automatisme de commande. Mais l'utilisation de séquenceurs, d'une part, et d'automates à instructions d'étapes d'autre part, permet une transcription directe du diagramme fonctionnel. Cette représentation graphique concise et facile à lire est aisément compréhensible par toute personne en relation avec le système automatisé, du concepteur à l'utilisateur sans oublier l'agent de maintenance. Utilisé industriellement, le GRAFCET est aussi enseigné dans les options techniques et l'enseignement supérieur. Depuis les premières publications le concernant et surtout depuis la norme française NF C03-190 de 1982, cet outil a été travaillé et enrichi par le groupe systèmes logiques de l'AF CET (Association Française pour la Cybernétique Economique et Technique). Il existe une documentation et symboles graphiques, diagramme fonctionnel "Grafcet" éditée par l'Union Technique de l'Electricité. UTE C03-190 Nov. 1990.[10]

- **Principe du grafcet :** Pour visualiser le fonctionnement de l'automatisme, le GRAFCET utilise une succession alternée d'étapes et de transitions. A chaque étape correspond une ou plusieurs actions à exécuter. Une étape est soit active, soit inactive. Les actions associées à cette étape sont effectuées lorsque celle-ci est active. Les transitions indiquent avec les liaisons orientées, les possibilités d'évolution entre étapes.

A chaque transition est obligatoirement associée une condition logique pouvant être vraie ou fausse. Cette condition de transition est appelée réceptivité. L'évolution d'une étape à une autre ne peut s'effectuer que par le franchissement d'une transition. Une transition ne peut être franchie, donc activer l'étape suivante que :

- > Si elle est validée par l'étape antérieure active.
- > Les conditions de réceptivité soient satisfaites.

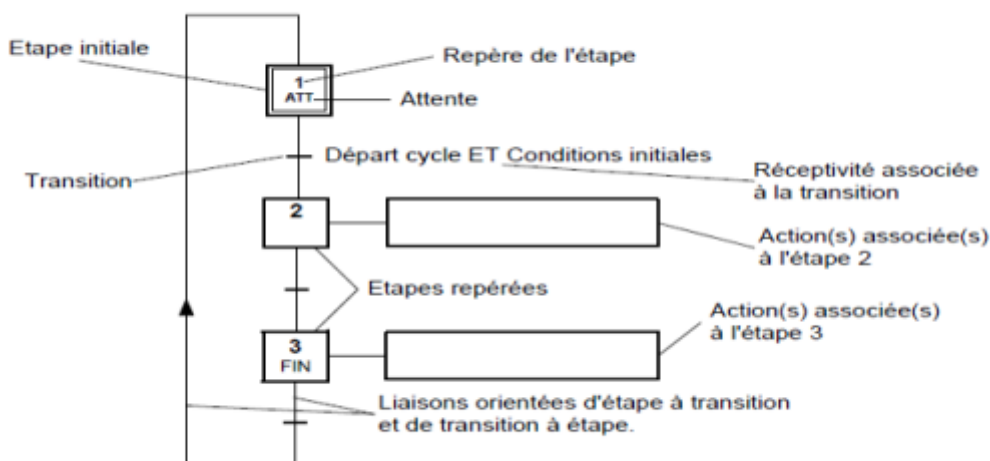


FIGURE 2.19 – Principe de grafcet.[10]

Etape initiale : représente une étape qui est active au début du fonctionnement. Elle se différencie de l'étape en doublant les côtés du carré.

Transition : la transition est représentée par un trait horizontal.

Réceptivité : les conditions de réceptivité sont inscrites à droite de la transition.

Etape : chaque étape est représentée par un carré repéré numériquement.

Action(s) : elles sont décrites littéralement ou symboliquement à l'intérieur d'un ou plusieurs rectangles reliés par un trait à la partie droite de l'étape.

Liaisons orientées : indique le sens du parcours.[10]

• **Règles d'évolution du grafcet :**

• **Règle 1 :** L'initialisation précise les étapes actives au début du fonctionnement. Elles sont activées inconditionnellement et repérées sur le GRAFCET en doublant les côtés des symboles correspondants.



FIGURE 2.20 – Etape initiale.[10]

• **Règle 2** : Une transition est soit validée soit non validée. Elle est validée lorsque toutes les étapes immédiatement précédentes sont activées. Elle ne peut être franchie que :

- Lorsqu'elle est validée.
- La réceptivité associée à la transition est vraie. La transition est alors obligatoirement franchie.

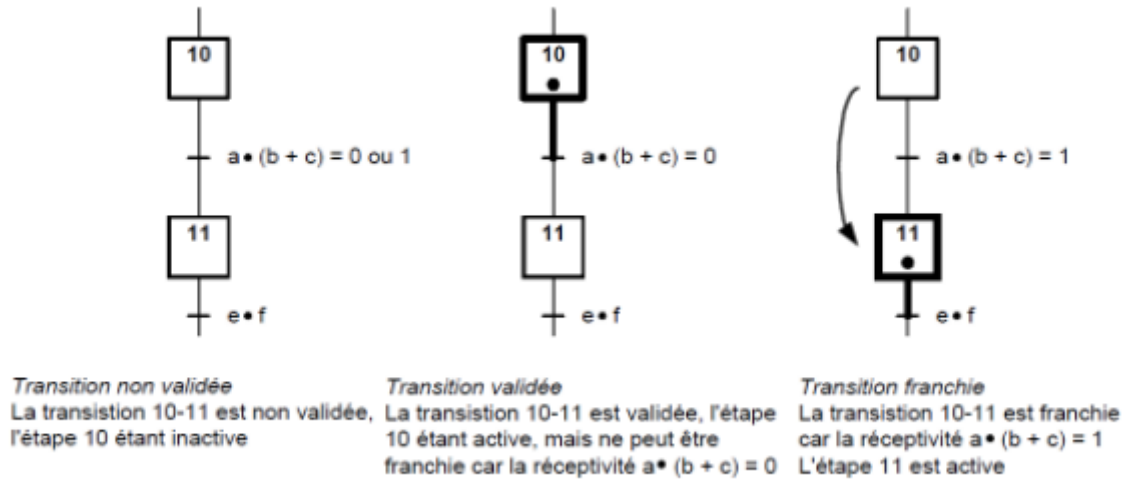


FIGURE 2.21 – Franchissement d'une transition.[10]

• **Règle 3** : Le franchissement d'une transition entraîne l'activation de toutes les étapes immédiatement suivantes et la désactivation de toutes les étapes immédiatement précédentes. Cette évolution du GRAFCET est donc synchrone. Il y a évolution asynchrone lorsque le franchissement de la transition entraîne l'activation des étapes suivantes et que c'est la vérification de cette activation qui autorise la désactivation des étapes précédentes.

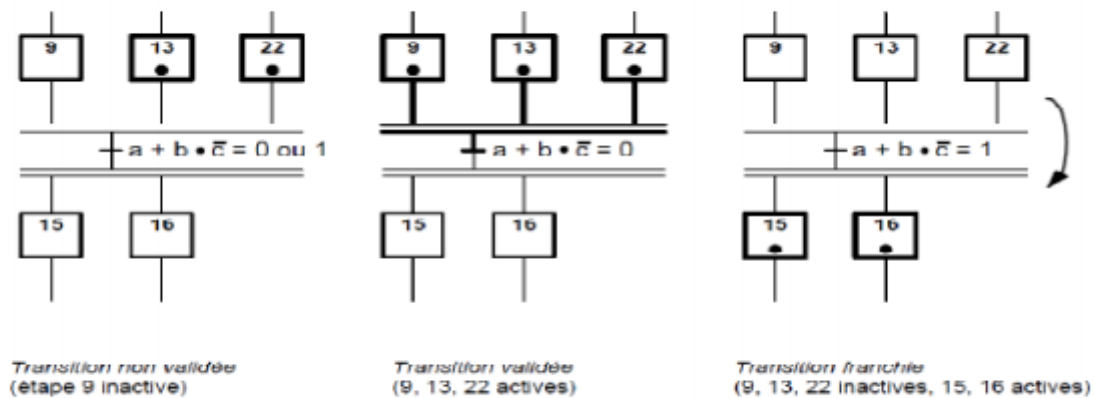


FIGURE 2.22 – Evolution des étapes active.[10]

- **Règle 4 :** Plusieurs transitions simultanément franchissables sont simultanément franchies.

- **Règle 5 :** Si au cours du fonctionnement, une même étape doit être désactivée et activée simultanément, elle reste activée. L'activation doit être prioritaire sur la désactivation au niveau d'une même étape.

Remarque :

La durée de franchissement d'une transition ne peut jamais être rigoureusement nulle, même si, théoriquement (règles 3 et 4), elle peut être rendue aussi petite que possible. Il en est de même de la durée d'activation d'une étape. En outre, la règle 5 se rencontre très rarement dans la pratique. Ces règles ont été ainsi formulées pour des raisons de cohérence théorique interne au GRAFCET.[10]

2.6.2 Langage CONTACT (LADDER)

- **Définition :**

Le langage à contacts (LD : Ladder Diagram) est composé de réseaux lus les uns à la suite des autres par l'automate. Ces réseaux sont constitués de divers symboles représentant les entrées/sorties de l'automate, les opérateurs séquentiels (temporisations, compteurs, ...), les opérations, ainsi que les bits systèmes internes à l'automate (ces bits permettent d'activer ou non certaines options de l'automate, telle que l'initialisation des grafkets).[9]

- **Les symboles utilisés :**

Il existe 3 types d'éléments de langage :

- les entrées (ou contact), qui permettent de lire la valeur d'une variable booléenne,
- les sorties (ou bobines) qui permettent d'écrire la valeur d'une variable booléenne,
- les blocs fonctionnels qui permettent de réaliser des fonctions avancées.[8]

Le tableau 2.4 donne les principaux éléments (contacte et bobines) d'un réseau LD.

a. Les principaux éléments :

| Object graphique | Nom |
|------------------|---|
| - - | 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 inconditionnel (vers le sous-programme appelant) |
| -cond-<return> | Retour conditionnel |
| -»Label | Saut inconditionnel |
| -cond-»Label | Saut conditionnel |

TABLE 2.4 – Les principaux éléments (contacte et bobines) d'un réseau LD.[8]

b. Les blocs fonctionnels (Les circuits séquentiels) :

- Les blocs temporisations possèdent une entrée I reliée aux éléments graphiques précédents et une sortie activée lorsque le temps écoulé depuis l'activation de la temporisation atteint la valeur prédéfinie.

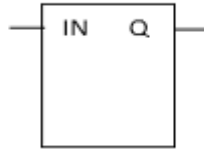


FIGURE 2.23 – Bloc temporisation.[9]

- Les fonctions comptage/décomptage peuvent être séparées ou réunies dans un seul bloc selon les marques. CU est l'entrée de comptage sur front montant, CN est l'entrée de décomptage front montant, R est l'entrée de remise à zéro de la valeur courante et S ou LD est l'entrée de chargement de la valeur prédéfinie. D est la sortie lorsque la valeur prédéfinie ou le sont atteints selon que l'on compte ou que l'on décompte.

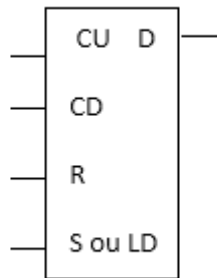


FIGURE 2.24 – Fonction comptage/décomptage.[9]

- Les monostables ne sont pas utilisés dans toutes les marques. S est l'entrée d'activation sur front montant du monostable et R sa sortie logique.

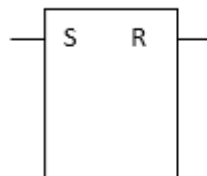


FIGURE 2.25 – Monostable.[9]

- Registre télémechanique. R est l'entrée de remise à zéro du registre, I est l'entrée stockage sur front montant et O est l'entrée déstockage sur front montant. La sortie E indique que le registre est vide et la sortie F qu'il est plein.

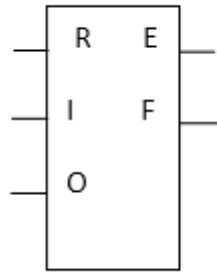


FIGURE 2.26 – Registre télémechanique. [9]

- Générateur d'impulsions SIEMENS. EN est l'entrée qui permet d'activer le train d'impulsions. ENO est la sortie qui permet de relier plusieurs générateurs en série plutôt qu'en parallèle ($ENO = EN$). Q0.X est la sortie du train d'impulsions. Ne peuvent être utilisés pour cette fonction que les sorties Q0.0 et Q0.1..

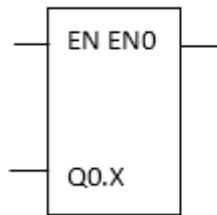


FIGURE 2.27 – Générateur d'impulsions SIEMENS. [9]

- **Règles d'évolution d'un réseau de contacts :**

La lecture d'un réseau se fait réseau connexe par réseau connexe (de haut en bas), puis de gauche à droite à l'intérieur d'un réseau connexe. Un réseau connexe est constitué d'éléments graphiques tous reliés entre eux, mais indépendants des autres éléments graphiques du réseau. Si l'on rencontre une liaison verticale de convergence, on évalue d'abord le sous-réseau qui lui est associé (toujours dans la même logique) avant de continuer l'évaluation du sous-réseau qui l'englobe.

Exemple :

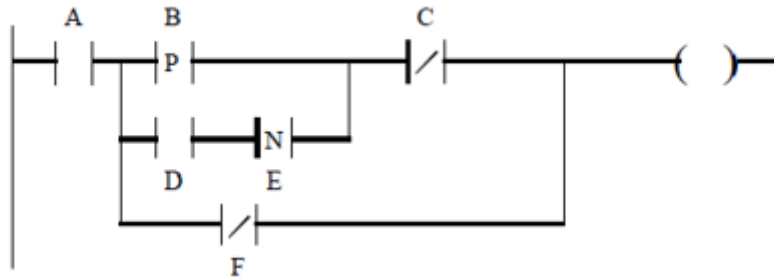


FIGURE 2.28 – Exemple d'un réseau Ladder. [9]

L'ordre d'exécution des éléments de ce réseau est le suivant :

- 1ère phase : lecture des contacts A et B jusqu'à la rencontre de la 1ère liaison verticale de convergence entre les contacts B et C.
- 2ème phase : lecture du premier sous-réseau, contacts D et E.
- 3ème phase : reprise de lecture de la première ligne du réseau connexe, contact E, jusqu'à la rencontre de la deuxième liaison verticale de convergence.
- 4ème phase : lecture du deuxième sous-réseau, contact F.
- 5ème phase : lecture de la bobine.

Remarque :

la mise à jour des sorties s'effectue en fin de cycle, après la lecture de tout le programme.[9]

2.6.3 Langage LOGIGRAMME

• **Définition :**

Un réseau LOG est composé d'une ou plusieurs boîtes d'opérations LOG. Au lieu d'utiliser des contacts, on affecte une ou plusieurs valeurs binaires comme entrées à une boîte d'opération LOG. Vous utilisez les sorties de l'opération pour connecter cette dernière à une opération consécutive ou pour achever le réseau. Ainsi, une seule opération LOG peut représenter la même fonction qu'un ensemble de contacts, bobines ou boîtes en schéma à contacts. Le réseau est achevé lorsque vous avez procédé à l'affectation de tous les paramètres de l'opération ou que vous les avez connectés à une autre opération.[9]

- Les symboles utilisés :

a) Les opérations sur bits :

- Cette boîte représente la fonction ET en associant deux bits Que l'on peut inverser à l'entrée de la boîte. Il est possible de rajouter jusqu'à 32 entrées.

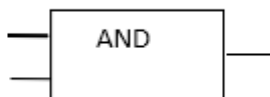


FIGURE 2.29 – Fonction ET. [9]

- Cette boîte représente la fonction OU en associant deux bits Que l'on peut inverser à l'entrée de la boîte. Il est possible de rajouter jusqu'à 32 entrées.

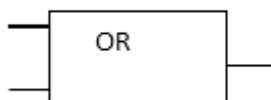


FIGURE 2.30 – Fonction OU. [9]



FIGURE 2.31 – Contact à front montant/descendant. [9]

- L'opération Sortie écrit la nouvelle valeur du bit de sortie dans la mémoire image.

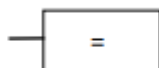


FIGURE 2.32 – Opération sortie. [9]

- L'opération SET met à 1 un nombre N de sorties à partir de l'adresse bit indiquée.

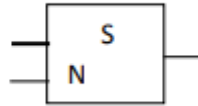


FIGURE 2.33 – Opération SET. [9]

- L'opération RESET met à 0 un nombre N de sorties à partir de l'adresse bit indiquée.

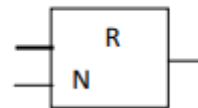


FIGURE 2.34 – Opération RESET. [9]

- Les opérations de transfert permettent de transférer l'octet d'entrée (IN) dans l'octet de sortie (OUT). Cette fonction est utilisable avec des mots (MOV-W), des doubles mots (MOV-DW), des réels (MOV-R), ...

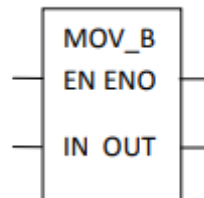


FIGURE 2.35 – Opération de transfert. [9]

- Les blocs de conversion permettent de convertir un nombre d'une base donnée dans une autre base (vu dans le module 2). Il est également possible de tronquer ou arrondir un nombre, de l'encoder ou le décoder.

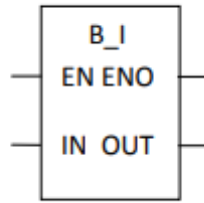


FIGURE 2.36 – Bloc de conversion. [9]

b) Les circuits séquentiels :

- Il existe trois types de temporisations : TON (retard à la montée), TONR (retard à la montée temporisé), TOF (retard à la descente). IN est l'entrée de validation et PT le temps prédéfini. La base de temps dépend du numéro de la temporisation.

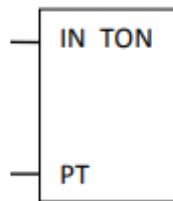


FIGURE 2.37 – Type de temporisation TON. [9]

- Il existe trois principaux types de compteurs/décompteurs : CTU (compteur incrémental), CTD (compteur décrémental), CTUD (compteur incrémental/décrémental). PV est la valeur prédéfinie et R la remise à zéro. L'entrée d'incrémental s'appelle CU et l'entrée de décrémentation CD.

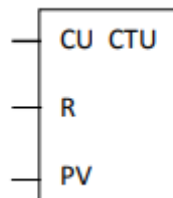


FIGURE 2.38 – Type de compteur CTU. [9]

- Le bloc PLS permet de générer des trains d'impulsions de rapport cyclique 50% (PTO) ou bien modulable (PWM).

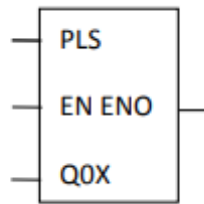


FIGURE 2.39 – Bloc PLS. [9]

c) Les blocs comparaison :

Ces blocs permettent de comparer des nombres, des bits, des octets ou des mots en supériorité, infériorité ou égalité.

d) Les opérations d'exécution :

Ces blocs regroupent plusieurs fonctions établissant l'ordre d'exécution du programme : saut de programme, retour de sous-programme, fin de programme, ... [9]

2.6.4 Langage LIST [9]

• Définition :

Langage basique des automatismes il représente une liste d'instructions qui met en œuvre, comme pour le langage booléen, des équations logiques.

Il permet également de résoudre quelques calculs numériques. Bien que les différents programmes en liste d'instructions des différents constructeurs d'API possèdent tous leurs spécificités, ils sont néanmoins structurés de la façon suivante. Chaque instruction est composée d'un code instruction et d'un opérande.

• Les instructions de base : [9]

a) Instructions de test :

- LD : contact normalement ouvert.
- LDN : contact normalement fermé.
- LDR ou EU : contact à front montant.
- LDF ou ED : contact à front descendant.
- AND : liaison série (ET) à un contact normalement ouvert.
- ANDN : liaison série (ET) à un contact normalement fermé.
- ANDR : liaison série (ET) à un contact à front montant.
- ANDF : liaison série (ET) à un contact à front descendant.

- OR : liaison parallèle (OU) à un contact normalement ouvert.
- ORN : liaison parallèle (OU) à un contact normalement fermé.
- ORR : liaison parallèle (OU) à un contact à front montant.
- ORF : liaison parallèle (OU) à un contact à front descendant.

Remarque : Les instructions AND et OR (et leurs dérivés) peuvent utiliser des parenthèses. Ces parenthèses permettent de traduire des schémas à contact de façon simple. Il est possible d'imbriquer plusieurs niveaux de parenthèses.

b) Instructions d'action :

- ST : bobine directe. L'objet bit associé prend la valeur du résultat de la zone test.
- STN : bobine inversée. L'objet bit associé prend la valeur inversée du résultat de la zone test.
- S : bobine d'enclenchement. L'objet bit associé est mis à 1 lorsque la valeur du résultat de la zone test est à 1.
- R : bobine de déclenchement. L'objet bit associé est mis à 0 lorsque la valeur du résultat de la zone test est à 1.

c) Instructions de saut :

- JMP : saut de programme inconditionnel.
- JMPC : saut de programme si le résultat de l'instruction test précédente est à 1.
- JMPCN : saut de programme si le résultat de l'instruction test précédente est à 0.
- SRN : branchement en début de sous-programme.
- RET : retour de sous-programme inconditionnel.
- RETC : retour de sous-programme si le résultat de l'instruction test précédente est à 1.
- RETCN : retour de sous-programme si le résultat de l'instruction test précédente est à 0.

d) Instructions d'arrêt :

- END : fin de programme inconditionnelle.
- ENDC : fin de programme si le résultat de l'instruction test précédente est à 1.
- ENDCN : fin de programme si le résultat de l'instruction test précédente est à 0.
- HALT : Arrêt de l'exécution du programme.

e) Opérations de transfert (SIEMENS) :

- MOVB : transfert d'un octet dans un autre.
- MOVW : transfert un mot dans un autre.

- MOVD : transfert d'un double mot dans un autre.
- MOVR : transfert d'un double mot réel dans un autre.[9]

2.6.5 Langage SCL [14]

- **Définition :**

Le langage littéral structuré est constitué d'expression littérale composée d'opérateur et d'opérande libellé conformément à la norme. Les opérateurs booléens du langage ST sont ressemblant aux précédents, on retrouve : AND, OR, XOR, et NOT .

- **L'écriture du programme :**[14]

Se fait sous forme d'énoncés littérale. Une ligne de programmation peut avoir la structure générale suivante :

Commentaires

Etiquettes : énoncé

- **AFFECTATION :**

L'énoncé d'affectation remplace la valeur actuelle d'une variable par le résultat de l'évaluation d'une expression. Un énoncé d'affectation doit se composer d'une référence de variable à gauche, suivie de l'opérateur d'affectation (:=), suivi de l'expression à évaluer.

- **SELECTION :**[14]

Les énoncés de sélection comprennent les énoncés IF et CASE :

* **L'énoncé IF** :indique qu'un groupe d'énoncés ne doit être exécuté que si la valeur prise par l'expression booléenne associée est 1 (vraie) ; THEN. Si la condition n'est pas vérifiée, soit aucun énoncé n'est exécuté, soit le groupe d'énoncés suivi du mot clé ELSE doit être exécuté.

* **L'énoncé CASE** : se compose d'une expression qui doit être évaluée sur une variable de type INT (que J'on appelle le "sélecteur"), et d'une liste de groupes d'énoncés, chacun d'entre eux étant étiqueté par une ou plusieurs plages de valeurs entières. Cet énoncé précise que le premier groupe d'énoncés, dont l'une des plages contient la valeur calculée du sélecteur, doit être exécuté. Si la valeur du sélecteur n'apparaît dans aucune plage d'aucun cas, la suite d'énoncés qui suit le mot clé ELSE. Dans le cas contraire aucune des suites d'énoncés ne doit être exécuté.

- **ITERATION** : Les énoncés d'itération comprennent les énoncés FOR, WHILE et REPEAT :

* **L'énoncé FOR** : indique qu'une suite d'énoncés doit être exécutée de manière répétitive jusqu'à ce que le mot clé END-FOR soit atteint, une suite de valeurs progressives étant affectée

à la variable de commande de la boucle FOR. La variable de commande, la valeur initiale et la valeur finale doivent être des expressions du même type entier (SINT, INT ou DINT).

* **L'énoncé WHILE** : augmente ou diminue la variable de commande d'une valeur initiale jusqu'à une valeur finale par incréments déterminés par la valeur d'une expression ; par défaut, cette valeur est 1. L'énoncé WHILE déclenche l'exécution répétée de la suite d'énoncés s'étendant jusqu'au mot clé END-WHILE, jusqu'à ce que l'expression booléenne associée soit fausse. Si, à l'origine, l'expression est fausse, le groupe d'énoncés n'est pas exécuté du tout.

* **L'énoncé REPEAT** : entraîne l'exécution répétée (et au moins une fois) de la suite d'énoncés s'étendant jusqu'au mot clé UNTIL, jusqu'à ce que la condition booléenne associée soit vraie. [14] L'énoncé EXIT doit permettre de sortie de toutes les boucles d'itérations.

2.7 Description de la station tri

Le tri fait partie de la fonction de manipulation des quantités changeantes. La section de convoyeur peut être dérivée pour le tri, les différentes portes de tri étant commutées en fonction de la pièce à usiner. Les pièces doivent procéder individuellement pour ne pas altérer les fonctions de commutation des portes. Dans la station de tri, les pièces symboliques sont triées en fonction de la matière et de la couleur. Les vérins pneumatiques assemblés sont triés en fonction de la couleur et du diamètre du piston. [11]

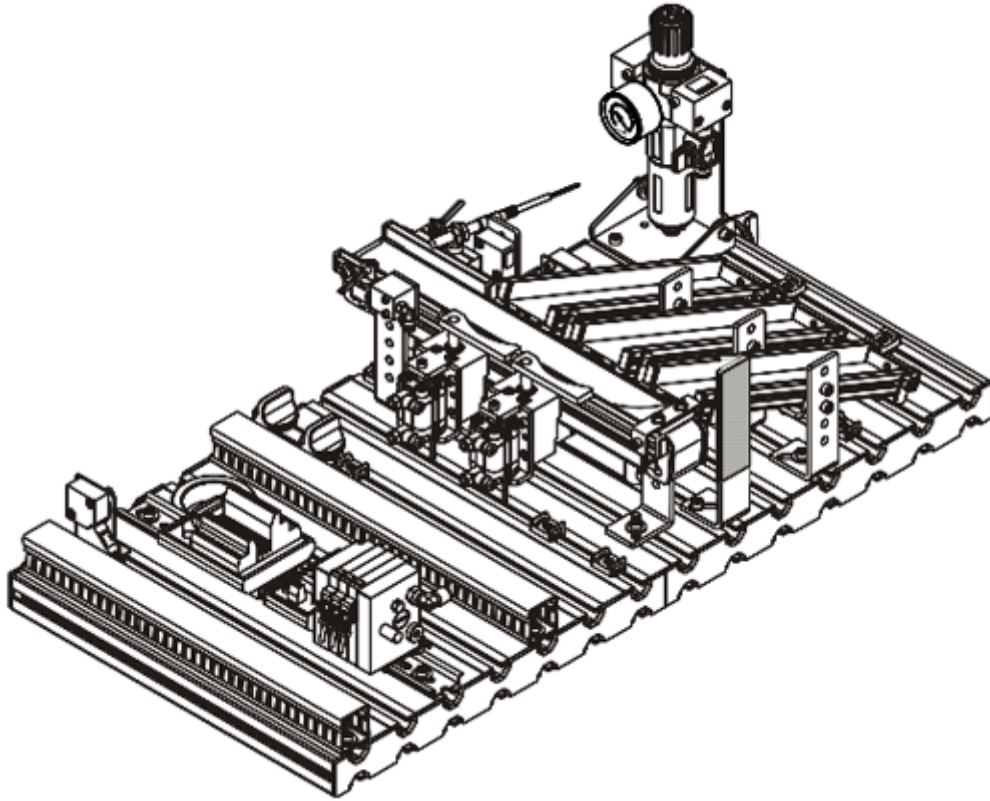


FIGURE 2.40 – Vue de la station de tri. [11]

La fonction de la station de tri est de Trier les pièces selon leurs caractéristiques. La station de tri comprend les éléments suivants :

- * Module de convoyeur de tri
- * Module de glissière
- * Plaque de profil
- * Chariot
- * Console de commande
- * Carte PLC [11]

2.8 Les composants

2.8.1 Module de Convoyeur

Le module convoyeur est utilisé pour transporter des pièces de l'usine vers les glissières.

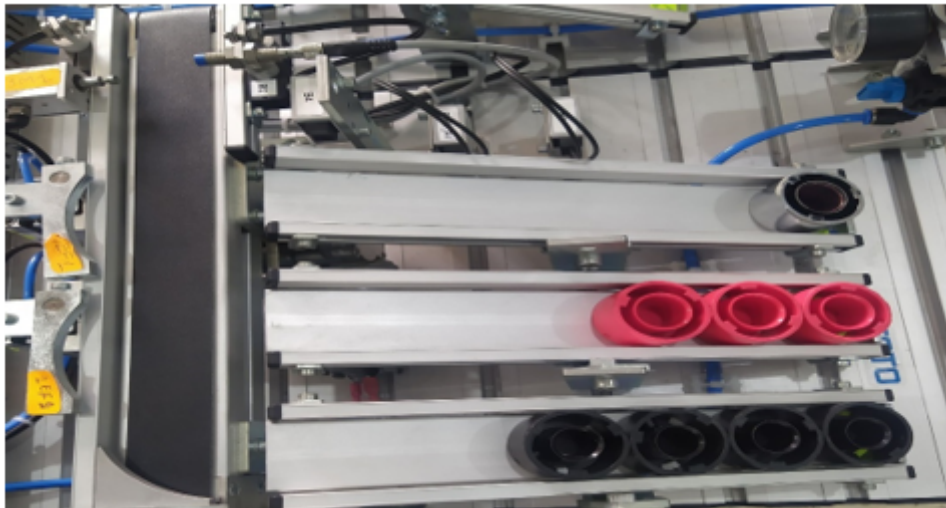


FIGURE 2.41 – Convoyeur et les glissières de la station.

2.8.2 Les entrées et les sorties de la station tri

Les entrées et les sorties qui sont claires se trouvent dans la figure 2.42.

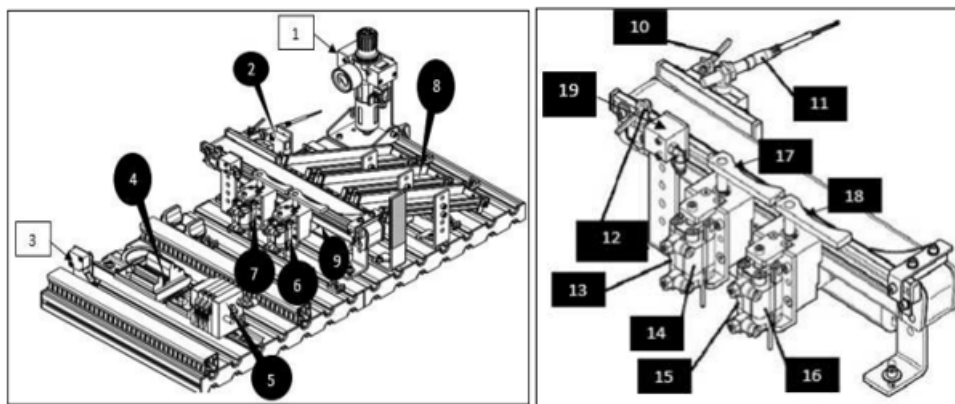


FIGURE 2.42 – Vue détail de convoyeur et de la station de tri. [11]

Puis Toutes les choses liées à la bande transporteuse ou à côté de la bande transporteuse, nous les rassemblons dans le tableau suivant :

| N° | Code | Désignations | Types | Utilités |
|----|------------|-------------------|-------------------------|---|
| 1 | FR | Filtre Régulateur | Pneumatique | Filtre régler le débit de l'air comprimé |
| 2 | B4 | Capteur | Optique à réflex | Glissière pleine |
| 3 | IP-FI | Capteur | Optique de proximité | (Non disponible) |
| 4 | PA1 | Pré-Actionnaire | Électrique | Distribution de l'énergie électrique |
| 4 | PA2 | Pré-Actionnaire | Pneumatique | Distribution de l'énergie Pneumatique |
| 6 | 1M1 | Actionnaire | Pneumatique | Sortir la dérivation 1 |
| 7 | 2M1 | Actionnaire | Pneumatique | Sortir la dérivation 2 |
| 8 | C1 /C2 /C3 | Buffer-effecteur | Buffer gravitationnelle | Stockage des pièces Noirs /Métalliques/Rouges |
| 9 | 3M1 | Actionnaire | Pneumatique | Arrêter la pièce pendant l'inspection |
| 10 | K1 | Actionnaire | Électrique | Moteur de la bande activé |
| 11 | B3 | Capteur | Optique de proximité | Pièce à usiner autre que noire |
| 12 | B2 | Capteur | Inductif | Pièce en métal |
| 13 | Part-AV | Capteur | Optique de proximité | Pièce présente |
| 14 | 1B1 | Capteur | Électromagnétique | Sortir la dérivation 2 |
| 15 | 1B2 | Capteur | Électromagnétique | Dérivation 1 sortie |
| 16 | 2B1 | Capteur | Électromagnétique | Dérivation 2 rentrée |
| 17 | 2B2 | Capteur | Électromagnétique | Dérivation 2 sortie |
| 18 | 1EF1 | Effecteur | Pneumatique | Orientation pièces Noirs |
| 19 | 1EF2 | Effecteur | Pneumatique | Orientation pièces Métalliques |

TABLE 2.5 – Entrées / Sorties de la station tri.

2.8.3 Module de barrage (stoppeur)

Il arrête la pièce pendant quelques secondes pour terminer le processus Identification de la couleur et du matériau.



FIGURE 2.43 – Module de barrage de la station.

2.8.4 Module de dérivation

Le module de dérivation ou de séparation pneumatique est installé directement sur le convoyeur. Il utilise des capteurs pour spécifier la livraison des pièces.



FIGURE 2.44 – Module de dérivation de la station.

2.8.5 Pupitre de commande



FIGURE 2.45 – Pupitres des commandes (station tri).

L'ensemble de la station est contrôlé au moyen du panneau de commande. Dans la figure 2.45 nous montrerons les pupitres de commande du poste de tri et par conséquent le tableau 2.6 des variables :

| Code | Désignations | Types | Utilités |
|------------------|--------------|------------------|-----------------------------------|
| S1-T Partie Trie | Capteur | Électromécanique | Touche start |
| S2-T Partie Trie | Capteur | Électromécanique | Touche stop (contact à ouverture) |
| S3-T Partie Trie | Capteur | Électromécanique | Sélecteur automatique/manuel |
| S4-Partie Trie | Capteur | Électromécanique | Touche mise en référence/reset |
| P1 | Actionnaire | Electrique | Lampe start |
| P2 | Actionnaire | Electrique | Lampe reset |
| P3 | Actionnaire | Electrique | Buffers pleins |

TABLE 2.6 – Entrées/sorties (Partie Commande).

2.9 Conclusion

Dans ce chapitre, nous avons vu les automates programmables industriels et leurs architectures, des moyens de communication, des langages de programmation utilisé pour programmer ce système, ensuite nous avons entamé la présentation et la description du système étudié.

CHAPITRE

3

L'AUTOMATISATION DE NOTRE STATION
DE TRI

3.1 Introduction

Dans ce chapitre, nous allons décrire et réaliserons le programme d'intégration de l'API. Pour ce faire, nous utiliserons le programme de conception de logiciel pour les systèmes d'automatisation SIMATIC STEP7, passant par la configuration des automates en respectant le matériel réel adéquat et ainsi la programmation de notre station via le modèle de programmation grafset sur l'outil step7.

3.2 Cahier des charges

Actuellement, Cette station procède au tri de pièces dans trois glissières de la manière suivante :

- L'arrive des pièces aléatoirement (rouge, noire, métallique).
 - La présence de pièce se détecte selon le capteur PART-AV.
 - Démarrage de tapis de convoyeur par un moteur K1 lorsque le capteur PART-AV détecte la pièce.
 - Le stoppeur 3M1 stoppe la pièce et attend l'ordre selon les capteurs (B2, B3).
- **Si $B3.\overline{B2}$** :
- Le stoppeur 3M1 rentrer et 1M1 tourner 1EF1, et alors la pièce va à buffer C1(Orientation des pièces rouges vers C1).
- **Si $B3.B2$** :
- Le stoppeur 3M1 rentrer et 2M1 tourner 1EF2, et alors la pièce va à buffer C2(Orientation des pièces métalliques vers C2).
- **Si $\overline{B3} .\overline{B2}$** :
- 3M1 rentrer et la pièce noire va à buffer C3(Orientation des pièces noires vers C3).

3.3 Problématique

Lorsque le capteur B4 détecte l'un de ces buffers sature le système s'arrête d'une manière automatique avec la non exploitation des deux autres Buffers. Comme nous avons remarqué qu'il y'a une grande perte à vide d'énergie pendant le fonctionnement du système ex la sortie de l'effectuer 1M1 avant l'arrivée de la pièce.

3.3.1 Solutions

Notre solution proposée pour le nouveau cahier des charges est basée sur :

- Au début nous avons supposé que nous ayons 18 pièces de trois types différents : (6 rouges, 6 noires, 6 métalliques) :
- La glissière C1 doit recevoir 6 pièces rouges.
- La glissière C2 doit recevoir 6 pièces noires.
- La glissière C3 doit recevoir 6 pièces métalliques.

Cette affectation dans le but d'exploiter les trois buffers et atteindre un stock et livraison maximum.

D'autre part la demande de nos exécutions via les actions adéquates seront bien précises afin d'optimiser l'énergie utilisée.

3.4 SIMATIC STEP7

3.4.1 Définition logiciel STEP7

Step7 est le progiciel de base pour la configuration et la programmation de systèmes d'automatisation SIMATIC, il fait partie de l'industrie logicielle SIMATIC. Il existe en plusieurs versions dans laquelle on a utilisé la version 5.5. Pour des applications sur SIMATIC S7-300/400, le step7 présente des fonctionnalités supplémentaires, tel que[16] :

- Possibilité d'extension grâce aux applications proposées par l'industrie logicielle SIMATIC.
- Possibilité de paramétrage de modules fonctionnels et de modules de communication.
- Forçage et fonctionnement multiprocesseur .
- Communication par données globales.
- Transfert de données commandé par événement à l'aide de blocs de communication et de blocs fonctionnels.
- Configuration de liaisons.

3.4.2 Les étapes fondamentales d'un projet d'automatisation sous Step7 :

a. Création d'un projet par step7

Après le démarrage du Windows, nous trouvons dans l'interface Windows une icône SIMATIC

manager qui nous permet d'accéder au logiciel Step7 :



FIGURE 3.1 – Fenêtre SIMATIC Manager.

Le démarrage de ce dernier nécessite un double-clic sur l'icône de la figure 3.1 qui nous permet d'ouvrir la fenêtre fonctionnelle représentée dans la figure ci-dessous :

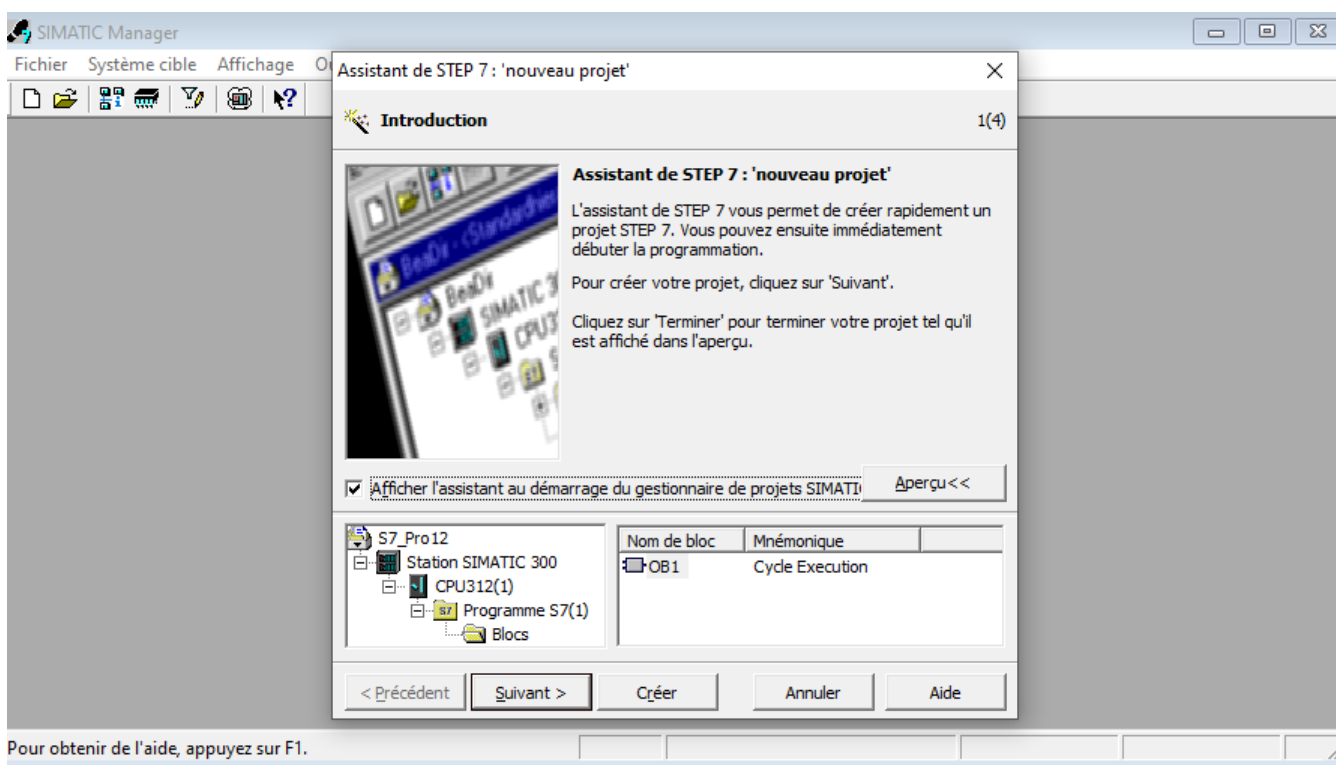


FIGURE 3.2 – Fenêtre de démarrage de Step7.

Le Clic sur l'icône « fichier » dans la barre de menu. Après cette action, nous cliquons sur nouveau et choisir le nom de projet par exemple « bachir et Walid ». En fin, nous faisons un clic sur ok. Cette méthode nous permet de choisir le type de CPU, afin de faire une configuration matérielle.

b. Configuration matérielle :

La configuration la station de tri sous l'outil step7. C'est une étape très importante qui consiste à l'agencement des châssis des modules.

Le but de La configuration matérielle est la configurer et paramétrer le support matériel ; ainsi la configurer de la liaison de communication.

Pour une bonne configuration matérielle, vous devez suivre les étapes suivantes :

- Insertion d'une nouvelle station :

Dans cette étape, nous pressons sur le bouton droit de la souris, puis cliquer sur « insérer un nouvel objet » et nous choisissons « Station SIMATIC 300 » la station utilisé dans notre cas. Ces étapes représentent dans La figure ci-dessous :

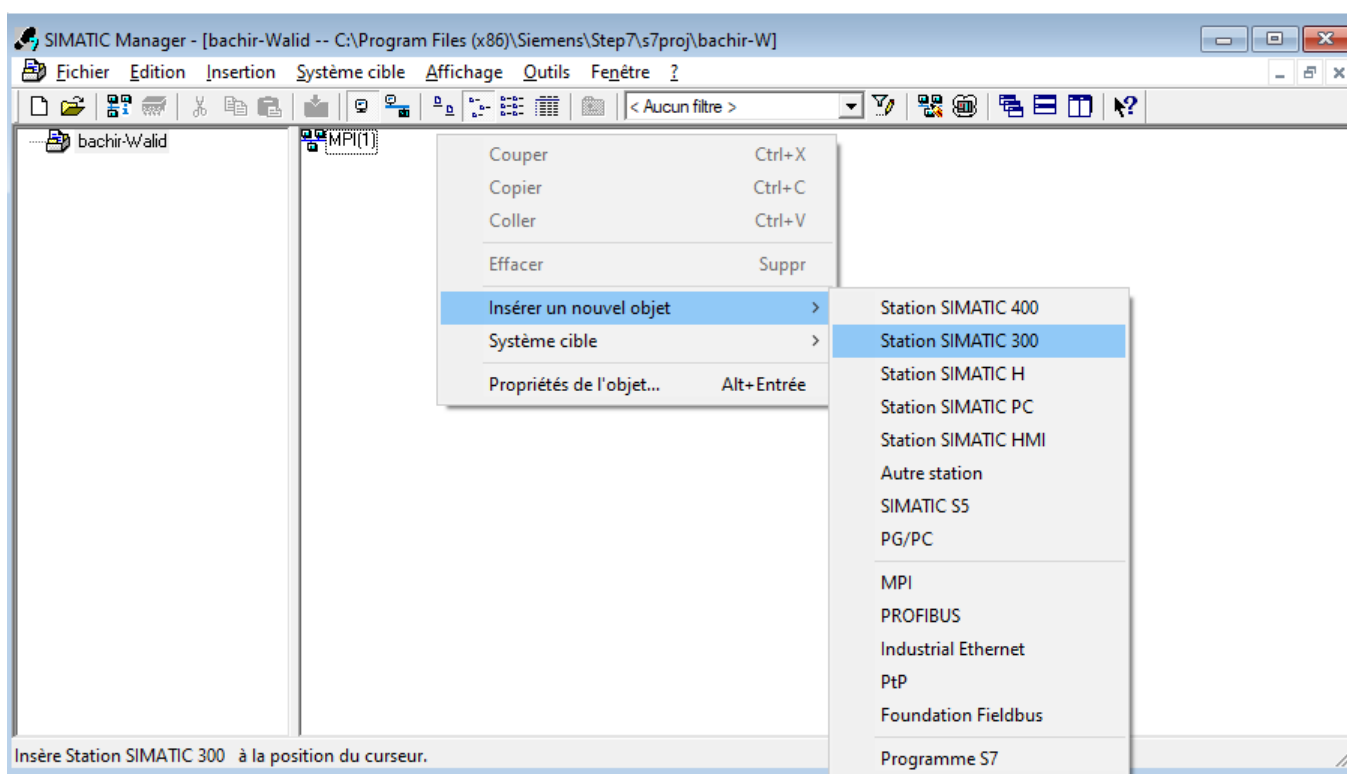


FIGURE 3.3 – Insertion d'une nouvelle station.

- Ouverture la fenêtre de configuration matérielle (HW Config) :

Pour cette étape, la Figure 3.4 montre les trois phases indispensables :

Phase1 : nous appuyons sur « SIMATIC 300 » la station déjà ajoutée.

Phase2 : nous cliquons sur « Matériels ».

Phase3 : La fenêtre de configuration matérielle « le châssis » s'ouvre directement.

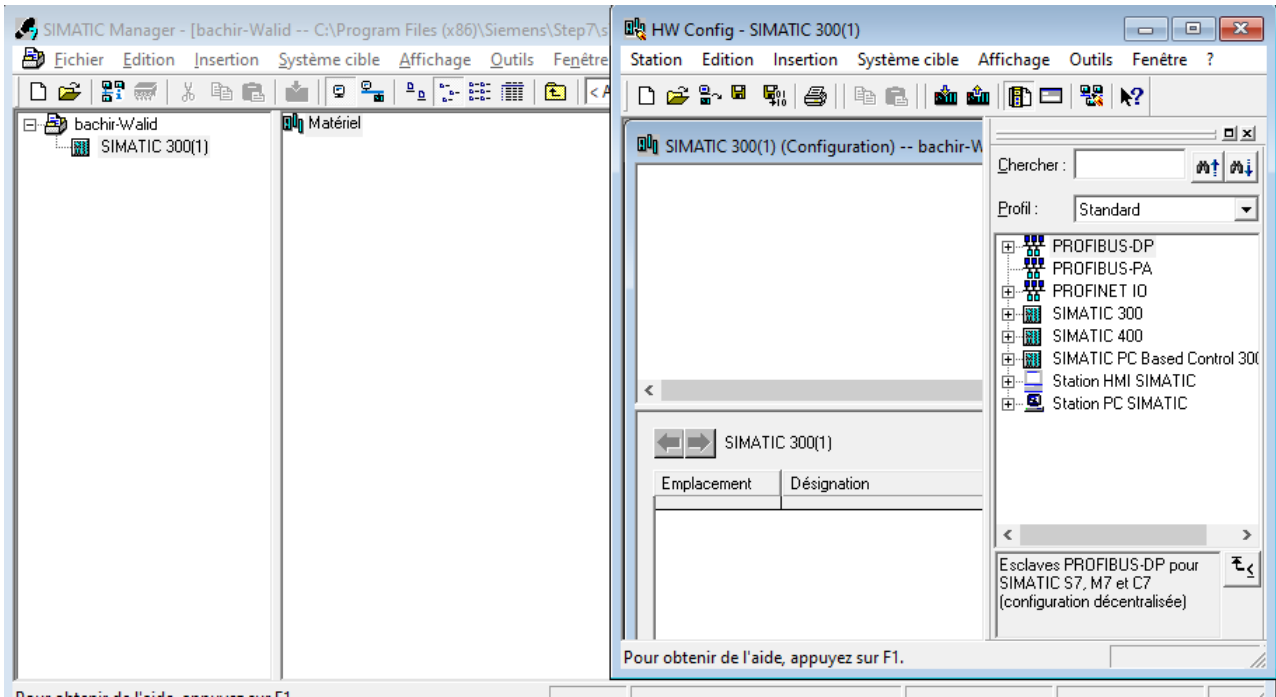


FIGURE 3.4 – Ouverture de la fenêtre de configuration matérielle (HW Config).

- Insertion un Rack :

Pour cette étape, nous cliquons sur « SIMATIC 300 », puis nous choisissons « RACK-300 », nous glissons « profilé support » vers le châssis et enfin nous choisissons le « RACK-300 ». Ces étapes représentent dans La figure ci-dessous :

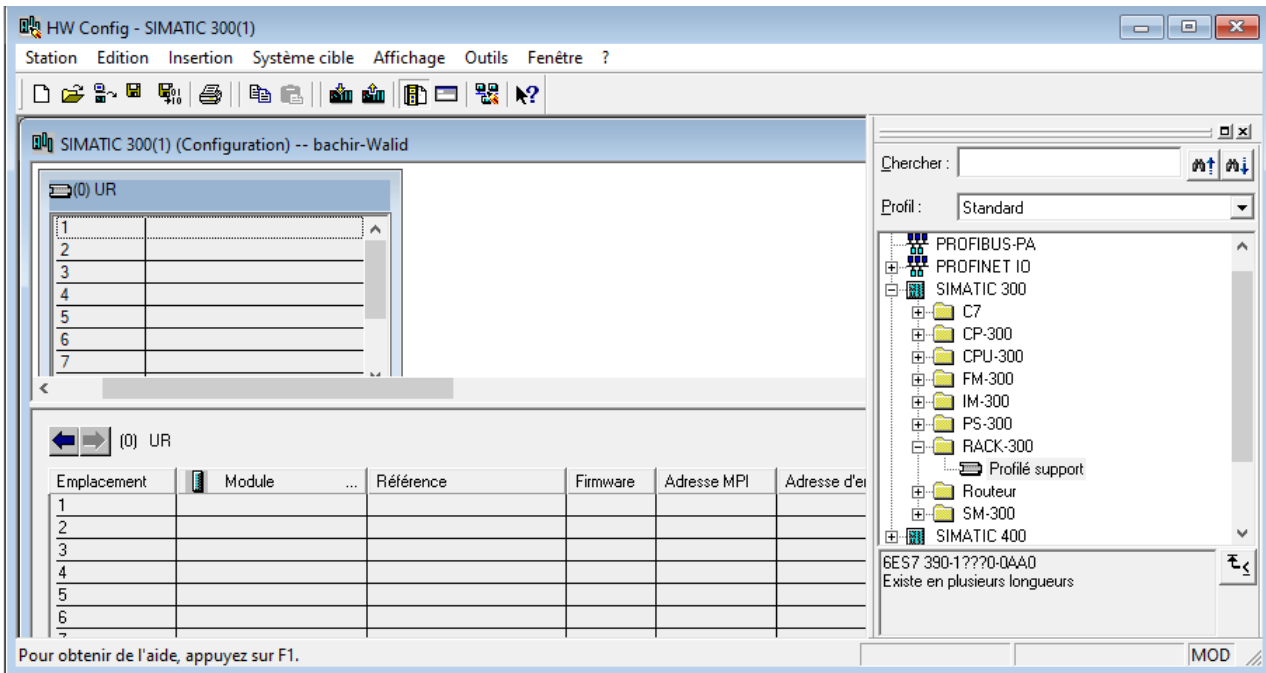


FIGURE 3.5 – Insertion du rack.

- implémentation une alimentation :

Dans cette étape, la Figure 3.6 décrit les Trois phases indispensables à suivre :

Phase1 : nous cliquons sur «PS-300 ».

Phase2 : nous choisissons «PS-307 5A ».

Phase3 : nous glissons «PS-307 5A » dans le « rack ».

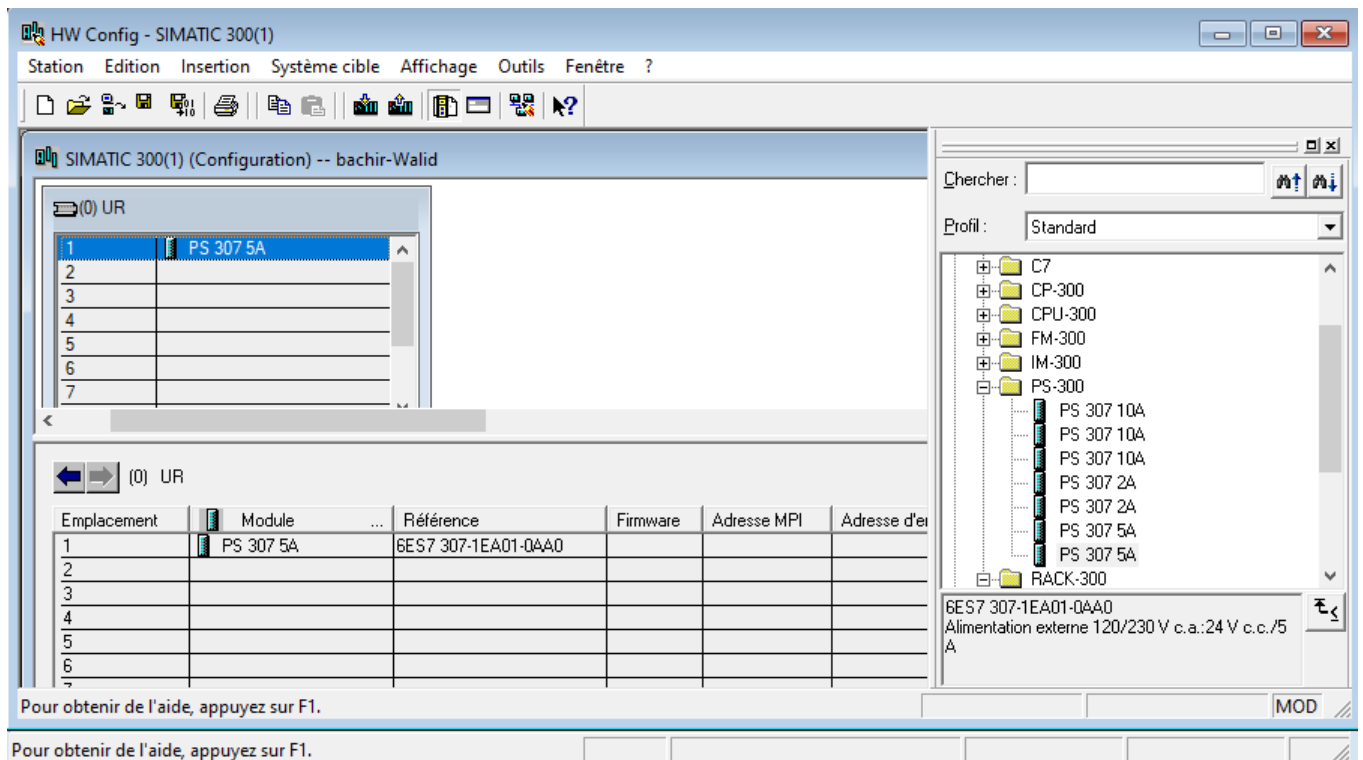


FIGURE 3.6 – Implémentation une alimentation :

- Choix de la CPU 314 PN/DP :

Dans cette étape, nous cliquons sur « CPU-300 », puis nous choisissons le type de CPU de notre travail « CPU 314C-2 PN/DP » et à la fin nous glissons « V3.3 » dans le « rack » (la zone vert). Ces étapes représentent dans La figure ci-dessous :

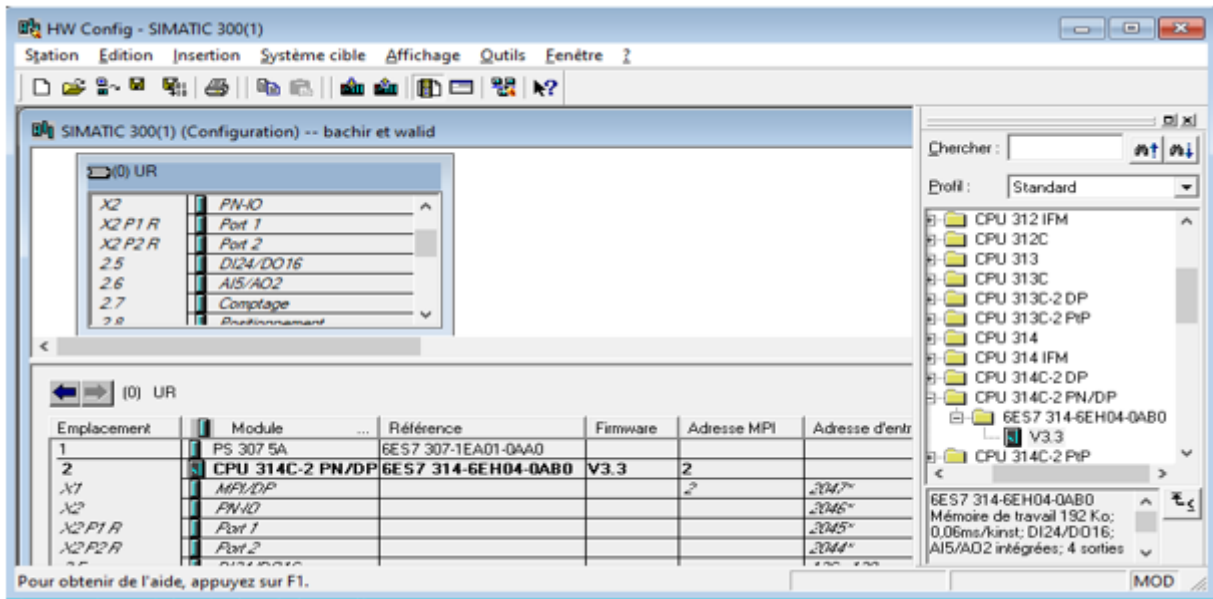


FIGURE 3.7 – Emplacement notre CPU.

- Communication via câble Ethernet :

La communication via le câble Ethernet se faire à base des phases suivantes :

Phase1 : le Clic sur l'icône « nouveau ».

Phase2 : nous cliquons sur « ok » pour assurer le moyen de communication par câblé Ethernet.

Phase3 : En appuyons « ok », donc la CPU sera ajoutée.

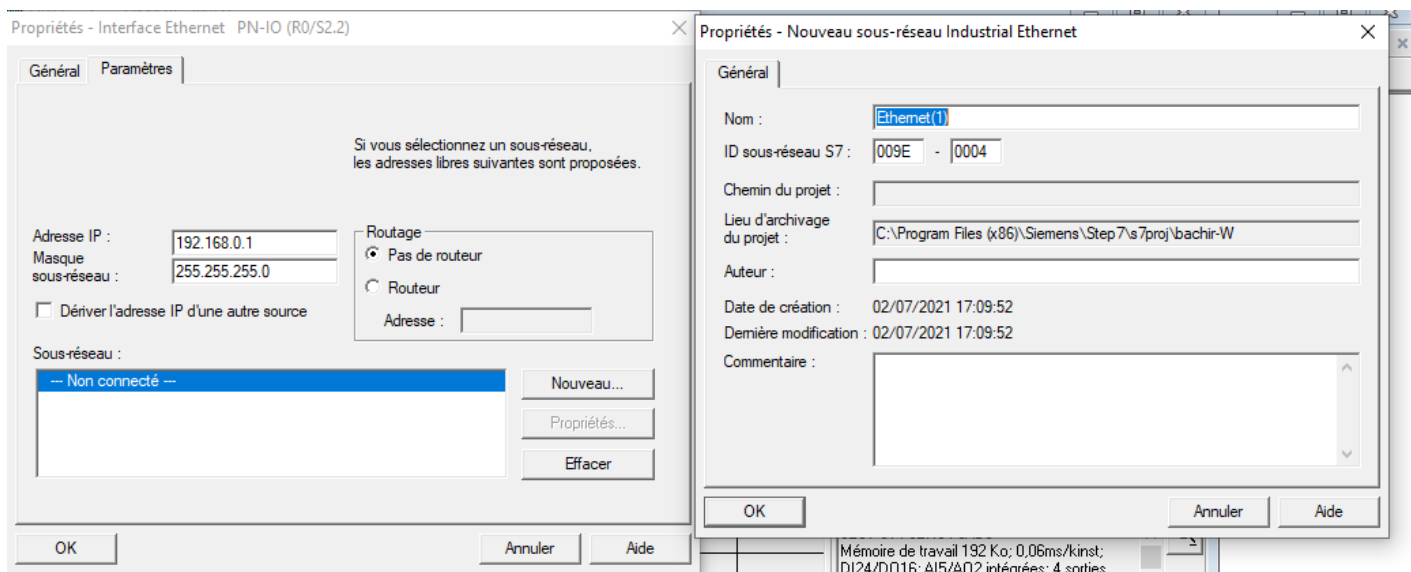


FIGURE 3.8 – Confirmation de communication via câble Ethernet.

- Configuration du réseau :

La configuration de réseau dans l'outil STEP7 est basée sur quelques points importants :

- L'ouvrir « PLCISIM » parce que ce point est très important pour que le logiciel donne la possibilité d'ouvrir la fenêtre de configuration du réseau mère
- Le clic sur « Charger dans le module ».
- En appuyons sur « NETPRO ».
- Nous glissons « connexion au réseau de couleur rouge » dans le « MPI », (une ligne en rouge), cela va ajouter la possibilité de communiquer par câblé MPI.
- Enfin, nous cliquons sur « enregistrer et compiler ».
- La ligne verte représente la communication par le réseau Ethernet qu'on a ajoutée.

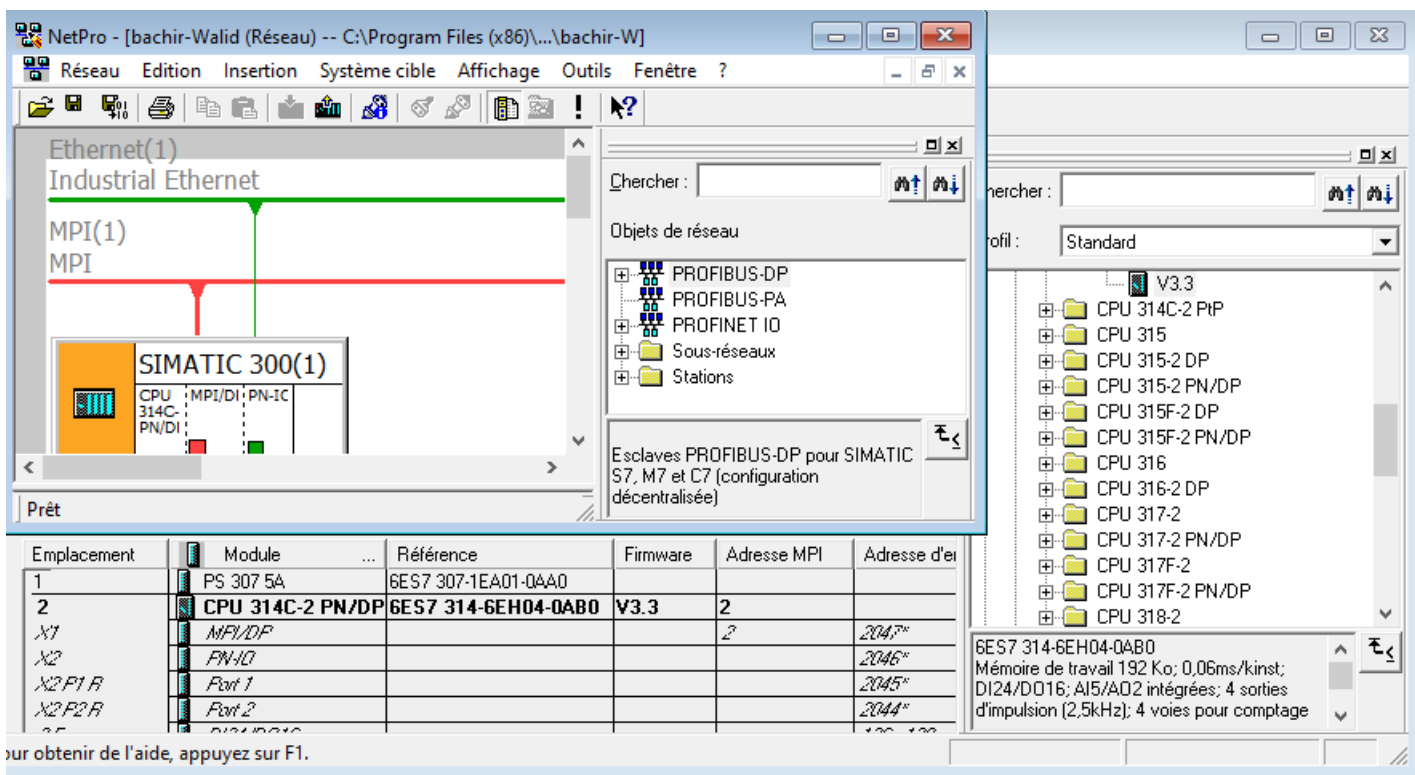


FIGURE 3.9 – Configuration réseau dans STEP 7.

Après ces étapes de la configuration matérielle, nous trouvons un dossier « programme STEP7 » est automatiquement inséré dans le projet, comme indiquer sur la figure ci-dessous :

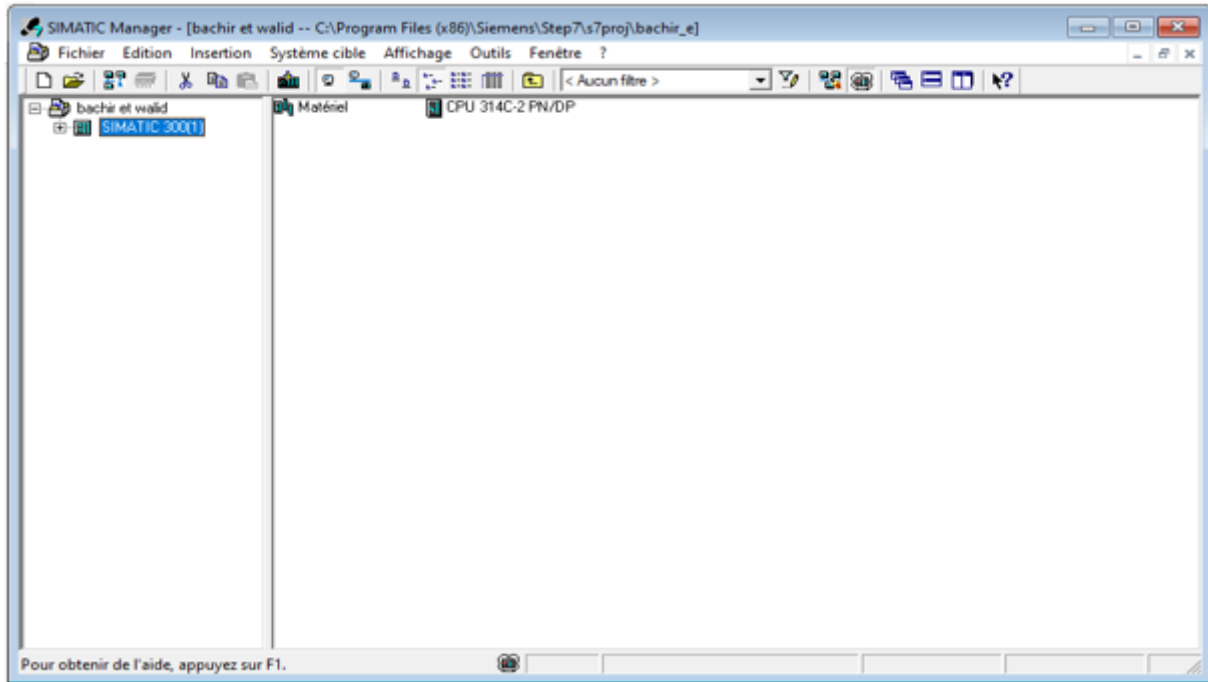


FIGURE 3.10 – Hiérarchie du programme STEP7.

3.5 Table des Mnémoniques

Nous pouvons remplacer les adresses par des mnémoniques locales ou globales avec une désignation plus significative afin de les utiliser dans notre programme. Ils sont destinés pour rendre le programme utilisateur très lisible, pour pouvoir remplacer les données globales par des mnémoniques dans un programme, nous devons les affecter dans la table des mnémoniques. Nous avons utilisé pour cela les adresses absolues qui sont composées d'un identificateur d'opérande et d'une adresse (A 4.0, E 1.1, Z 0).

The screenshot shows the 'Editeur de mnémoniques' window for a SIMATIC S7 program. The table lists 20 variables with their states, mnemonics, operands, data types, and comments.

| | Etat | Mnémonique / | Opérande | Type de do | Commentaire |
|----|------|--------------------|----------|------------|----------------------------------|
| 1 | | 1B1 | E 0.4 | BOOL | éjecteur Buffer 1 en bas |
| 2 | | 1B2 | E 0.5 | BOOL | éjecteur Buffer 1 en haut |
| 3 | | 1M1 | A 0.1 | BOOL | éjecteur vers le Buffer 1 |
| 4 | | 2B1 | E 0.6 | BOOL | éjecteur Buffer 2 en bas |
| 5 | | 2B2 | E 0.7 | BOOL | éjecteur Buffer 2 en haut |
| 6 | | 2M1 | A 0.2 | BOOL | éjecteur vers le Buffer 2 |
| 7 | | 3M1 | A 0.3 | BOOL | Vérin stoppeur |
| 8 | | B2 | E 0.1 | BOOL | capteur Métalique |
| 9 | | B3 | E 0.2 | BOOL | capteur de couleur rouge optique |
| 10 | | Compteur globale | Z 0 | COUNTER | comptage globale |
| 11 | | Compteur Métalique | Z 2 | COUNTER | comptage métalique |
| 12 | | Compteur Noir | Z 3 | COUNTER | comptage Noir |
| 13 | | Compteur Rouge | Z 1 | COUNTER | comptage rouge |
| 14 | | G7_STD_3 | FC 72 | FC 72 | |
| 15 | | K1 | A 0.0 | BOOL | convoyeur |
| 16 | | P1 | A 1.0 | BOOL | lamps start |
| 17 | | P2 | A 1.1 | BOOL | lamps reset |
| 18 | | P3 | A 1.2 | BOOL | glissière pleine |
| 19 | | P4 | A 1.3 | BOOL | |
| 20 | | Part-av | E 0.0 | BOOL | capteur de présence de pièce |

FIGURE 3.11 – Variables utilisées dans notre programme.

3.6 Programme de la station tri

3.6.1 Les blocs de programmation

i. Bloc Fonction (FB1) :

Pour créer un nouveau bloc fonctionnel nous cliquons sur « insertion », après nous appuyons sur « Bloc S7 » et enfin choisissons « 2 Bloc Fonctionnel » avec langage de création « GRAPH », dans la figure ci-dessous nous trouvons ce bloc sera ajouté :

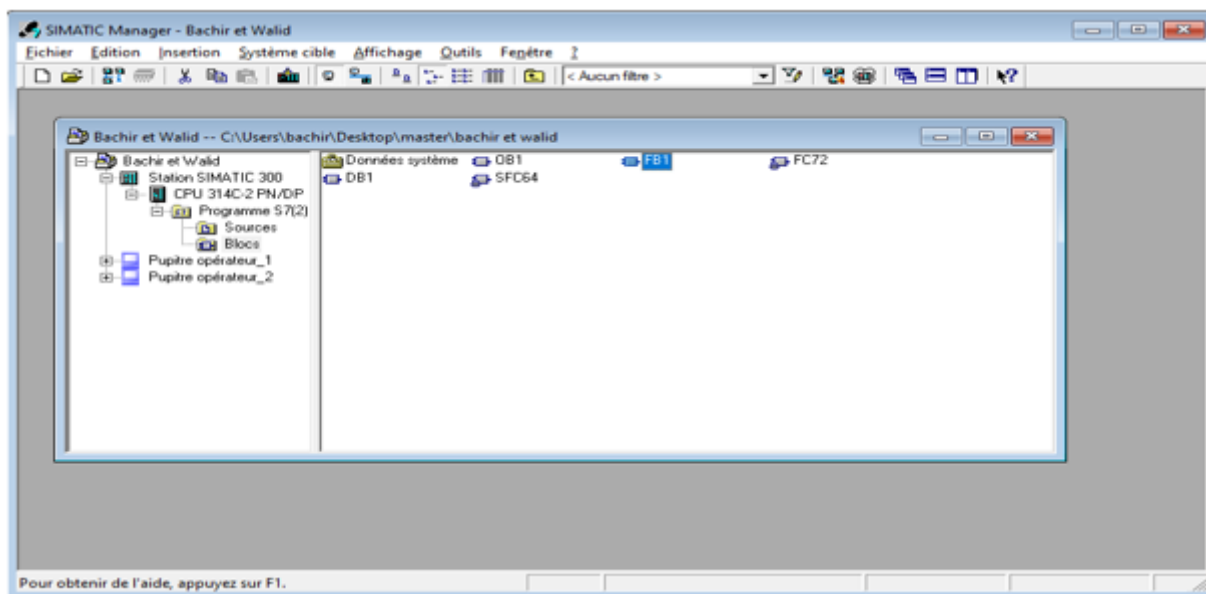


FIGURE 3.12 – Ajouter un bloc FB1.

- Après la création de notre bloc FB1, double cliquer sur ce bloc et nous écrivons notre grafset dans l'espace de commande, comme montre dans les deux figures (3.13 et 3.14) :

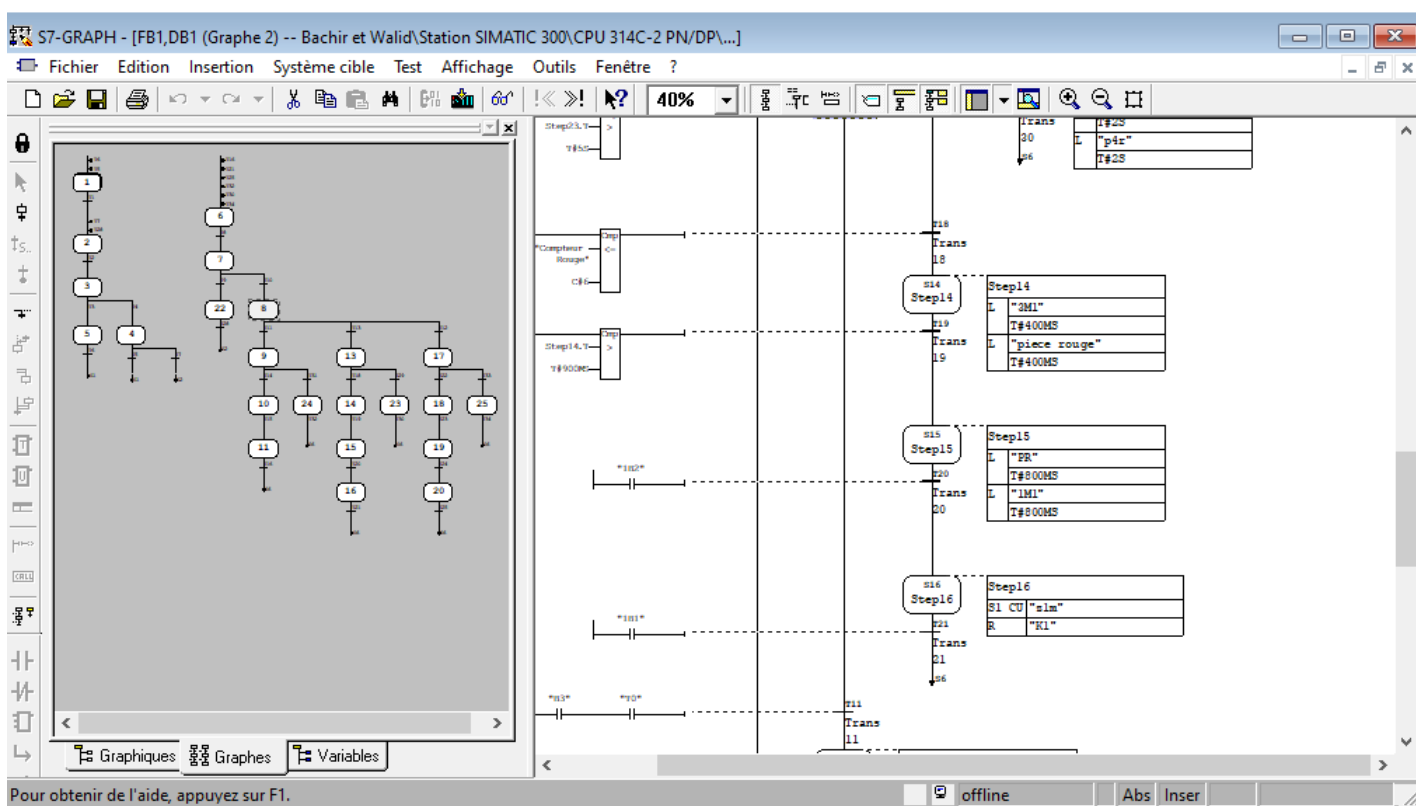


FIGURE 3.13 – S7-GRAPH de la station trie.

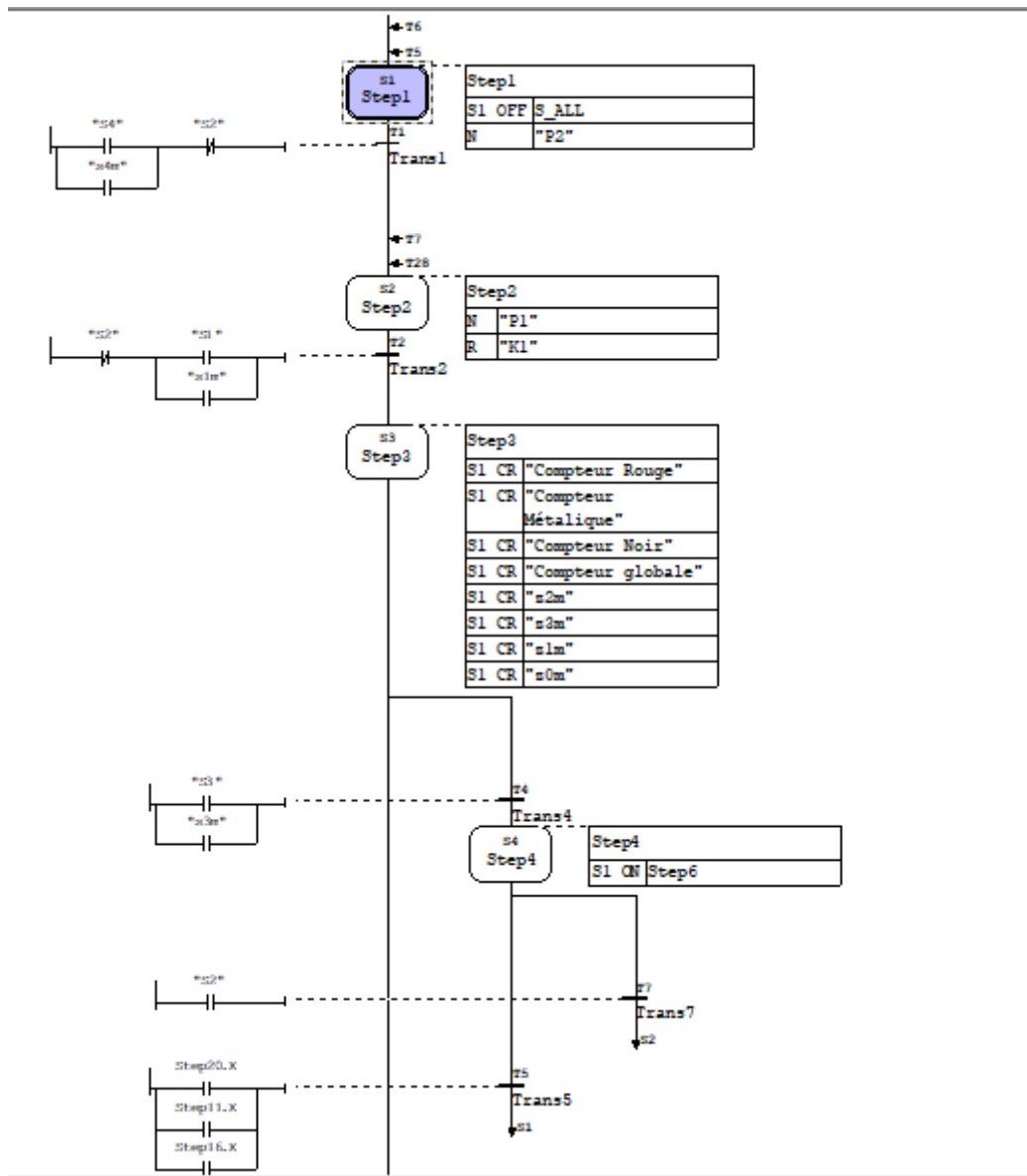


FIGURE 3.14 – Grafset de commande.

ii. Bloc Fonction (OB1) :

Après l'étape précédente, maintenant nous allons faire la liaison entre le bloc FB1 et OB1 via DB1, comme indiqué sur la figure ci-dessous :

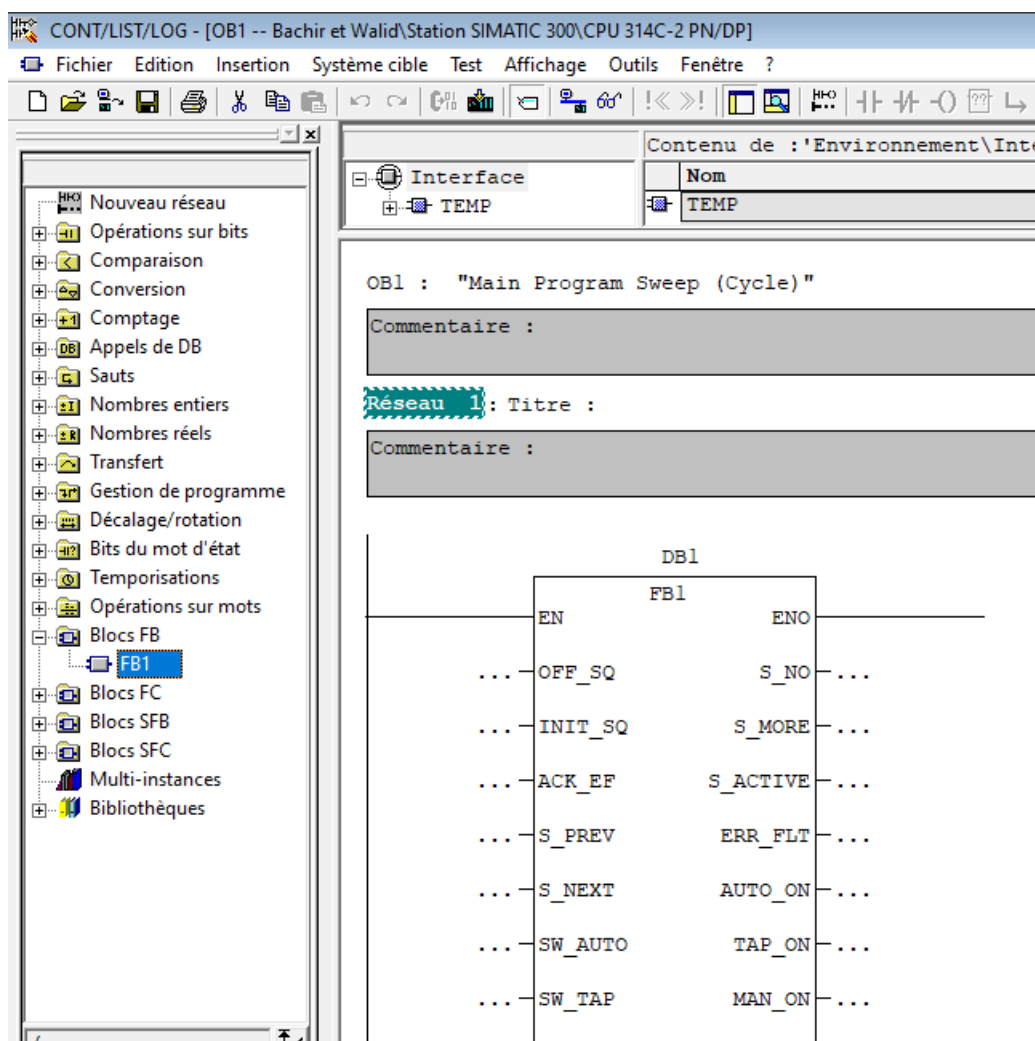


FIGURE 3.15 – Liaison du bloc FB par OB1 via DB1.

3.7 Compilation et simulation du programme

3.7.1 L'EasyPort USB

* Qu'est-ce que l'EasyPort USB :

L'interface de process EasyPort USB sert à la transmission bidirectionnelle de signaux de process entre un process de commande réel en technique très basse tension (24 V=) et un PC. Afin d'exclure les rétroactions du process sur le PC, la transmission des données entre l'EasyPort USB et le PC fait systématiquement appel à des interfaces à séparation galvanique. [17]



FIGURE 3.16 – L'EasyPort USB.

Les possibilités d'utilisation de l'EasyPort USB sont multiples :

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

* Présentation et fonctionnement :[17]

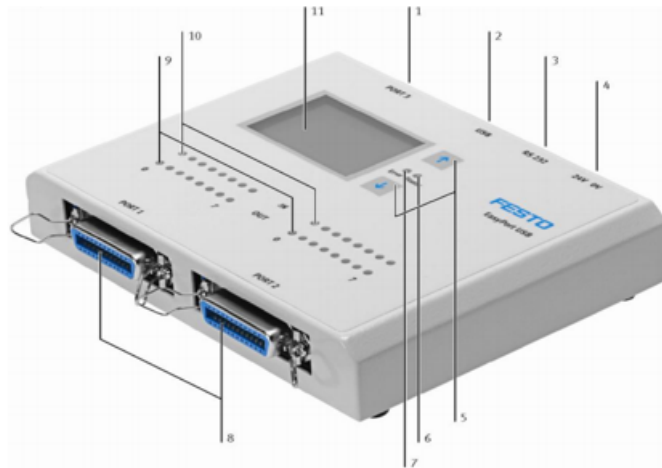


FIGURE 3.17 – Présentation de l'EasyPort USB [17].

- (1) Connecteur Sub D femelle (port 3)
- (2) Interface USB
- (3) Interface RS 232
- (4) Bornes à vis d'alimentation
- (5) Touches
- (6) LED STATUS (verte)
- (7) LED ERROR (rouge)
- (8) Interfaces SysLink (port 1 et port 2)
- (9) LED d'indication d'état des sorties numériques (jaunes)
- (10) LED d'indication d'état des entrées numériques (vertes)
- (11) Écran LCD d'affichage des signaux analogiques et de l'adresse de l'EasyPort.

L'émulateur EasyPort permet de commander notre station (réel) via le PLCSIM, ainsi, en l'absence d'un automate réel, qui est branché directement avec le câble des données des entrées/sorties (Connecteur SysLink) de la station, en modifiant préalablement la configuration sur l'interface EzOPC. [17]

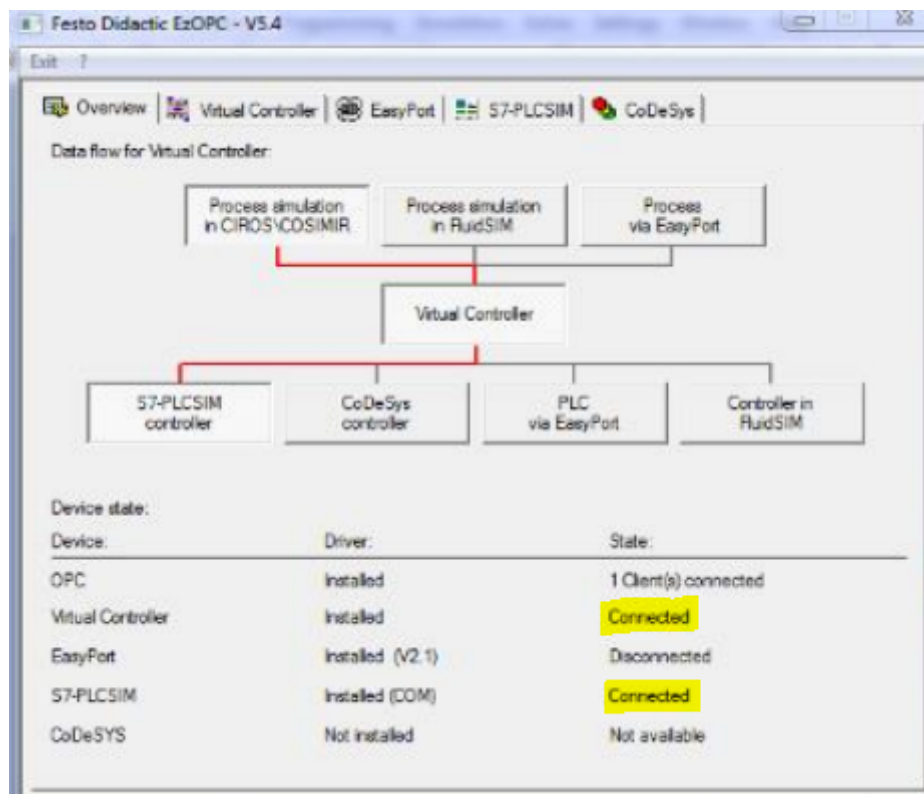


FIGURE 3.18 – Interface EzOPC.

3.7.2 Test du programme dans S7-PLCSIM

Pour exécuter et tester notre programme en l'absence d'un automate réel, il est possible d'utiliser un automate virtuel PLC Sim comme alternative à ce dernier. Il suffit juste d'activer ce simulateur PLC Sim de Logiciel Step7. Tous les accès à l'interface de l'automate sont simulés de manière interne par le logiciel de simulation S7-PLCSIM. C'est un outil très puissant.

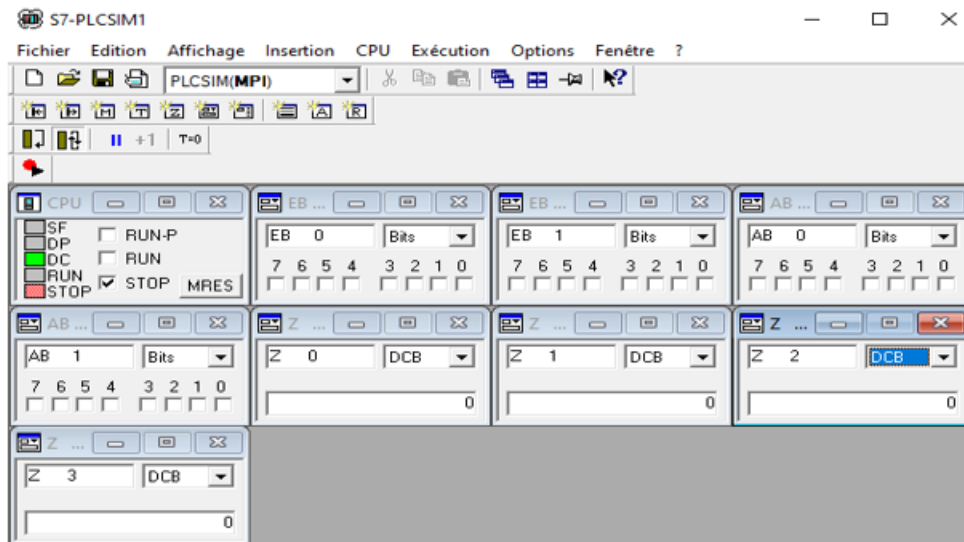


FIGURE 3.19 – S7-PLCSIM.

L'activation de simulation nécessite le clic sur « Activer/désactiver la simulation » puis charger le programme dans la CPU pour l'exécution. Sélectionner « Blocs » et puis cliquer sur « charger ». A la fin, confirmer le message obtenues (voir la figure ci-dessous) :

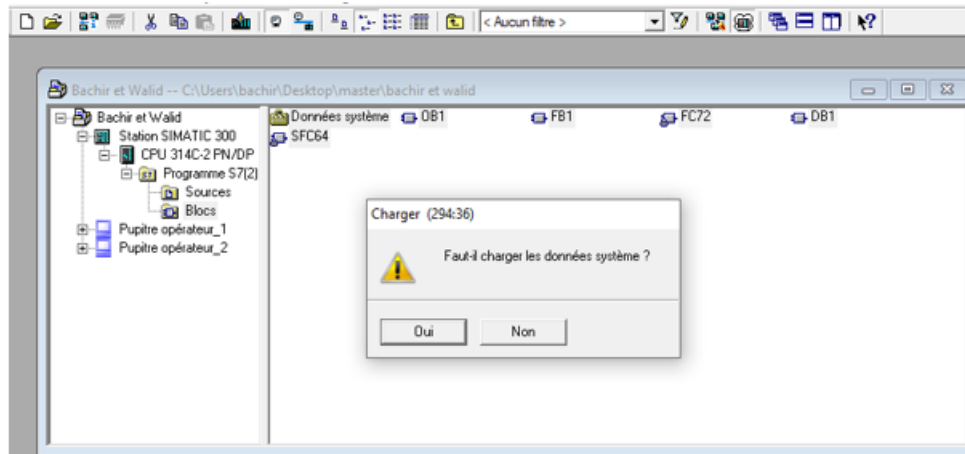


FIGURE 3.20 – Chargement des blocs.

3.7.3 Visualisation

On Lance la simulation dans notre station en suivant les étapes suivantes :

Etap1 : nous cliquons sur « RUN-P » pour démarrer le CPU.

Etap2 : l'ouvrir votre programme et cliquons sur « Monitor » pour visualiser la simulation.

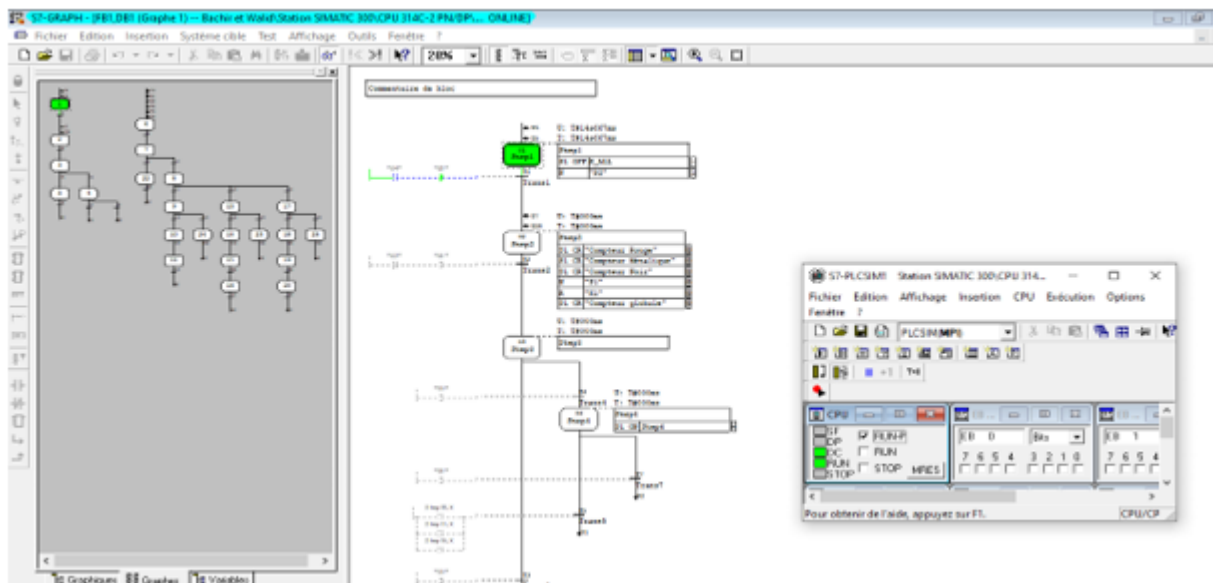


FIGURE 3.21 – Simulation de programme [FB1] (graphe de commande).

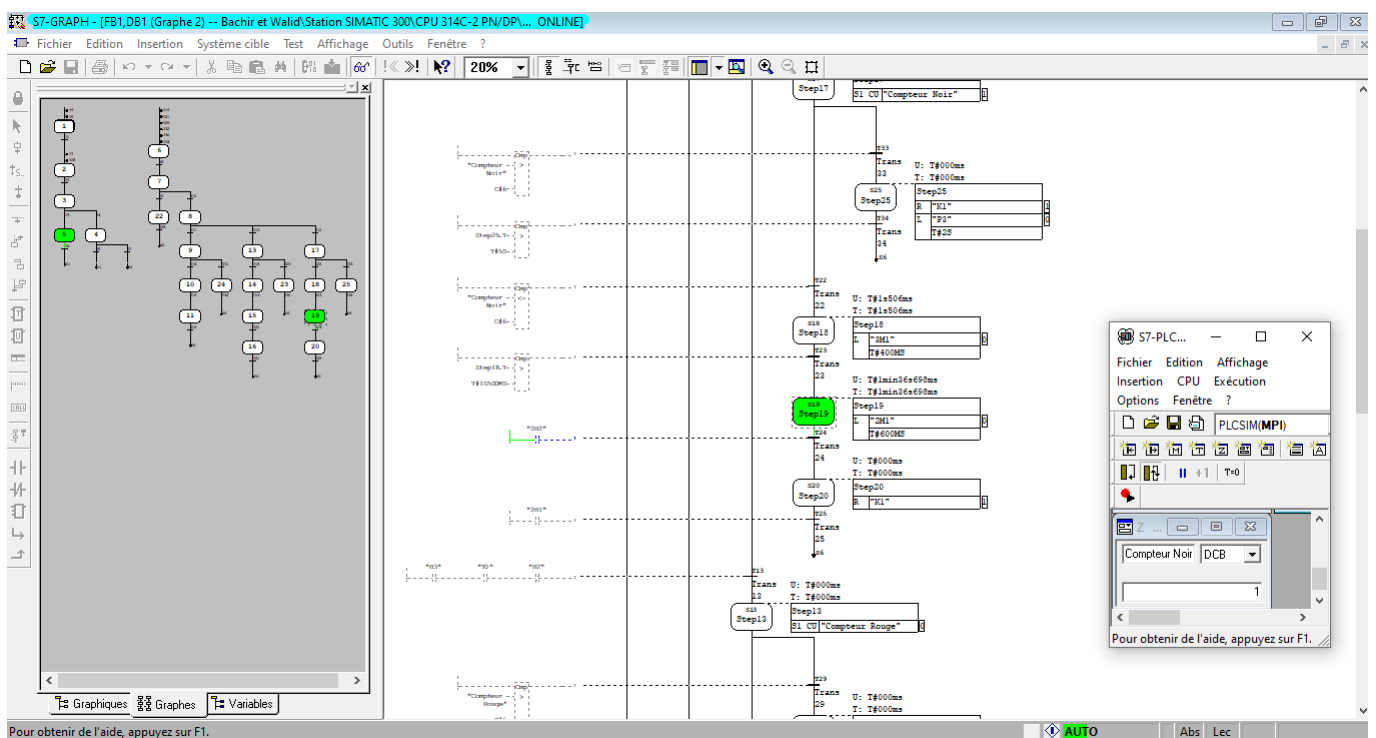


FIGURE 3.22 – Simulation de programme [FB1].

3.8 Conclusion

Dans ce chapitre nous avons présenté le logiciel de programmation SIMATIC STEP7, dont nous avons abordé toutes ses tâches essentielles pour travailler avec notre station.

Ce logiciel est adopté également pour la simulation via le PLC Sim, les programmes et les concepts de commande automatisés avant leur implantation sur le système réel.

CONCLUSION GÉNÉRALE

L'automatisation industrielle a pris en charge le processus de production dans les industries et il est très difficile d'imaginer une ligne de production sans systèmes d'automatisation.

Cette automatisation présente des procédures, des objectifs principaux liée au noyau du système par des automates programmables industriels que nous programmons ou reprogrammons. En premier, l'objectif de notre travail était l'étude et la simulation de la commande, après l'injection en temps réel de la station de tri dans un automate programmable Siemens S7-300 se trouvant au niveau de labo de recherche productique MELT de l'université Abou Bekr Belkaid de Tlemcen, aussi nous avons proposé un cahier de charges fonctionnel suivant l'étude du processus de la station.

Ensuite, nous avons créé notre propre programme en utilisant le logiciel SIMATIC STEP7, afin de choisir le langage graphique « SFC ou bien GRAFCET », enfin en utilisant le logiciel PLCSIM qu'il nous a permis de faire une simulation afin de voir le bon fonctionnement de programme.

La synthèse qu'avons appris durant ce parcours :

- Ces travaux nous ont permis d'approfondir nos connaissances des systèmes automatisés et des automates programmables industriels.
- Nous avons obtenu plusieurs notions sur le système de commande dans le système étudié. Avec l'aide du cahier de charge du système, nous avons pu programmer le fonctionnement de système à partir de nos connaissances.
- Nous avons augmenté nos connaissances en programmation des APIs à l'aide du logiciel SIMATIC Manager, qui inclut l'automate virtuel de la simulation PLCSIM que nous avons utilisé pour simuler le fonctionnement de l'API dans notre application. Nous trouvons que l'objectif principal de notre étude a été atteint.

Comme perspectives nous souhaitons les personnes qui suivent notre travail de faire créer un système MPS variant englobe un ensemble des sous station pour une cible bien précise.

BIBLIOGRAPHIE

- [1] « electronicshub-org-introduction-to-industrial-automation-Introduction » Disponible : <https://www.electronicshub.org/introduction-to-industrial-automation/Introduction> [Consulté le 14/04/2021].
- [2] **M. F. H. Mlle AREZKI Karima** : « Etude simulée de la commande de la bouée SPM par un automate programmable », Université A. MIRA de Bejaia, 2015/2016.
- [3] **K. Amieur** : « Etude et réalisation d'un système de guidage d'une nappe à l'aide d'un automate omron », Université Abderrahmane Mira de Bejaia, 2014.
- [4] **IGUENI ZEDJIGA, GADA SALIHA** : « Etude et Automatisation de la chaîne de production de portes de réfrigérateur par un API S7-300 », mémoire de fin d'études , Université Mouloud Mammeri, Tizi-Ouzou Faculté de Génie Electrique et Informatique Département Automatique, 2007 – 2008.
- [5] <https://www.techno-science.net/definition/3690.html> [Consulté le 14/04/2021].
- [6] **Belhadj Mohammed** : « Capteurs et Instrumentation », UUniversité Tahri Mohammed Béchar, *April*2017.
- [7] **Ronald LII, Claude BERRA** : « capteurs », Enseignements Technologiques Transversaux, version 3/02/2013, Disponible : <http://www.mytopschool.net/mysti2d/activites/polynesie2/ETT/C044/32/Capteurs1/index.html?Les-capteurspotentiometriques.html> [Consulté le 14/04/2021].

- [8] **Mohammed MAATOU, Abderrahman BELLAGH** : « automatiser et réalisation à petite échelle (maquette) d'une chaîne transporteuse de briques », Université Hassiba Benbouali de Chlef, *Juin*2016.
- [9] **G.Boujat, J.P.Pesty** : « AUTOMATISME, REPRODUCTION INTERDITE ASSOCIATION OUVRIERE DES COMPAGNONS DU DEVOIR DU TOUR », DE France, *FÉVRIER*2004.
- [10] **mpeck rousel** : « Cours Complet sur le Grafcet Exercices Corrigés », Centre Universitaire de Technologie Franco-Libanais - CUT Automates et Informatique industrielle.
- [11] « <https://www.festo-didactic.com> » [Consulté le 20/05/2021].
- [12] **ABOUBAKEUR HADJAISSA** : « Automates Programmables Industriels (Description et programmation, Support de cours) », Université Amar Telidji de Laghouat, *13mars*2019.
- [13] « <https://ladderlogicworld.com/plc-architecture/> » [Consulté le 20/05/2021].
- [14] **Redouane Redess** : « LES AUTOMATISMES LA PROGRAMMATION GJC Lycée L.RASCOL 10 », Rue de la République BP 218. 81012 ALBI CEDEX.
- [15] **EL HAMMOUMI** : « Cours Automatismes Logiques Industriels –GE1, école supérieure en technologie », université SIDI MOHAMED BEN ABDELLAH DES FES
- [16] **AG SIEMENS** : « langage CONT pour SIMATIC S7 300/400, programmation des blocs », industrie Siemens, 2010.
- [17] **Festo Didactic GmbH Co. KG** :, 73770 Denkendorf, EasyPort USB, Germany, 2008.

Résumé :

Ce travail est réalisé dans le laboratoire de recherche de l'Université ABOU BEKR BELKAID de Tlemcen. Afin d'automatiser le système étudié, nous avons mené des recherches sur ce sujet. La solution est d'optimiser le remplissage des trois buffers ainsi de réduire la perte d'énergie pendant le fonctionnement du système via un nouveau programme sous langage Grafcet en utilisant l'automate programmable de type Siemens S7-314C 2PN/DP. Le travail est fini par le simulateur en utilisant l'automate virtuel PLC Sim ainsi nous avons analysé le travail et visualiser les résultats en temps réel sur l'automate réel en activant la visualisation.

Mots-Clés : API (PLC), SIMATIC manager, Step7, Grafcet, PLC Sim, TOR, Capteur, Actionnaire, CPU, EzOPC, Easyport USB.

Abstract :

This work is carried out in the research laboratory of the University ABOU BEKR BELKAID of Tlemcen. In order to automate the studied system, we have conducted research on this subject. The solution is to optimize the filling of the three buffers as well as to reduce the loss of energy during the operation of the system via a new program under Grafcet language by using the PLC of type Siemens S7-314C 2PN/DP. The work is finished by the simulator using the virtual PLC Sim so we analyzed the work and visualize the results in real time on the real PLC by activating the visualization.

Key-Words : API, SIMATIC manager, Step7, Grafcet, PLC Sim, PLC, TOR, capteur, Actionnaire, CPU, EzOPC, Easyport USB.

المخلص :

يتم تنفيذ هذا العمل في مخبر أبحاث جامعة أبو بكر بالقائد تلمسان. من أجل أتمتة النظام المدروس، أجرينا بحثاً حول هذا الموضوع.

الحل هو تحسين ملء المخازن المؤقتة الثلاثة وتقليل فقد الطاقة أثناء تشغيل النظام عبر برنامج جديد بلغة Grafcet باستخدام نوع PLC Siemens S7-314C 2PN / DP. تم الانتهاء من العمل بواسطة جهاز المحاكاة باستخدام

PLC Sim الافتراضية لذلك قمنا بتحليل العمل وتصور النتائج في الوقت الفعلي على PLC الحقيقي من خلال تنشيط التصور.

الكلمات المفتاحية : TOR, Grafcet, PLC Sim, Step7, SIMATIC manager, API (PLC)

.Easyport USB, EzOPC, CPU, Actionnaire, Capteur