



الجمهورية الجزائرية الديمقراطية الشعبية
République Algérienne Démocratique et Populaire
وزارة التعليم العالي والبحث العلمي



Ministère de l'Enseignement Supérieur et de la Recherche Scientifique

جامعة غرداية

N° d'enregistrement
/...../...../...../.....

Université de Ghardaïa

كلية العلوم والتكنولوجيا

Faculté des Sciences et de la Technologie

قسم الآلية والإلكتروميكانيك

Département d'automatique et électromécanique

Mémoire

Pour l'obtention du diplôme de Master

Domaine: *Sciences et Technologies*

Filière: *Automatique*

Spécialité: *Automatique et Système*

Thème

**Commande et supervision d'une bande transporteuse à base de
l'automate
Siemens S7-1200.**

Par :

**LAZIZ Abde nour
ABAHAMMOU Abderrahmane**

Devant le jury composé de:

Hamed Boukhari	MCB	Univ. Ghardaïa	Examineur
Khattara Abdelouahab	MCA	Univ. Ghardaïa	Examineur
Mosbah Charaf Abdelkarim	MCB	Univ. Ghardaïa	Encadreur

Année universitaire 2020/2021

Remerciements

Nos remerciements vont tout premièrement à dieu tout puissant pour la volonté, la santé et la patience, qu'il nous a données durant toutes ces longues années.

Nous tenons à remercier notre encadreur Mr : MOSBAH Charaf Abdelkarim, pour son encadrement sans faille, son soutien moral, sa rigueur au travail, ses multiples conseils, ses orientations et sa disponibilité malgré ses multiples occupations.

Nous tenons à adresser chaleureux remerciements à Mr: BOUKHARI Hamed, pour son aide durant toute la période de notre stage et pour le soutien afin de réaliser ce modeste travail.

Je remercie chaleureusement les membres du jury pour l'honneur qu'ils m'ont fait en acceptant d'évaluer mon projet.

Mes remerciements vont aussi à tous les professeurs, enseignants et toutes les personnes qui m'ont soutenu jusqu'au bout, et qui ne cessent de me donner des conseils.

Dédicace

A mes chers parents,

*pour tous leurs sacrifices leur soutien et leurs prières tout au long
de mes études,*

A mes chers frères

pour leur appui et leur encouragement,

A toute ma famille

pour leur soutien tout au long de mon parcours universitaire,

*Que ce travail soit l'accomplissement de vos vœux tant allégués, et le
fruit de votre soutien infaillible,*

Merci d'être toujours là pour moi

Abdenour – Abderahim

اصبحت الصناعات في العصر الحالي تعتمد كليا على الأتمتة ، بدءا من تجهيز المواد الخام الى تسليم المنتج للمستهلك . في هذا المشروع قمنا بدراسة نموذج لخط إنتاج «Industrial control work-cell 290-00 / SI» ذو حزام ناقل ، يتم التحكم فيه باستخدام المبرمج الصناعي PLC S7-1200 لشركة سيمنز . الهدف من المشروع هو التحكم والمراقبة في النموذج من اجل تسهيل عملية الفرز لتقليل الجهد والتكلفة في المصانع .

تم تقسيم العمل الى جزئين : جزء التحكم يتم فيه برمجة ال PLC باستخدام برنامج (STEP7) TIA Portal لفرز القطع على أنماط مختلفة وتخزينها ، وجزء اخر لانشاء واجهة الجهاز البشري HMI ، باستخدام برنامج WinCC .

الكلمات المفتاحية : مبرمج صناعي "سيمنز" ، نموذج خط انتاج ، حزام ناقل ، برنامج "تيا بورتال" ، برنامج "وينسس" ، واجهة الجهاز البشري .

Abstract :

Industries in the current ege are entirely dependent on automation starting from the preparation of raw materials to deliver the product for consumer

In this project, we studied a model of production line «Industrial control work-cell 290-00 / SI» contains a conveyor belt, controlled using a PLC S7-1200 of Siemens. The goal of the project is to control and monitor the model to facilitate the sorting process and reduce effort and cost in factories.

The work divided into two parts: the control part in which the PLC programmed by software TIA Portal (STEP7) to sort the pieces into different patterns and store, another part to create HMI, using software WinCC.

Keywords: programmable logic « SIEMENS PLC », model of production line, conveyor belt, TIA Portal, WinCC, HMI.

Résumé :

Les industries dans l'âge actuel, dépendent entièrement de l'automatisation à partir de la préparation de matières premières pour fournir le produit pour le consommateur.

Dans ce projet qui a été réalisé en laboratoire, nous avons étudié un prototype ligne de production «Industrial control work-cell 290-00 / SI» contient une bande transporteuse, qui est command à l'aide d'un API S7-1200 de Siemens, le but du projet est de command et supervision le prototype afin de faciliter le processus de dépistage, en réduisant les efforts et les coûts dans les usines.

Le travail a été divisé en deux parties : la partie commande est programmé l'automate à l'aide du logiciel TIA Portal (STEP7) pour trier les pièces en différents modèles et les stocker, et une autre partie pour créer l'IHM, en utilisant logiciel WinCC.

Mots clés : Automate "Siemens PLC", prototype ligne de production, bande transporteuse, logiciel "Tia Portal", "WinCC", IHM.

Abréviations

API : Automate Programmable Industriel.

CPU : Unité centrale de l'automate (Central processing unit).

E/S : Entrée / Sortie.

HMI : Interface Homme Machine.

PLC: Automate programmable (Programmable Logic Controller).

SIMATIC: Siemens Automatic.

TIA: Totally Integrated Automation

Win CC: Windows Control Center.

GRAFCET : GRAPhe Fonctionnel de Commande Etape Transition.

EEPROM : Mémoire Morte programmable effaçable électroniquement

TOR : Tout Ou Rien.

Table des matières

Table des matières.....	I
Liste des figures.....	I
Liste des tableaux.....	I
Chapitre I : Automate programmable industriel	
I.1. Introduction:.....	15
I.1.1. Définition et Historique des A.P.I.....	15
I.1.2. Domaines d'emploi des automates :.....	15
I.1.3. Nature des informations traitées par l'automate:.....	15
I.2. Architecture des A.P.I :.....	16
I.2.1. Module d'alimentation :.....	16
I.2.2. L'unité centrale de traitement (CPU) :.....	17
I.2.3. Interfaces d'entrée/sortie :.....	18
I.2.4. Console de programmation :.....	18
I.2.5. Module de communication :	19
I.3. Principe de fonctionnement d'un A.P.I:	20
I.4. Le langage des A.P.I :.....	21
I.4.1. Introduction:.....	21
I.4.2. Les divers types de langages :.....	21
I.4.2.1. Le langage GRAFCET (SFC : Sequential Function Chart) :.....	21
I.4.2.2. Le langage LADDER (LD : Ladder diagram) :.....	22
I.4.2.3. Le langage Booléen (FBD : Function Bloc Diagram) :.....	22
I.4.2.4. Le langage mnémonique (IL : Instruction list) :.....	23
I.4.2.5. Blocs fonction-Blocs opération :	23
I.5. Câblage de l'automate :	24
I.5.1. Alimentation de l'automate :	24
I.5.2. Alimentations des entrées d'automate :	25
I.5.3. Alimentations des sorties d'automate :	26
I.6. Les avantages et les inconvénients d'un automate :	26
I.7. Critères de choix d'un API :	27
I.8. Sécurité :	28
I.9. L'automate S7-1200.....	30
I.9.1 Présentation de automate S7-1200:.....	30

I.9.2. Ingrédients de l'automate S7-1200 :.....	31
I.9.3. Vue de face de la CPU 1214C DC/DC/DC :	32
I.9.4. L'installation d'appareils S7-1200 :	33
I.9.5. Ménagez un dégagement adéquat pour le refroidissement et le câblage:	33
I.9.6. Adresse des signaux d'entrée/sortie de SIMATIC S7-1200 :	34
I.9.7. Conseils de câblage :	35
I.9.7.1. Conditions requises :	36
I.9.8. Connexion à la CPU via le protocole TCP/IP, et retour aux paramètres d'usine:	36
I.9.9. Modules d'extension S7-1200 :	37
I.9.10. Modes de fonctionnement de la CPU :	40
I.9.11. Visualisations d'état et d'erreur :	40
I.9.12. Choix de la CPU :	41
I.9.12.1 Le choix de Platinas d'extension, modules d'E/S, modules de Communication : ...	41
I.10. Conclusion :	42

Chapitre II : Description du système à bande transporteuse

II.1. Introduction :	44
II.2. Description du système:.....	45
II.2.1. Au niveau matériel (hardware) :	45
II.2.1.1. Alimentation Electrique:.....	45
II.2.1.2. Moteurs à courant continu :	46
II.2.1.3. Bande transporteuse:.....	50
II.2.1.4. Les capteurs :	51
II.2.1.5. Source de pression et réservoir de d'air comprimé :	55
II.2.1.6. Les vérins :	56
II.2.1.7. Connexions PLC :	59
II.2.1.8. Mise en marche et arrêt de l'unité API:	60
II.2.1.9. Les modes De Contrôle :	61
II.2.1.10. Zone de stockage :	63
II.2.2. En niveau du logiciel (software) :	64
II.3. Conclusion:	66

Chapitre III: Présentation logiciel TIA Portal & WinCC

III.1. Introduction :	68
III.1.1. Les interfaces de logiciel :	68
III.1.1.1. Interface du portail:	68

III.1.1.2. Interface du projet :	68
III.1.2. Création d'un projet et configuration d'une station de travail :	69
III.1.2.1. Création d'un projet :	69
III.1.3. Configuration et paramétrage du matériel.....	71
III.1.4. Adressage des E/S :	72
III.1.2.1. Adresse Ethernet de la CPU :	72
III.1.5. Les variables API :	73
III.1.5.1. Adresses symbolique et absolue :	73
III.1.5.2. Table des mnémoniques :	74
III.1.6. L'éditeur de programme :	75
III.1.7. Compilation et chargement de la configuration matérielle	76
III.1.8. Simulation :	77
III.1.8.1. Le simulateur PLCSIM :	77
III.2. Présentation du logiciel WinCC :	78
III.2.1. SIMATIC WINCC :	78
III.2.2. Wincc Runtime :	78
III.2.3. Ecran de supervision IHM :	78
III.2.4. Configuration de la connexion IHM avec l'API :	79
III.3. Conclusion:.....	81

Chapitre IV: Commande et supervision du système

IV.1. Introduction :	83
IV.2. Projet 1 : Sélection des pièces selon la hauteur	83
IV.2.1. But de projet :	83
IV.2.2. Cahier des charges :	84
IV.2.3. Modélisation de projet :	84
IV.2.3.1. Table des mnémoniques :	84
IV.2.3.2. Modélisation par l'Algorithme et l'Organigramme :	85
IV.2.3.3. Élaboration de Grafcet :	86
IV.2.3.4. Programmation par TIA Portal :	87
IV.3. Projet 2 : Sélection des pièces selon largeur (diamètre)	89
IV.3.1. But de projet :	89
IV.3.2. Cahier des charges :	89
IV.3.3. Modélisation par l'Algorithme et l'Organigramme :	89
IV.3.4. Élaboration de Grafcet :	92
IV.3.5. Programmation par TIA Portal :	93

IV.4. Projet 3 : Sélection des pièces selon largeur et la hauteur à la fois.....	96
IV.4.1. But de Projet :	96
IV.4.2. Cahier des charges :	96
IV.4.3. Modélisation par l'Algorithme et l'Organigramme :	97
IV.4.4. Elaboration de Grafcet:	98
IV.4.5. Programmation par TIA Portal :	100
IV.5. Projet 4 : Supervision de système sous WinCC.....	104
IV.5.1. Description de projet:.....	104
IV.5.2. Cahier de charge :	104
IV.5.3. Choix de l'Interface Homme-Machine(IHM) :.....	104
IV.5.4. Caractéristiques du SIMATIC IHM KTP900 BASIC PN :.....	105
IV.5.5. Liaison créés par le système lors de l'intégration :.....	106
IV.5.6. Editeur de mnémoniques IHM :.....	107
IV.5.7. Exécution de Runtime :.....	107
IV.6. Conclusion :	109

Liste des figures

Figure I. 1: Architecture materiel d'un API	16
Figure I. 2: Module d'alimentation d'un API	17
Figure I. 3: Module CPU d'un API.....	17
Figure I. 4: Modules d'E/S.....	18
Figure I. 5: Console de programmation	19
Figure I. 6: Modules de communication	19
Figure I. 7: Réseaux de communication Industriel	19
Figure I. 8: Cycle typique d'exécution des programmes d'un A.P.I.....	20
Figure I. 9: Exemples de programmation GRAFCET	22
Figure I. 10: Exemples de programmation LADDER	22
Figure I. 11: Exemple de programmation Booléenne	23
Figure I. 12: Exemple de programmation mnémonique	23
Figure I. 13: Exemple de programmation de blocs fonctions	24
Figure I. 14: Alimentation de l'automate.....	25
Figure I. 15: Alimentations des entrées d'automate.....	26
Figure I. 16: Alimentations des sorties de automate	26
Figure I. 17: automate siemens s7-1200	30
Figure I. 18: La face de la CPU 1214C DC/DC/DC	32
Figure I. 19: Espace pour la connexion Automate Siemens S7-1200.....	33
Figure I. 20: Les entrées et sorties des automates siemens S7-1200	34
Figure I. 21: câblage de l'automate siemens S7-1200.....	35
Figure I. 22: la connexion TCP/IP entre ordinateur et automate siemens s7-1200.....	37
Figure I. 23: Central Processing Unit (CPU)	37
Figure I. 24: Power Module (PM).....	38
Figure I. 25: Signal Boards (SB).....	38
Figure I. 26: Signal Module (SM).....	39
Figure I. 27: Communication Module (CM).....	39
Figure I. 28: Les cartes mémoire.....	40
Figure I. 29: Le mode STOP ou RUN et bouton MRES.....	40
Figure I. 30: le mode de Fonctionnement de la CPU (les LED d'états RUN/STOP).....	41
Figure II. 1: Banc d'essai d'une bande trasporteuse [1].....	44
Figure II. 2: les connexions du prototype « Industrial control work-cell » [1].....	45
Figure II. 3: Commutateur de mode de fonctionnement (manuel/automatique) [1].....	45
Figure II. 4: moteur à courant continu	46
Figure II. 5: principe de fonctionnement d'un moteur à courant continu.....	47
Figure II. 6: Connexion / déconnexion de la commande moteur [1]	50
Figure II. 7: Convoyeurs à bande industriels	50
Figure II. 8: System barrage [6]	53
Figure II. 9: Modification de niveau du capteur [1].....	54
Figure II. 10: Capteur de faisceau optique	54
Figure II. 11: Réglage du capteur Dans le simulateur [1].....	55
Figure II. 12: vérins pneumatiques à simple effet.....	55
Figure II. 13: (a) Pompe à Air et réservoir de pression (b) unité de conditionnement d'air FLR[1].	56
Figure II. 14: Emplois des vérins [5]	57
Figure II. 15: Vue en coupe d'un vérin pneumatique [5].....	57
Figure II. 16: Vérin double effet avec son distributeur [5]	58
Figure II. 17: Vérin simple effet avec son distributeur [5]	58
Figure II. 18: vérins pneumatiques à simple effet (V.S.E.) [1].....	59
Figure II. 19: Connexions PLC [1]	60
Figure II. 20: Interrupteur marche / arrêt de l'unité PLC [1].....	60
Figure II. 21: variateur de vitesse d'une bande transporteuse (VR) [1]	62

Figure II. 22: PLC programs [1]	63
Figure II. 23: les zones des stockages [1]	63
Figure II. 24: Les logiciels des contrôles logiques programmables du prototype «Industrial control work-cell »	65
Figure III. 1: Interface du portail.....	68
Figure III. 2: L'interface du projet	69
Figure III. 3: icône program Tia portal dans l'ordinateur	69
Figure III. 4 : Création d'un projet TIA Portal.....	70
Figure III. 5: Configuration matériel.....	70
Figure III. 6: Ajout d'API	71
Figure III. 7: Vue de l'appareil	71
Figure III. 8: Adressage des entrées/sorties	72
Figure III. 9: Adresse Ethernet de CPU	73
Figure III. 10: Vue l'adresse de variable.....	74
Figure III. 11: L'éditeur des mnémoniques.....	75
Figure III. 12: Vue de l'éditeur de programme	75
Figure III. 13: Vue de compilation et chargement	76
Figure III. 14: La fenêtre de choix de mode et l'interface de communication.....	76
Figure III. 15: Charge de program	77
Figure III. 16: Simulateur PLCSIM	78
Figure III. 17: Ajoute d'IHM	79
Figure III. 18: Vue de l'IHM.....	80
Figure III. 19: Création de la liaison de communication	80
Figure IV-1: Étapes de développement du système.	83
Figure IV-2 : Organigramme du projet 1.	85
Figure IV-3: Grafcet du projet 1	86
Figure IV-4: Réseaux 1 du projet 1.....	87
Figure IV-5 : Réseaux 2 du projet 1	87
Figure IV-6 : Réseaux 3 du projet 1	87
Figure IV-7 : Réseaux 4 du projet 1	88
Figure IV-8 : Réseaux 5 du projet 1	88
Figure IV-9 : Réseaux 6 du projet 1	89
Figure IV-10 : Organigramme de projet 2.	91
Figure IV-11 : Grafcet de sélection de pièces par différente largeur.	92
Figure IV-12 : Réseaux 1 du projet 2.....	93
Figure IV-13 : Réseaux 2 du projet 2	93
Figure IV-14 : Réseaux 3 du projet 2.....	94
Figure IV-15 : Réseaux 4 du projet 2.....	94
Figure IV-16 : Réseaux 5 du projet 2.....	94
Figure IV-17 : Réseaux 6 du projet 2.....	95
Figure IV-18 : Réseaux 7 du projet 2.....	95
Figure IV-19 : Réseaux 8 du projet 2.....	95
Figure IV-20 : Réseaux 9 du projet 2.....	96
Figure IV-21 : Organigramme de projet 3.	98
Figure IV-22 : Grafcet de sélection de pièces par différente hauteur et largeur.	99
Figure IV-23 : Réseaux1 du projet 3.....	100
Figure IV-24 : Réseaux 2 du projet 3.....	100
Figure IV-25 : Réseaux 3 du projet 3.....	101
Figure IV-26 : Réseaux 4 du projet 3.....	101
Figure IV-27 : Réseaux 5 du projet 3.....	102
Figure IV-28 : Réseaux 6 du projet 3.....	102
Figure IV-29 : Réseaux 7 du projet 3.....	102

Figure IV-30 : Réseaux 8 du projet 3	103
Figure IV-31 : Réseaux 9 du projet 3	103
Figure IV-32 : Réseaux 10 du projet 3	103
Figure IV-33 : Ajout d'IHM.	105
Figure IV-34 : L'interface de supervision KTP900 BASIC PN.	106
Figure IV-35 : Les paramètres de liaison d'une IHM.	106
Figure IV-36 : Editeur de mnémoniques.....	107
Figure IV-37 : Vue les compteurs des pièces.....	107

Liste des tableaux

Tableau 1 : Adresse des signaux d'entrée/sortie	36
Tableau 2 : Différentes façons de se connecter les bobinages du l'inducteur et de l'induit	48
Tableau 3 : Commutateurs de commande Manuels	61
Tableau 4 : Table des mnémoniques	84



Introduction
Générale

Introduction générale

Avec le développement de la technologie dans l'âge actuel, les usines sont entièrement dépendantes de l'automatisation dans toutes les opérations en commençant par la préparation de matières premières afin de fournir le produit pour le consommateur.

Pour atteindre cet objectif, nous avons étudié un modèle de ligne de production «Industrial control work-cell 290-00 / SI» contient une bande transporteuse dans l'objectif de faire le tri des pièces de différent modèle.

Les objectifs fixés dans notre travail sont regroupés en quatre chapitres comme suit :

- Le premier chapitre consiste à la présentation des automates programmables industriels de Siemens.
 - Le deuxième chapitre est consacré à la description et l'instrumentation de système automatisé à bande transporteuse.
 - Le troisième chapitre consiste à la présentation de logiciel de programmation et simulation TIA Portal et logiciel de supervision WinCC.
 - Dans le quatrième chapitre on présente notre application selon quatre projets qui sont:
la programmation et simulation de trois projet sur logiciel TIA Portal :
 - Projet1 pour la sélection des pièces selon la hauteur.
 - Projet2 pour la sélection des pièces selon largeur (diamètre).
 - Projet3 pour la sélection des pièces selon largeur et la hauteur à la fois.
- Enfin,
- Projet4 pour simulation de la supervision par logiciel WinCC.

Une conclusion générale marque la fin de notre mémoire.



Chapitre I :
***Automate programmable
industriel***

I.1. Introduction:

On utilise les API dans tous les secteurs industriels donc Dans ce chapitre, on va présenter les automates programmables et Architecture et Fonctionnement API ainsi que les langages de programmation, puis nous terminerons par la description du traitement de l'information dans l'API.

I.1.1. Définition et Historique des A.P.I

L'Automate Programmable industriels (API) est un dispositif électronique programmables adapté aux environnements industriels qui exécutent des fonctions automatisées pour contrôler les pré-actionneurs et les actionneurs basés sur des informations logiques, analogiques ou numériques. Les API sont apparues à la fin des années soixante aux États-Unis vers 1969. Ils étaient principalement utilisés pour automatiser les chaînes de montage d'automobiles à la demande de l'industrie automobile américaine (GM), qui réclamait plus d'adaptabilité de leurs systèmes de commande.

I.1.2. Domaines d'emploi des automates :

L'API convient à tous les domaines industriels, il permet de contrôler les machines (transport, emballage, etc.) ou les chaînes de production (automobile, agro-alimentaire ... etc), et peut également assurer des fonctions de régulation de processus (métallurgie, chimie ... etc). Il est de plus en plus utilisé dans le secteur de la construction (secteurs tertiaire et industriel) pour contrôler le chauffage, l'éclairage, la sécurité ou les alarme [1].

I.1.3. Nature des informations traitées par l'automate:

Les informations peuvent être de type :

- Tout ou rien (T.O.R.) : l'information ne peut prendre que deux états (vrai/faux, 0 ou 1 ...). C'est le type D'information délivrée par un détecteur, un bouton poussoir ...
- Analogique : l'information est continue et peut prendre une valeur comprise dans une plage bien Déterminée. C'est le type d'information délivrée par un capteur (pression, température ...)

- Numérique : l'information est contenue dans des mots codés sous forme binaire ou bien hexadécimale. C'est le type d'information délivrée par un ordinateur ou un module intelligent (COURS : Les Automates Programmables Industriels) [1].

I.2. Architecture des A.P.I :

Un API comprend généralement des modules arrangés l'un à côté de l'autre, tels que l'alimentation, l'unité centrale (CPU) à base de microprocesseur dotée d'une carte de Mémoire, les interfaces d'entrées et de sorties, les interfaces de communication, les cartes spéciales et un dispositif de programmation (voir Figure I. 1). On peut effectivement considérer qu'il s'agit d'une unité contenant un grand nombre de relais, compteurs, temporisateurs et unités de stockage de données distincts (généralement EEPROM) [2].

La figure montre la disposition de base d'un API :

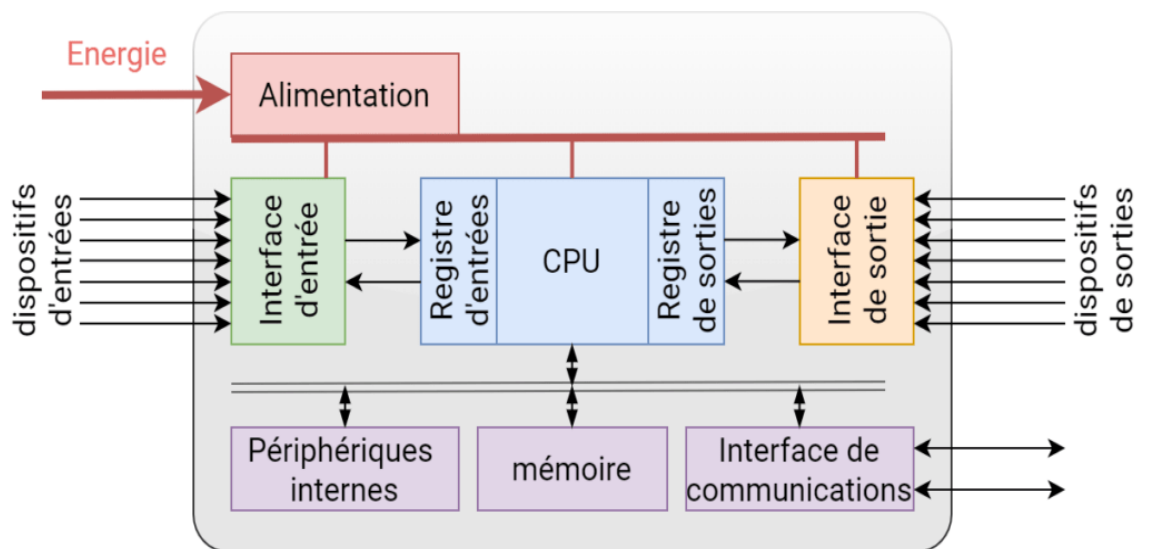


Figure I. 1: Architecture matériel d'un API

I.2.1. Module d'alimentation :

Le bloc d'alimentation (power supply), (voir Figure I. 2), est nécessaire pour convertir La tension d'entrée alternative (220 V) du secteur en une tension continue (24V, 48V...) Nécessaire au processeur et aux circuits des modules d'interface d'entrée et de sortie [2].

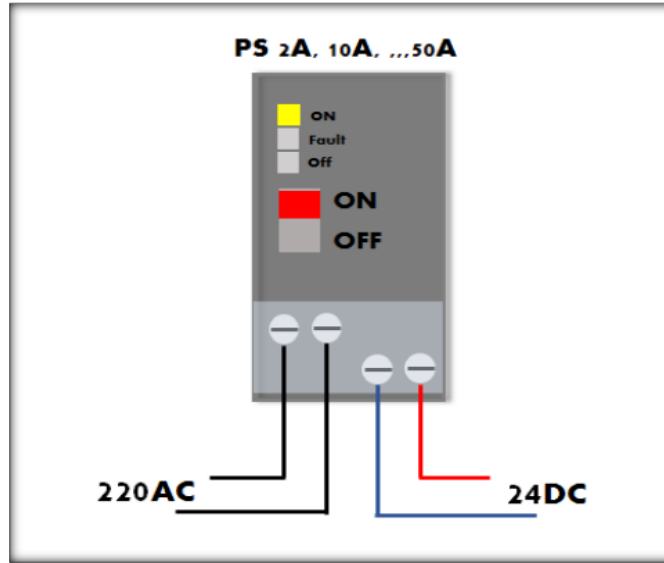


Figure I. 2: Module d'alimentation d'un API

La puissance des alimentations varie entre un API et un autre et demandent un courant allant de 2A à 50A, en fonction du nombre d'interfaces d'E/S alimentées par cette alimentation [2].

I.2.2. L'unité centrale de traitement (CPU) :

Le module CPU est l'unité contenant le microprocesseur (voir Figure I. 3). Cette unité interprète les signaux D'entrée et exécute les actions de commande en fonction du programme enregistré dans sa Mémoire, communiquant les décisions sous forme des signaux d'actions aux sorties, Aussi, Ce module contient Un interface de programmation afin de communiquer avec la console De programmation suivant un protocole bien déterminé (Par exemple TCP/IP, MPI-bus ...etc.) [2].

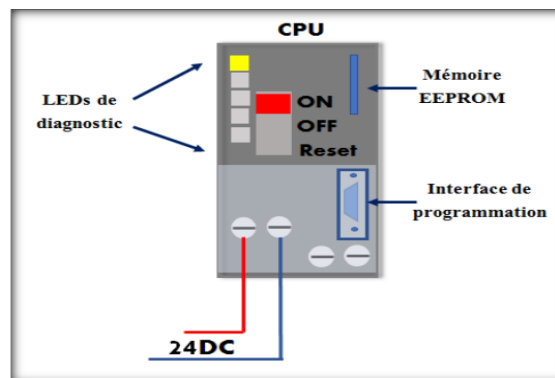


Figure I. 3: Module CPU d'un API

La mémoire de programme est l'endroit où le programme stocké contenant les actions De contrôle à exécuter par le microprocesseur, généralement c'est une mémoire ROM Effaçable électriquement (EEPROM) d'une capacité varie du 4KB jusqu'au 50KB [2].

I.2.3. Interfaces d'entrée/sortie :

Les cartes d'E/S permettent au processeur de recevoir des informations de périphériques Externes (capteurs) et de Les communiqués aux périphériques externes (Pré-actionneurs Et actionneurs) (voir Figure I. 4), généralement il y a deux types d'E/S, type Tout ou Rien (DI/DO) et Analogique (AI/AO) [2].

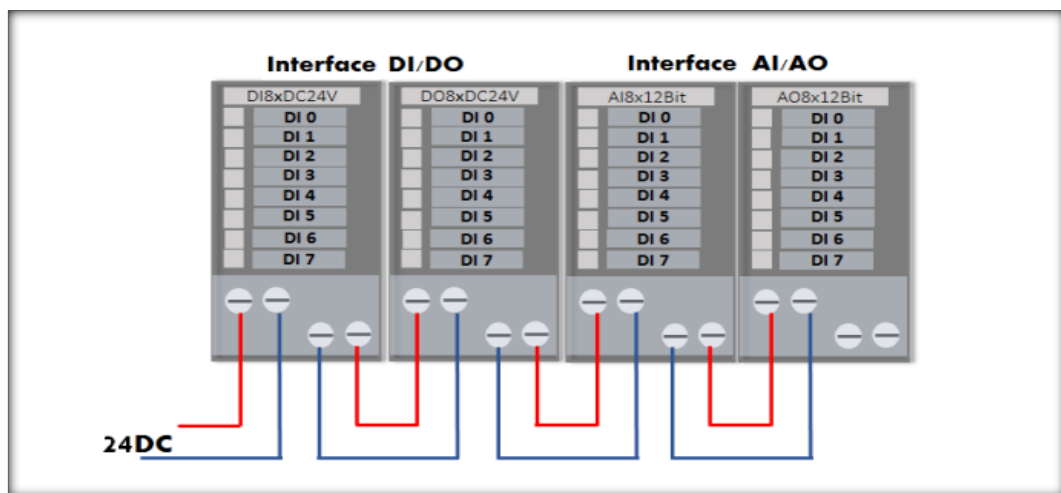


Figure I. 4: Modules d'E/S

Plus de ces modules, on trouve des modules spéciaux d'E/S (carte PID, carte de Comptage rapide ...etc.), ce type des cartes dotés des microprocesseurs, afin de simplifier Les tâches et soulager le module CPU [2].

I.2.4. Console de programmation :

Le dispositif de programmation est utilisé pour introduire le programme souhaité dans La mémoire programmable (voir Figure I. 5). Généralement le programme est développé dans un PC ou Une console spéciale donnée par le constructeur, puis transféré dans la mémoire du CPU Par l'intermédiaire d'un câble de communication adéquat (MPI-bus, TCP/IP...Etc.), voir L'exemple de la figure [2].

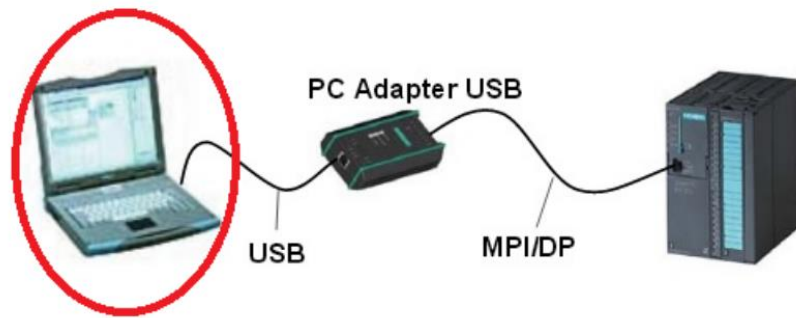


Figure I. 5: Console de programmation

I.2.5. Module de communication :

L'interface de communication est utilisée pour recevoir et transmettre des données sur des réseaux de communication depuis ou vers d'autres systèmes distants Tels qu'API, SCADA F&G, HMI, serveur OPC ...etc. (voir Figure I. 6). Il concerne des actions telles que la vérification du périphérique, l'acquisition de données, la synchronisation entre les systèmes et la gestion de la connexion [2].

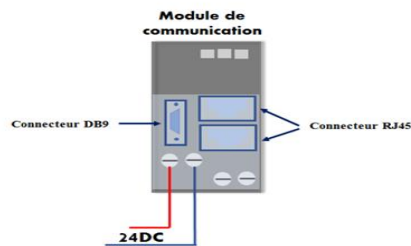


Figure I. 6: Modules de communication

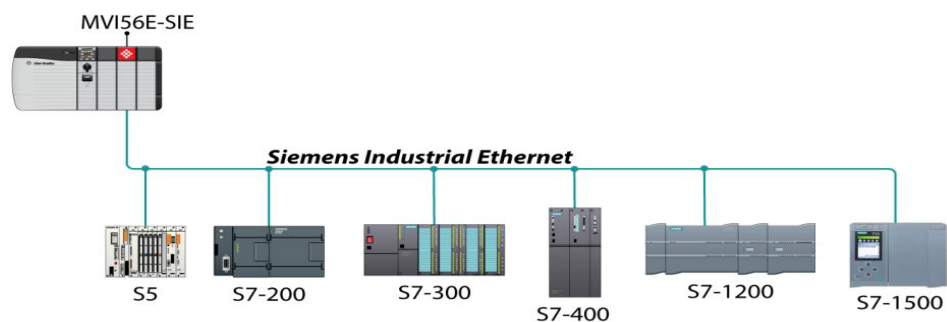


Figure I. 7: Réseaux de communication Industriel

I.3. Principe de fonctionnement d'un A.P.I:

Dans un ordinateur l'exécution d'un programme se fait en général ligne par ligne et d'une façon asynchrone. Une des caractéristiques de l'automate est de fonctionner différemment c.à.d. de façon cyclique. En effet avant d'exécuter quoi que ce soit, l'automate lit entièrement son programme ; et une fois l'exécution terminée recommence les mêmes opérations [3].

On définit alors la notion de cycle et de temps de cycle (entre 1ms et 30ms environ). Il existe plusieurs types de cycle mais le plus répandu est le celui représenté sur la Figure I .8.

Ce cycle comprend 5 phases :

- **Phase 1** : Lecture ou Acquisition des entrées: Prise en compte des informations des modules D'entrées et écriture de leur valeur dans RAM (zone DONNEE).
- **Phase 2** : Exécution Des programmes ou Traitement des données : Lecture du programme (située Dans la RAM programme) par l'unité de traitement, lecture des variables (RAM données), Traitement et écriture des variables (internes, sorties ...) dans la RAM données.
- **Phase 3** : Traitement de toute demande de communication.
- **Phase 4** : Exécution du test d'autodiagnostic (Gestion du système Autocontrôle de l'automate).
- **Phase 5** : Ecriture des sorties : Lecture des variables de sorties dans la RAM données et transfert vers le module de sorties.

Le temps de scrutation de chaque cycle est vérifié par un temporisateur appelé **Watchdog** (chien de Garde) qui enclenche une procédure d'alarme en cas de dépassement de celui-ci (réglé par l'utilisateur).

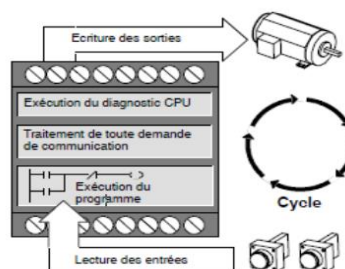


Figure I. 8: Cycle typique d'exécution des programmes d'un A.P.I.

I.4. Le langage des A.P.I :

I.4.1. Introduction:

Les langages des A.P.I. sont des langages intermédiaires entre le langage évolué et le langage machine. Ils ont l'avantage d'avoir un jeu d'instructions incluant uniquement les fonctions logiques, cela a comme conséquences, une meilleure compréhension par les automaticiens et une simplification du compilateur de la console de programmation et du logiciel constructeur [3].

I.4.2. Les divers types de langages :

Malheureusement il n'y a pas eu d'unicité chez les constructeurs quant au langage de programmation [3].

Néanmoins 4 langages sont parmi les plus utilisés (norme CEI 61131-3) :

- Le langage LADDER (LD : Ladder diagram).
- Le langage booléen (FBD : Function Bloc Diagram).
- Le langage GRAFCET (SFC : Sequential Function Chart).
- Le langage mnémonique (IL : Instruction list).

Chaque automate se programmant via une console de programmation propriétaire ou par un ordinateur équipé du logiciel constructeur spécifique [1].

I.4.2.1. Le langage GRAFCET (SFC : Sequential Function Chart) :

Le GRAFCET, langage de spécification, est utilisé par certains constructeurs d'automate (Schneider, Siemens) pour la programmation (voir Figure I. 9). Parfois associé à un langage de programmation, il permet une programmation aisée des systèmes séquentiels tout en facilitant la mise au point des programmes ainsi que le dépannage des systèmes. On peut également traduire un grafcet en langage en contacts et l'implanter sur tout type d'automate.

Certains logiciels permettent une programmation totale en langage GRAFCET et permettent de s'adapter à la plupart des automates existants (logiciels CADEPA ou AUTOMGEN) [1].

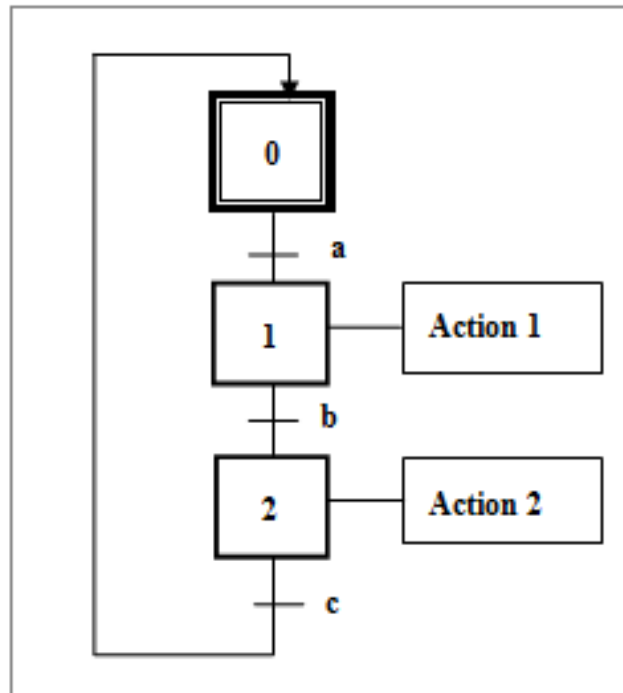


Figure I. 9: Exemples de programmation GRAFCET

I.4.2.2. Le langage LADDER (LD : Ladder diagram) :

Langage graphique développé pour les électriciens. Il utilise les symboles tels que : Contacts, relais et blocs fonctionnels et s'organise en réseaux (labels). C'est le plus utilisé (voir Figure I. 10) [1].

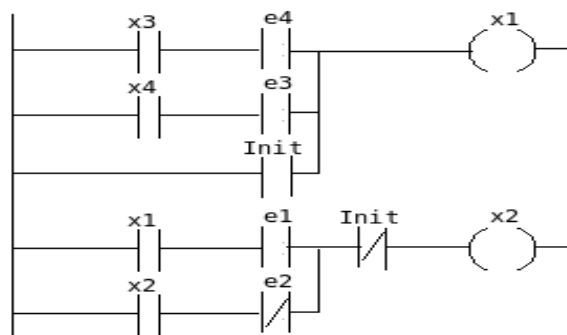


Figure I. 10: Exemples de programmation LADDER

I.4.2.3. Le langage Booléen (FBD : Function Bloc Diagram) :

Langage graphique où des fonctions sont représentées par des rectangles avec les entrées à gauche et les sorties à droites. Les blocs sont programmés (bibliothèque) ou programmables. Utilisé par les automaticiens (voir Figure I. 11) [1].

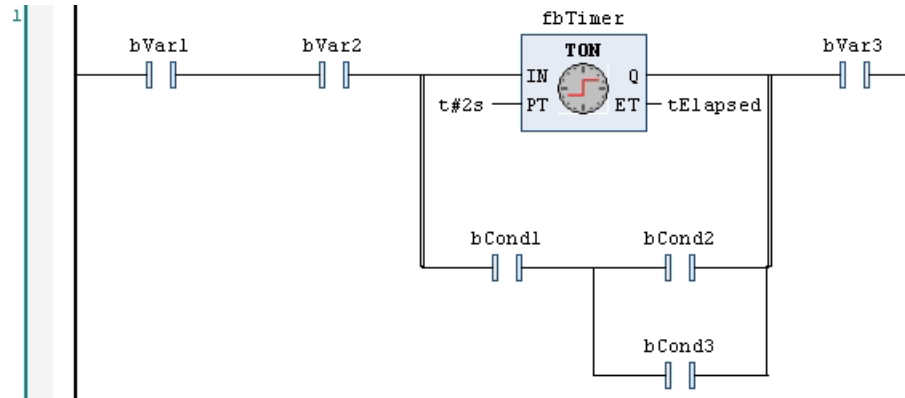


Figure I. 11: Exemple de programmation Booléenne

I.4.2.4. Le langage mnémotechnique (IL : Instruction list) :

Langage textuel de même nature que l'assembleur (programmation des microcontrôleurs) (voir Figure I. 12).

Très peu utilisé par les automaticiens [1].

```

! %L0 : LD      %I1.0
        ANDN   %M12
        OR [   %TM4.Q
        AND   %M17
        ]
        AND   %I1.7
        ST   %Q2.5
! %L5 : LD      %I1.10
        ANDN  %Q2.3
        ANDN  %M27
        IN   %TM0
        LD   %TM0.Q
        AND  %M25
        AND  %M00.XS
        [ %M01S := %M01S+500]
    
```

Figure I. 12: Exemple de programmation mnémotechnique

I.4.2.5. Blocs fonction-Blocs opération :

En complément aux possibilités de représentation, les langages de programmation permettent l'utilisation de blocs fonctions et de blocs opérations. Ces blocs sont des fonctions préprogrammées, paramétrables, utilisables directement par le programmeur (voir Figure I. 13).

Exemple de blocs fonctions : Temporisateur, Compteur [3].

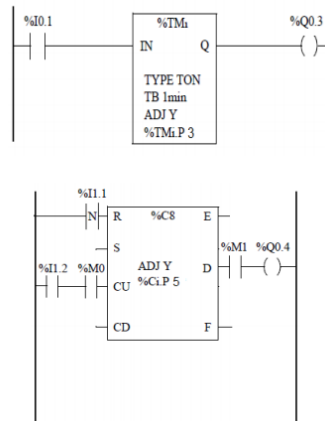


Figure I. 13: Exemple de programmation de blocs fonctions

I.5. Câblage de l'automate :

I.5.1. Alimentation de l'automate :

L'automate est alimenté généralement par le réseau monophasé 230V; 50 Hz, mais d'autres alimentations sont possibles (110 V etc...). La protection est de type magnétothermique (voir Figure I. 14). Il est souhaitable d'asservir l'alimentation de l'automate par un circuit de commande spécifique (contacteur KM1). De même, les sorties seront asservies au circuit de commande et alimentées après validation du chien de garde [4].

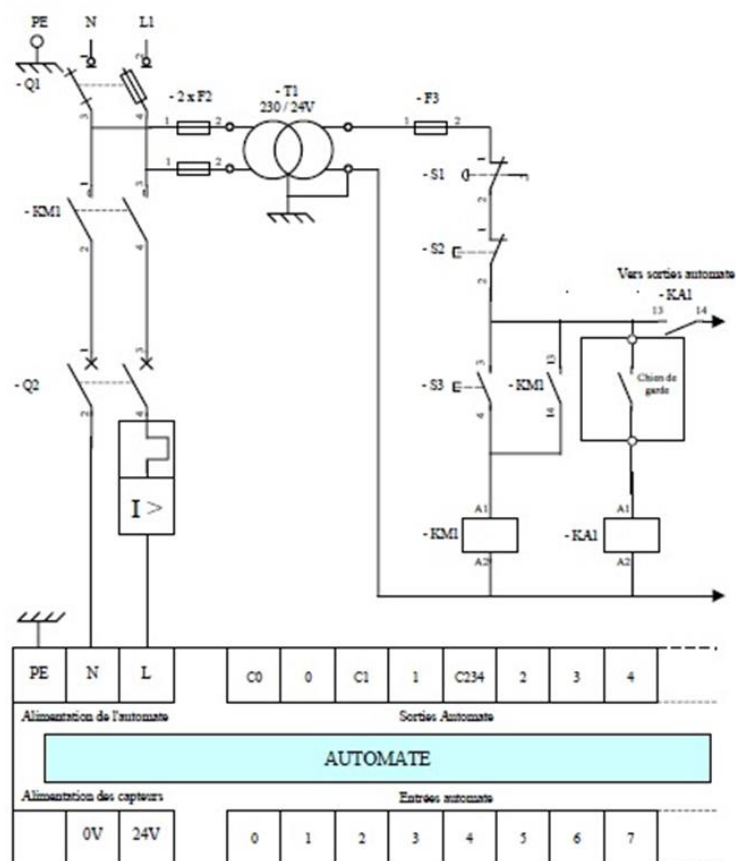


Figure I. 14: Alimentation de l'automate

I.5.2. Alimentations des entrées d'automate :

L'automate est pourvu généralement d'une alimentation pour les capteurs/détecteurs (attention au type de logique utilisée : logique positive ou négative) (voir Figure I. 15). Les entrées sont connectées à l'0V (commun) de cette alimentation. Les informations des capteurs/détecteur sont traitées par les interfaces d'entrées [4].

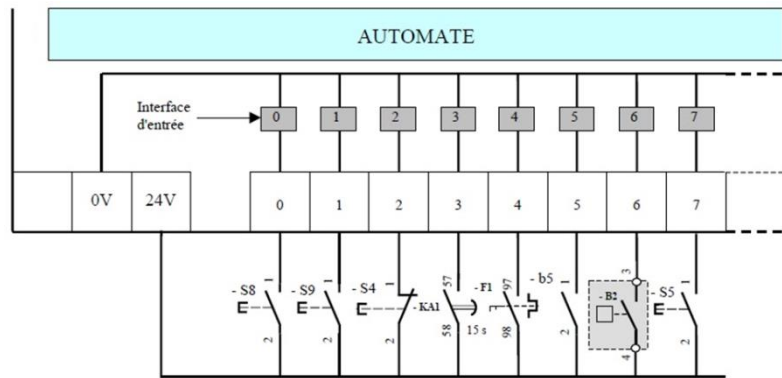


Figure I. 15: Alimentations des entrées d'automate

I.5.3. Alimentations des sorties d'automate :

Les interfaces de sorties permettent d'alimenter les divers pré-actionneurs, Il est souhaitable d'équiper chaque pré-actionneur à base de relais de circuits RC (non représentés) (voir Figure I. 16) [4].

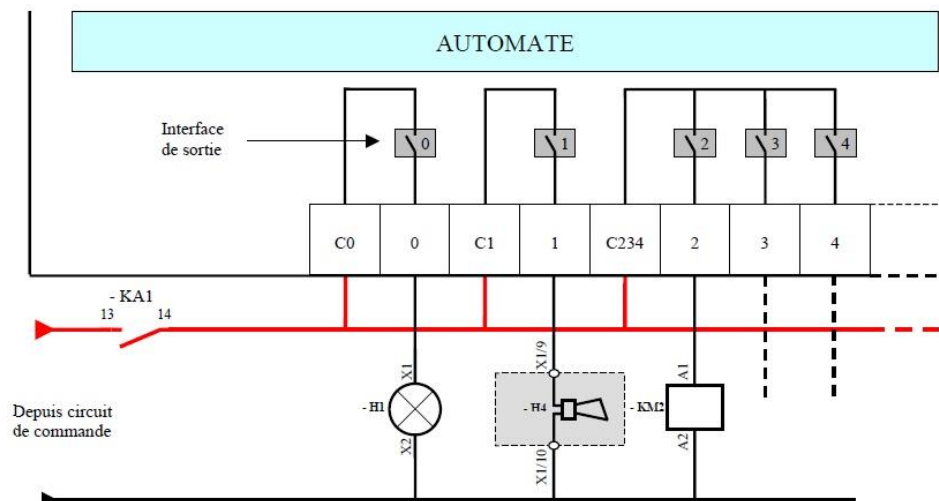


Figure I. 16: Alimentations des sorties de automate

I.6. Les avantages et les inconvénients d'un automate :

Les avantages sont :

- Améliorer les conditions de travail en éliminant les travaux répétitifs.
- Améliorer la productivité en augmentant la production.
- Améliorant la qualité des produits ou en réduisant les coûts de production.

- Automates programmables sont programmés facilement et ont un langage de programmation facile à comprendre (logique programmé) alors la Modification du programme facile par rapport à la logique câblée [4].
- Simplification du câblage.
- Puissance et rapidité.
- Facilité de maintenance (l'API par lui-même est relativement fiable et peut aider l'homme dans sa recherche de défauts).
- Augmenter la sécurité.
- Possibilités de communication avec l'extérieur (ordinateur, autre API)
- énorme possibilité d'exploitation.
- plus économique.

Les inconvénients sont :

- Plantage.
- Il y a trop de travail requis dans les fils de connexion.
- Besoin de formation.

I.7. Critères de choix d'un API :

Plusieurs critères existent pour le choix d'un API, citant par exemple :

- Le rapport Qualité/Prix.
- La simplicité de programmation qui offre un langage destiné à l'automaticien suivant la norme IEC 61131.
- Possibilités de simulation et de visualisation qui apportent à l'utilisateur une aide efficace à la mise au point et à l'exploitation, par exemple S7-PLCSIM de SIEMENS.
- La puissance de traitement et un ensemble des cartes spécialisées permettant un développement aisé d'applications particulières : communication, asservissement d'axes, régulation...etc.
- Possibilités d'extension en termes d'entrées et de sorties.
- Standardisation des protocoles de communication [2].

I.8. Sécurité :

Les systèmes automatisés sont, par nature, source de nombreux dangers (tensions utilisées, déplacements mécaniques, jets de matière sous pression ...) [3].

Placé au cœur du système automatisé, l'automate doit être un élément fiable car :

- un dysfonctionnement pourrait provoquer de graves répercussions sur la sécurité des personnes.
- les coûts de réparation de l'outil de production sont généralement très élevés.
- un arrêt de la production peut avoir de lourdes conséquences sur le plan financier.

Aussi, l'automate fait l'objet de nombreuses dispositions pour assurer la sécurité :

- Contraintes extérieures : l'automate est conçu pour supporter les différentes contraintes du Monde industriel et a fait l'objet de nombreux tests normalisés (tenue aux vibrations, CEM ...)
- Coupures d'alimentation : l'automate est conçu pour supporter les coupures d'alimentation et permet, par programme, d'assurer un fonctionnement correct lors de la réalimentation (reprises à froid ou à chaud)
- Mode RUN/STOP : seul un technicien peut mettre en marche ou arrêter un automate et la remise en marche se fait par une procédure d'initialisation (programmée)
- Contrôles cycliques : Procédures d'autocontrôle des mémoires, de l'horloge, de la batterie, de la tension d'alimentation et des entrées / sorties
- Vérification du temps de scrutation à chaque cycle appelée Watchdog (chien de garde), et enclenchement d'une procédure d'alarme en cas de dépassement de celui-ci (réglé par L'utilisateur)
- Visualisation : Les automates offrent un écran de visualisation où l'on peut voir l'évolution des entrées / sorties

La défaillance d'un automate programmable provoque de graves répercussions en matière de sécurité, les normes interdisent la gestion des arrêts d'urgence par l'automate ; celle-ci doit être réalisée en technologie câblée.

Chapitre I : Automate programmable industriel

On peut également ajouter des modules de sécurité à l'automate (sécurité des machines). Il existe enfin des automates dits de sécurité (API dS) qui intègrent des fonctions de surveillance et de redondance accrues et garantissent la sécurité des matériels.

I.9. L'automate S7-1200

I.9.1 Présentation de automate S7-1200:

L'automate SIMATIC S7-1200 est un mini-contrôleur modulaire utilisé pour les petites performances (voir Figure I. 17). Il existe un éventail complet de modules pour une adaptation optimisée à la tâche d'automatisation. Le contrôleur S7 est composé d'une CPU qui est équipée d'entrées et de sorties de signaux numériques et analogiques.

Des modules additionnels d'entrées/sorties (modules IO) peuvent être installés si les entrées et sorties intégrées ne sont pas suffisantes pour l'application désirée. Si besoin sont des modules de communication RS232 ou RS485 sont ajoutés. Une interface TCP/IP intégrée est obligatoire pour toutes les CPU.

Avec le programme S7, l'API surveille et contrôle une machine ou un processus. Les modules IO sont interrogés dans le programme S7 au moyen d'adresses d'entrées (%I) et référencés au moyen d'adresses de sorties (%Q) [5].

Le système est programmé avec le logiciel TIA Portal (STEP 7).



Figure I. 17: automate siemens s7-1200



I.9.2. Ingrédients de l'automate S7-1200 :

Il comprend :

- un contrôleur avec interface PROFINET intégrée pour la communication avec une console de programmation, une interface homme-machine ou d'autres contrôleurs SIMATIC.
- des fonctions technologiques performantes intégrées, par exemple : comptage, mesure, régulation et motion control - entrées/sorties TOR et analogiques intégrées,
- des signaux Boards utilisables directement sur un contrôleur,
- des modules d'entrées/sorties pour l'extension des contrôleurs par des canaux d'entrées/sorties
- des modules de communication pour l'extension des contrôleurs par des interfaces de communication,
- des accessoires, par ex. alimentation, modules de commutation ou SIMATIC Memory Card. On peut ainsi lui citer les qualités suivantes :
- un effet d'automatisation maximal pour un coût raisonnable,
- simplicité de montage, de programmation et de manipulation,
- hautement intégré, peu encombrant,
- convenant pour les applications d'automatisation de petite à moyenne envergure,
- convenant pour les applications fermées jusqu'ici à la logique programmée pour des raisons de coûts,
- toutes les CPU utilisables en mode autonome, en réseau et dans des architectures décentralisées [6].

Pour la réalisation de ce projet, nous avons utilisé l'automate SIMATIC S7-1200

- Unité centrale 1412C qui a les caractéristiques techniques suivantes :
- 14 entrées et sorties intégrées
- Interface Ethernet intégrée.
- Contrôleur PID de base
- Horloge en temps réel intégrée
- Entrées d'alarme.

Bornes détachables sur tous les modules

I.9.3. Vue de face de la CPU 1214C DC/DC/DC :

Avec une alimentation intégrée de 24V et des entrées et sorties TOR intégrées, la CPU 1214C DC/DC/DC est prête à l'emploi, sans que des composants supplémentaires soient nécessaires (voir Figure I. 18).

Pour communiquer avec une console de programmation, la CPU est équipée d'un port TCP/IP intégré.

Au travers d'un réseau ETHERNET, la CPU est capable de communiquer avec des pupitres opérateurs IHM ou avec d'autres CPU [8].

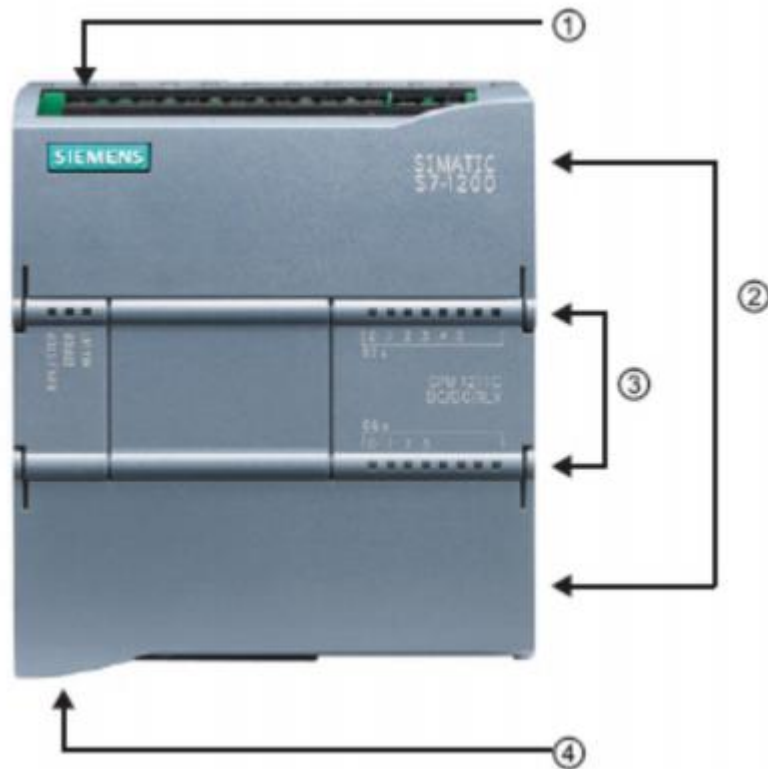


Figure I. 18: La face de la CPU 1214C DC/DC/DC

- 1) Alimentation 24V
- 2) Borniers enfichables pour un câblage utilisateur (derrière les caches plastiques)
- 3) LED d'état des E/S intégrées et pour les modes de fonctionnement de la CPU
- 4) Port TCP/IP (sous la CPU)

I.9.4. L'installation d'appareils S7-1200 :

Le matériel S7-1200 est conçu pour être facile à installer. Vous pouvez monter l'automate

S7-1200 sur un panneau ou sur un profilé support et l'orienter horizontalement ou verticalement. La petite taille du S7-1200 permet une optimisation de l'espace [7].

I.9.5. Ménagez un dégagement adéquat pour le refroidissement et le câblage:

Lors de l'implantation du S7-1200 dans votre panneau, tenez compte des appareils sources de chaleur et placez les appareils de type électronique dans les zones plus fraîches de votre armoire.

Les appareils S7-1200 sont conçus pour un refroidissement par convection naturelle. Pour que le refroidissement se fasse correctement, vous devez laisser un espace libre d'au moins 25 mm au-dessus et en dessous des appareils. Vous devez également avoir une profondeur d'au moins 25 mm entre l'avant des modules et l'intérieur de l'enceinte (voir Figure I. 19).

Lorsque vous planifiez la disposition de votre système S7-1200, réservez suffisamment

D'espace pour le câblage et les connexions de câbles de communication.

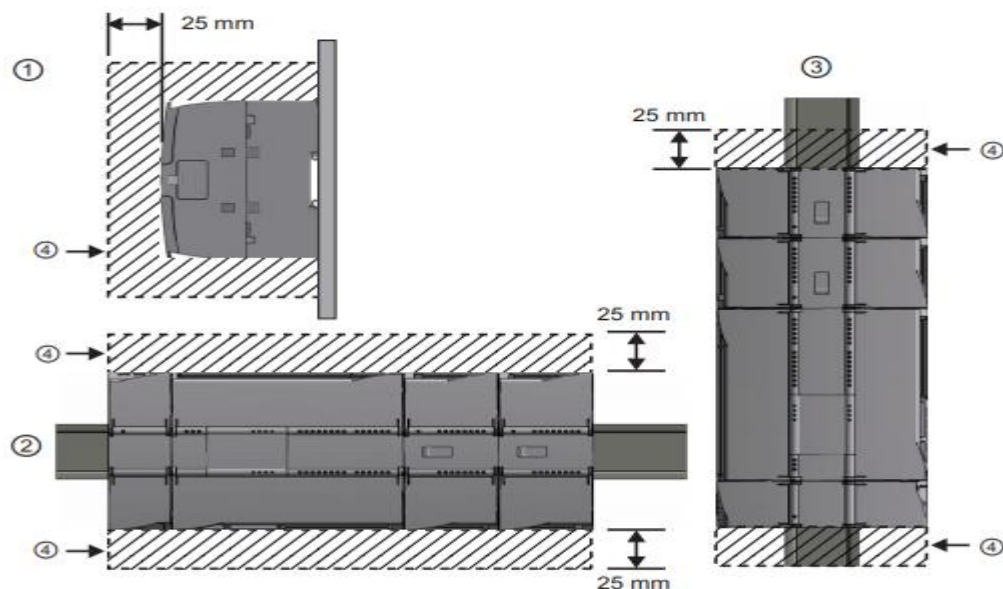


Figure I. 19: Espace pour la connexion Automate Siemens S7-1200

1) Vue de côté

- 2) Montage horizontal
- 3) Montage vertical
- 4) Zone de dégagement

Si votre système se situe dans un environnement à fortes vibrations ou est disposé verticalement, le montage du S7-1200 sur panneau offrira un niveau de protection plus élevé [7].

I.9.6. Adresse des signaux d'entrée/sortie de SIMATIC S7-1200 :

La déclaration d'une entrée ou sortie donnée à l'intérieur d'un programme s'appelle l'adressage. Les entrées et sorties des automates sont la plupart du temps regroupées en groupes de huit entrées ou sorties numériques (voir Figure I. 20). Cette unité de huit entrées ou sorties est appelée un octet. Chaque groupe reçoit un numéro que l'on appelle l'adresse d'octet.

Afin de permettre l'adressage d'une entrée ou sortie à l'intérieur d'un octet, chaque octet est divisé en huit bits. Ces derniers sont numérotés de 0 à 7. On obtient ainsi l'adresse du bit. L'automate programmable représenté ici a les octets d'entrée 0 et 1 ainsi que les octets de sortie 4 et 5 [5].

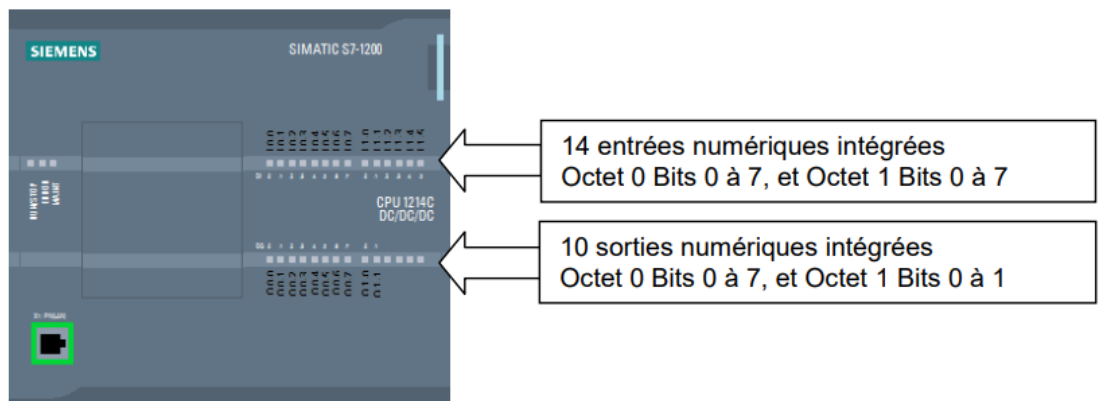


Figure I. 20: Les entrées et sorties des automates siemens S7-1200



Tableau 1 : Adresse des signaux d'entrée/sortie

Connection	Input/output	Description	Memorable name	Internal Address
DI0.0	Input	Cylinder reed switch 1	Reed 1	%I0.0
DI0.1	Input	Cylinder reed switch 2	Reed 2	%I0.1
DI0.2	Input	Cylinder reed switch 3	Reed 3	%I0.2
DI0.3	Input	Optical part sensor 1	Sensor 1	%I0.3
DI0.4	Input	Optical part sensor 2	Sensor 2	%I0.4
DI0.5	Input	Red button	Red button	%I0.5
DI0.6	Input	Green button	Green button	%I0.6
DQ0.0	Output	Cylinder 1	Cylinder 1	%Q0.0
DQ0.1	Output	Cylinder 2	Cylinder 2	%Q0.1
DQ0.2	Output	Cylinder 3	Cylinder 3	%Q0.2
DQ0.3	Output	Conveyor enable	Conveyor	%Q0.3
DQ0.4	Output	Conveyor reverse	Reverse	%Q0.4
DQ0.5	Output	Red lamp	Red lamp	%Q0.5
DQ0.6	Output	Green Lamp	Green Lamp	%Q0.6

I.9.7.1. Conditions requises :

Assurez-vous, avant de mettre à la terre ou de câbler tout appareil électrique, que cet Appareil a été mis hors tension. Assurez-vous également que tout équipement associé a été Mis hors tension [7].

I.9.8. Connexion à la CPU via le protocole TCP/IP, et retour aux paramètres d'usine:

Pour programmer le SIMATIC S7-1200 à partir d'un PC ou d'un ordinateur portable, vous avez besoin d'une connexion TCP/IP (voir Figure I. 22).

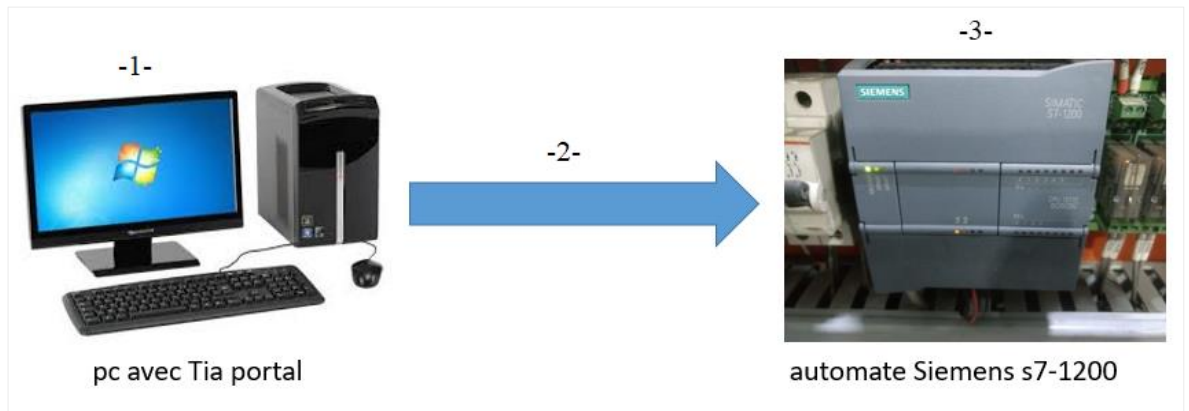


Figure I. 22: la connexion TCP/IP entre ordinateur et automate siemens s7-1200

- 1) Créer le programme de l'API avec Tia portal (Step 7) sur le PC.
- 2) Connecter le PC avec l'interface TCP/IP de l'API.
- 3) Charger le programme du PC dans la mémoire de l'API [5].

I.9.9. Modules d'extension S7-1200 :

Le SIMATIC S7-1200 est un automate modulaire et tout un éventail de modules l'accompagnent. Les voici :

- Modules centraux CPU (Central Processing Unit) avec différentes capacités, entrées/sorties intégrées, et une interface PROFINET (par exemple, la CPU 1214C) (voir Figure I. 23).



Figure I. 23: Central Processing Unit (CPU)

- Module de puissance PM (Power Module) avec une entrée AC 120/230V, 50Hz/60Hz, 1.2A/0.7A, et une sortie DC 24V/2.5A (voir Figure I. 24)



Figure I. 24: Power Module (PM)

- « Signal Boards » SB pour ajouter des entrées ou sorties analogiques ou numériques, la taille de la CPU étant fixée (voir Figure I. 25)



Figure I. 25: Signal Boards (SB)

(Les « signal Boards » peuvent être utilisés avec les CPU 1211C/1212C et 1214C)

- Modules de signal SM (Signal Module) pour les entrées et sorties analogiques et numériques (voir Figure I. 26)



Figure I. 26: Signal Module (SM)

(Pour les CPU 1212C un maximum de 2 SM Peuvent être utilisés, pour la 1214C max.8)

- Modules de communication CM (Communication Module) pour une communication série RS 232 / RS 485 (voir Figure I. 27)



Figure I. 27: Communication Module (CM)

- Les cartes mémoire 2Mo ou 24Mo pour stocker les données du programme et pour un remplacement simple des CPU lors des maintenances (voir Figure I. 28) [5]



Figure I. 28: Les cartes mémoire

I.9.10. Modes de fonctionnement de la CPU :

La CPU peut se trouver dans l'un des trois modes de fonctionnement suivants :

- En mode STOP, la CPU n'exécute pas le programme et vous pouvez charger un projet.
- En mode STARTUP, la CPU entame une procédure de démarrage.
- En mode RUN, le programme est exécuté de façon cyclique.

La CPU n'a pas de commutateur physique pour modifier de mode de fonctionnement.

Le mode STOP ou RUN se modifie par le pupitre opérateur du logiciel STEP 7 Basic. De plus, le Pupitre opérateur est muni d'un bouton MRES pour effectuer un effacement général. Il affiche Aussi les LED d'état de la CPU (voir Figure I. 29) [8].

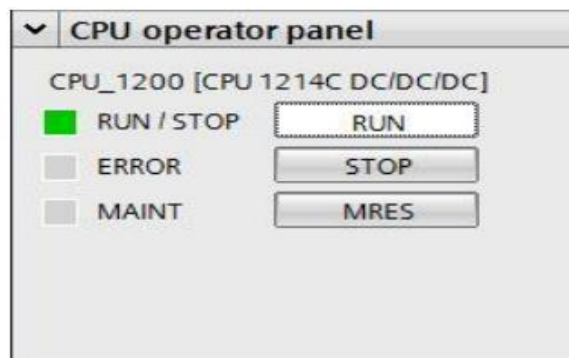


Figure I. 29: Le mode STOP ou RUN et bouton MRES

I.9.11. Visualisations d'état et d'erreur :

La couleur de la LED d'état RUN/STOP sur la face avant de la CPU indique le mode de Fonctionnement actuel (voir Figure I. 30).



Figure I. 30: le mode de Fonctionnement de la CPU (les LED d'états RUN/STOP)

- Une lumière jaune indique le mode STOP.
- Une lumière verte indique le mode RUN.
- Une lumière clignotante indique le mode STARTUP [4].

En outre, les DEL ERROR et MAINT indiquent respectivement si une erreur est survenue et si Une maintenance est requise [8].

I.9.12. Choix de la CPU :

I.9.12.1 Le choix de Platines d'extension, modules d'E/S, modules de Communication :

Les CPU du système SIMATIC S7-1200 se déclinent en trois classes de performances : CPU 1211 C, CPU1212 C et CPU 1214 C, chacune d'elles pouvant être étendue en fonction des besoins de la machine. Sur chaque CPU, il est possible de greffer une platine d'extension pour ajouter des E/S TOR ou analogiques supplémentaires sans modification de l'encombrement de l'automate. Des modules d'E/S supplémentaires peuvent être ajoutés du côté droit de la CPU pour étendre la capacité d'E/S TOR ou analogiques. La CPU 1212 C supporte deux modules et la CPU 1214 C en supporte huit. Il est en outre possible, sur toutes les CPU SIMATIC S7-1200, de connecter jusqu'à trois modules de communication du côté gauche de la CPU, ce qui permet tous les types de communication: PROFINET, PROFIBUS, AS-i, communication série, WAN ou GPRS [9].

I.10. Conclusion :

Dans ce chapitre nous avons donné une présentation générale des Automates Programmables Industriels (API). Pour cela, nous avons présenté les différentes architectures et le principe de fonctionnement. Par la suite, nous avons présenté les différentes méthodes de contrôle et les langages de programmation les plus utilisés. Finalement, nous avons décrit l'automate Siemens S7-1200 en donnant le choix approprié du CPU et le câblage nécessaire.



Chapitre II :
*Description du système
à bande transporteuse*

II.1. Introduction :

Les systèmes automatisés sont largement utilisés dans l'industrie. Les API (Automate Programmable Industrielle) représentent l'organe essentiel dans un système automatisé. Ils ont été introduits au cours des années 1980 et ont révolutionné la façon dont les processus automatisés étaient contrôlés.

Un exemple concret d'un système automatisé est illustré dans le principe de fonctionnement d'une ligne de production qui contient un bande transporteuse contrôlé par un automate programmable industriel pour déplacer les pièces au-delà de deux capteurs de faisceaux lumineux afin de les stockés dans un bac (voir Figure I. 17). Cette bande transporteuse s'arrête lorsque le bac de stockage contient un nombre préprogrammé de pièces. L'automaticien est l'ingénieur chargé d'assurer la programmation de l'API pour atteindre cet objectif. Dans une chaine automatisée l'automate programmable prend l'information du capteur sous forme d'un signal pour faire la décision logique est envoie par la suite le signal de commande à l'actionneur pour concrétiser la décision prise [10].



Figure II. 1: Banc d'essai d'une bande transporteuse [10]

II.2. Description du système:

II.2.1. Au niveau matériel (hardware) :

II.2.1.1. Alimentation Electrique:

Le banc d'essai de la bande transporteuse dispose une alimentation externe de 12 V qui doit être connectée à la prise d'entrée sur le côté droit du système. Le commutateur à côté de la prise d'alimentation permet la mise en service ou bien la mise à l'arrêt de l'unité principale, y compris l'électronique interne et les différents dispositifs, tels que le moteur et les lampes. Les quatre commutateurs de défaut situé à droite du port USB doivent être en position basse pour un fonctionnement normal (voir Figure I. 2). Une fois l'appareil est mis sous tension et allumé, un commutateur est utilisé pour sélectionner le mode de fonctionnement manuel ou automatique (voir Figure I. 3).



Figure II. 2: les connexions du prototype « Industrial control work-cell » [10]



Figure II. 3: Commutateur de mode de fonctionnement (manuel/automatique) [10]

II.2.1.2. Moteurs à courant continu :

Un moteur à courant continu est utilisé pour entrainer le tapis roulant (bande transporteuse) (voir Figure I. 4). Ce moteur est associé à un variateur de vitesse statique (variateur électronique) dont la technologie est plus simple et peu onéreuse tout en demandant peu d'entretien et en offrant des performances élevées dans une plage de vitesse très large (de 1 à 100 %) [2]. Dans ce qui suit nous présentons le principe et les différents caractéristiques du MCC.



Figure II. 4: moteur à courant continu

II.2.1.2.1. Principe de fonctionnement :

Le moteur à courant continu se compose [11]:

- de l'inducteur ou du stator,
- de l'induit ou du rotor,
- du collecteur et des balais.

Lorsque le bobinage d'un inducteur de moteur est alimenté par un courant continu, sur le même principe qu'un moteur à aimant permanent (voir Figure I. 5), il crée un champ magnétique (flux d'excitation) de direction Nord-Sud.

Une spire capable de tourner sur un axe de rotation est placée dans le champ magnétique. De plus, les deux conducteurs formant la spire sont chacun raccordés électriquement à un demi collecteur et alimentés en courant continu via deux balais frotteur.

Si le système balais-collecteurs n'était pas présent (simple spire alimentée en courant continu), la spire s'arrêterait de tourner en position verticale sur un axe appelé communément "ligne neutre". Le système balais-collecteurs a pour rôle de faire commuter le sens du courant dans les deux conducteurs au passage de la ligne neutre. Le courant étant inversé, les forces motrices sur les conducteurs le sont aussi permettant ainsi de poursuivre la rotation de la spire [11].

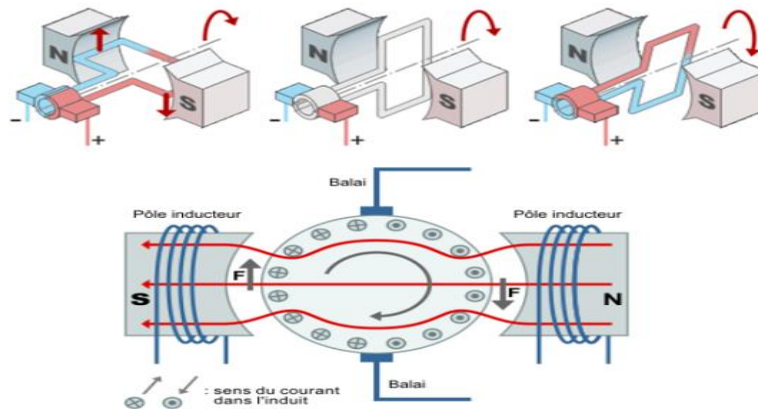


Figure II. 5: principe de fonctionnement d'un moteur à courant continu

II.2.1.2.2. Caractéristiques:

Les avantages et inconvénients du moteur à courant continu sont repris ci-dessous [11]:

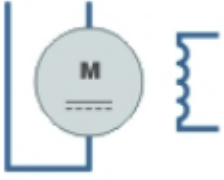
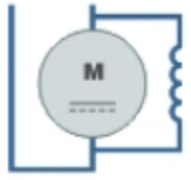

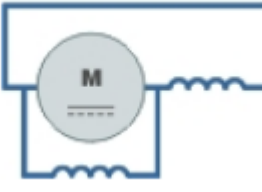
- **Les avantages :**
- accompagné d'un variateur de vitesse électronique, il possède une large plage de variation (1 à 100 % de la plage).
- régulation précise du couple.
- son indépendance par rapport à la fréquence du réseau fait de lui un moteur à large champ d'application.
- **Les inconvénients :**

- peu robuste par rapport à la machine asynchrone.
- investissement important et maintenance coûteuse (entretien du collecteur et des balais).

II.2.1.2.3. Type de moteur à courant continu:

Suivant l'application, les bobinages du l'inducteur et de l'induit peuvent être connectés de manière différente. On retrouve en général [11]:

Tableau 2 : Différentes façons de se connecter les bobinages du l'inducteur et de l'induit

Moteur à excitation indépendante	Moteur à excitation parallèle
	
Moteur à excitation série	Moteur à excitation composée
	

La plupart des machines d'ascenseur sont configurées en excitation parallèle ou indépendante. L'inversion du sens de rotation du moteur s'obtient en inversant soit les connexions de l'inducteur soit de l'induit [11].

II.2.1.2.4. Variation de la vitesse :

Au vu des relations existant entre la vitesse, le flux et la force contre-électromotrice, il est possible de faire varier la vitesse du moteur de deux manières différentes [11]. On peut:

- Augmenter la force contre-électromotrice E en augmentant la tension à la borne de l'induit tout en maintenant le flux de l'inducteur constant. On a un fonctionnement dit à "couple constant". Ce type de fonctionnement est intéressant au niveau de la conduite d'ascenseur.
- Diminuer le flux de l'inducteur (flux d'excitation) par une réduction du courant d'excitation en maintenant la tension d'alimentation de l'induit constante. Ce type de fonctionnement impose une réduction du couple lorsque la vitesse augmente.

II.2.1.2.5. Connexion/déconnexion de la commande du moteur analogique :

Le moteur de la bande transporteuse comporte trois commandes :

- Activer (E) – Démarrer/arrêter le moteur de la bande transporteuse
- Marche arrière (R) – Contrôle de la direction de la bande transporteuse
- Vitesse (M) – Commande de la vitesse du moteur de la bande transporteuse

Le contrôle de la vitesse de la bande transporteuse nécessite une programmation plus complexe, de sorte que certaines tâches pratiques nécessitent le retrait de la connexion analogique de contrôle de la vitesse. Utiliser un tournevis pour retirer la connexion du bornier M (voir Figure I.6).

Une fois la connexion analogique retirée et le moteur activé, la bande transporteuse fonctionnera à pleine vitesse.

Pour les tâches pratiques qui utilisent le contrôle de vitesse analogique, il suffit de rebrancher le fil [10].

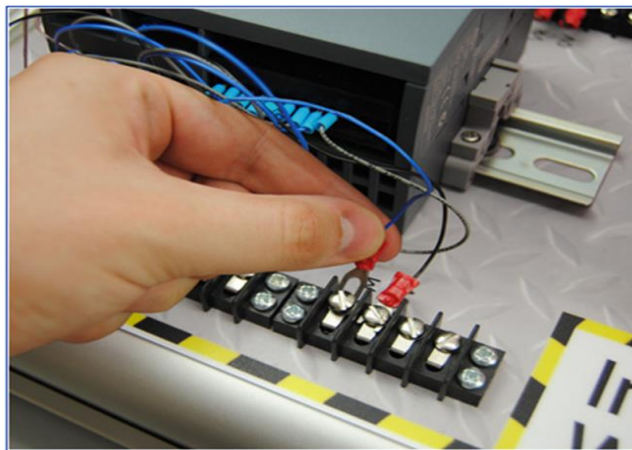


Figure II. 6: Connexion / déconnexion de la commande moteur [10]

II.2.1.3. Bande transporteuse:

Une bande transporteuse, ou courroie transporteuse ou convoyeur à bande ou encore transporteur à bande, est un dispositif de transport ou de manutention permettant le déplacement continu de marchandises en vrac ou de charges isolées. C'est un outil très pratique, qui est encore assez méconnu des professionnels dans le secteur de l'industrie. Elle peut prendre la forme d'un tapis roulant à bacs, ou encore de fonds mobile pour les engins auto-déchargeurs.

Elle est constituée essentiellement d'une bande en matériau souple entraînée et supportée par des poulies motorisées. Lequel supporte et entraîne la marchandise posée dessus (voir Figure I. 7). Elle peut être munie de nervures en chevrons permettant un meilleur entraînement de la marchandise.

Le parcours de la bande transporteuse peut être horizontal ou ascendant ou descendant. Il peut être rectiligne ou comporter des courbes. La plupart du temps, elle sert au transport et au déchargement de matériaux lourds. Dans le domaine du BTP, elle sert au transport de produits tels le charbon, le minerai, le sable, et bien d'autres encore. Des mécanismes de bande transporteuse sont utilisés comme composants dans les systèmes de distribution et d'entreposage automatisés. Combinés à des équipements de manutention de palette commandés par ordinateur, ils permettent une distribution plus efficace des produits manufacturés, de détail ou de gros. Ces systèmes permettent de traiter rapidement des volumes de marchandises plus importants tant en réception qu'en expédition, avec des volumes de stockage plus réduits, autorisant d'intéressants gains de productivité aux entreprises [12].



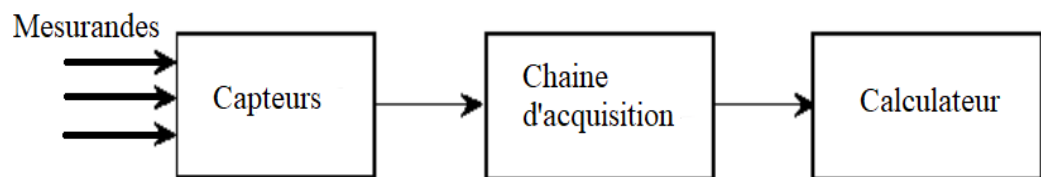
Figure II. 7: Convoyeurs à bande industriels

II.2.1.4. Les capteurs :

II.2.1.4.1. Définition :

Un capteur est un dispositif qui transforme une grandeur physique d'entrée, appelée mesurande, en une grandeur de nature électrique [13].

II.2.1.4.2. Localisation du capteur dans la chaîne :



- Les capteurs sont les premiers éléments de la chaîne.
- Ce sont les interfaces entre le "monde physique " et le "monde électrique" [13].

II.2.1.4.3. Classification des capteurs :

Les capteurs présentent des caractéristiques différentes, leurs Différences peuvent se faire selon [13]:

- Le principe de fonctionnement :
- capteurs actifs.
- capteurs passifs.

Le principe de traduction du mesurande :

- capteur résistif.
- capteur à effet Hall.

II.2.1.4.4. Capteurs actifs :

Ces capteurs sont basés sur des effets physiques permettant de transformer directement le mesurande en grandeur électrique [13].

Exemples :

Mesurande	Effet	Grandeur de sortie
Température	Thermoélectricité	Tension
Force, pression	Piézo-électricité	Charge

II.2.1.4.5. Capteurs passifs :

Le capteur est un matériau utilisé en tant qu'impédance dont l'un des paramètres est sensible au mesurande [13].

Exemples :

Mesurande	Caractéristique électrique	Matériau
Température	Résistivité	Cuivre, platine
Flux optique	Résistivité	Semi-conducteurs

La mesure de l'impédance permet ensuite de déduire la valeur du mesurande. Cette mesure nécessite l'utilisation d'un conditionneur.

II.2.1.4.6. Système barrage :

Caractéristiques du système barrage (voir Figure I. 8) [15]:

- 2 boîtiers
- Émetteur: qui envoie le faisceau lumineux
- Récepteur: celui qui reçoit le faisceau lumineux
- Portée : 30m
- Ne détecte pas les objets transparents

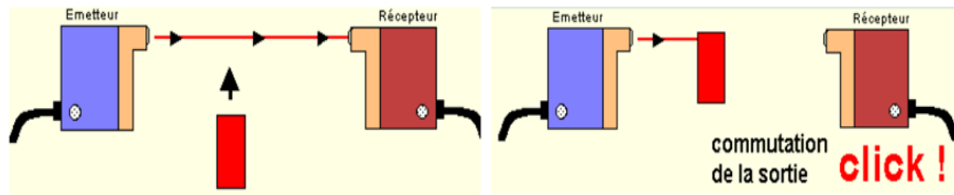


Figure II. 8: System barrage [15]

II.2.1.4.7. Modification de la hauteur du capteur de faisceau optique :

Les émetteurs (T) et les récepteurs (R) sont fixés au carter du convoyeur à l'aide de vis à molette. Leur hauteur peut être modifiée indépendamment en desserrant les vis et en déplaçant les supports de capteur vers le haut ou vers le bas (voir Figure I. 9). Les capteurs infrarouges peuvent être réglés sur deux positions, basse et haute. Dans la position basse, les parties courtes et hautes sont détectées. En position haute, seules les parties hautes sont détectées (voir Figure I. 10).

L'émetteur et le récepteur de chaque ensemble de capteurs doivent être changés simultanément. Après avoir réglé la position des capteurs, utiliser les voyants d'état des capteurs pour s'assurer que le jeu de capteurs est correctement aligné avant (et après) le serrage des vis de serrage. Si le faisceau du capteur est ininterrompu, le voyant correspondant doit être allumé [10].

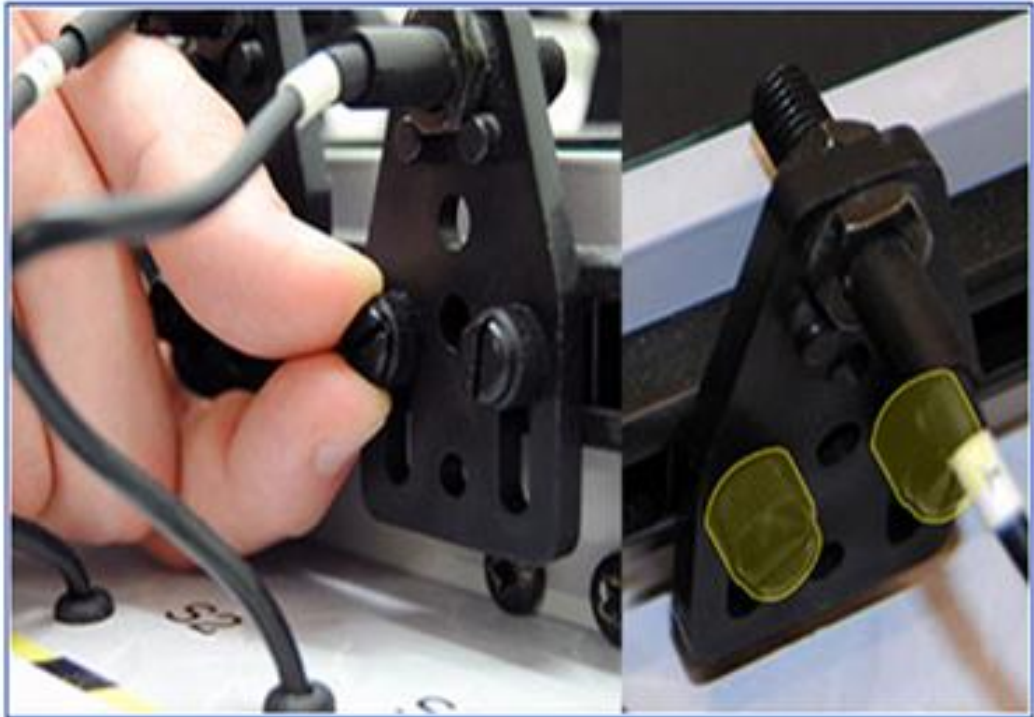


Figure II. 9: Modification de niveau du capteur [10]



Figure II. 10: Capteur de faisceau optique

II.2.1.4.8. Modification de niveau du capteur dans le Simulateur:

Dans le simulateur, la hauteur peut basculer entre les deux positions en cliquant simplement sur le capteur une fois le programme lancé (voir Figure I. 11) [10].

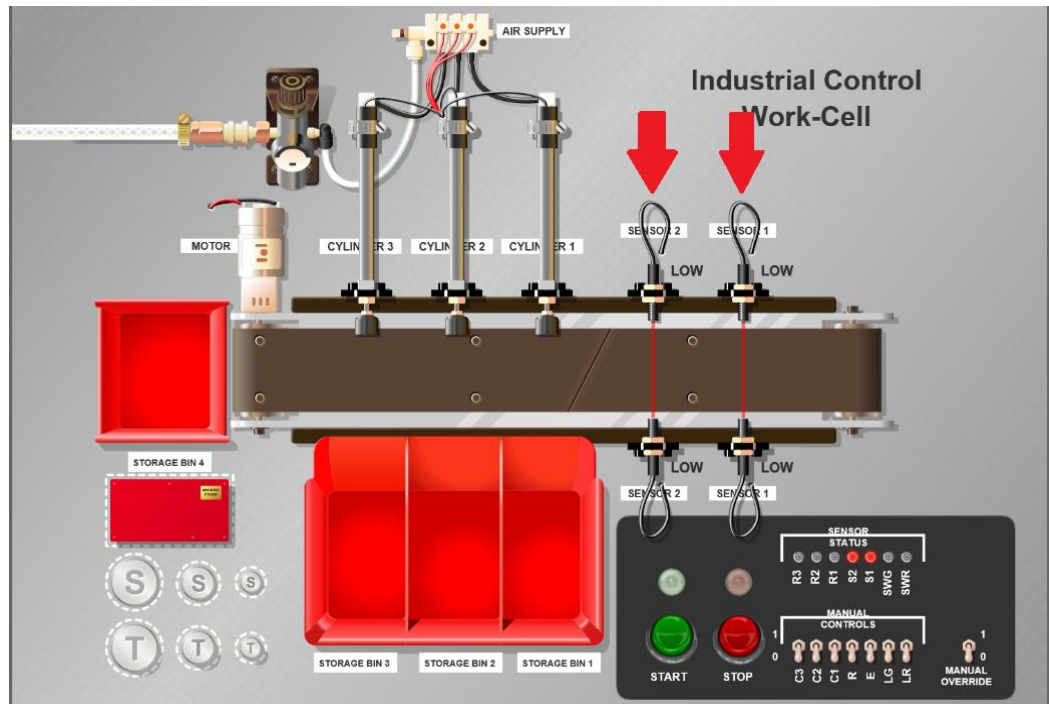


Figure II. 11: Réglage du capteur Dans le simulateur [10]

II.2.1.5. Source de pression et réservoir de d'air comprimé :

La sélection des pièces dans ce système est faite à l'aide des trois vérins pneumatiques à simple effet (voir Figure I. 12). Ces vérins assurant la conversion de l'énergie pneumatique en énergie mécanique sous forme de mouvement de la tige du vérin pour faire la sélection des pièces. Le générateur de pression utilisé dans ce système est une pompe à air muni d'un réservoir d'air comprimé (voir Figure I. 13 (a)). Une pression de 0,2 Mpa est suffisante pour faire travailler les pistons des vérins. Cette valeur peut être lue directement sur un manomètre qui se trouve dans l'unité FLR (voir Figure I. 13 (b)).

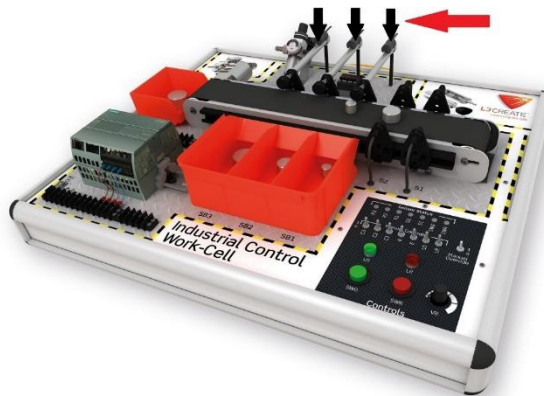


Figure II. 12: vérins pneumatiques à simple effet

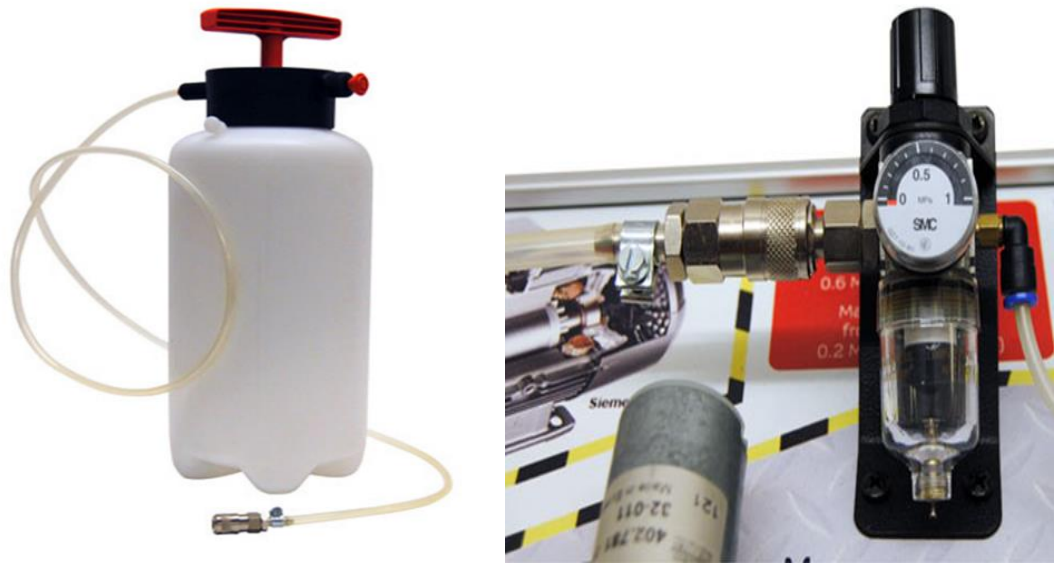


Figure II. 13: (a) Pompe à Air et réservoir de pression (b) unité de conditionnement d'air FLR[1]

L'unité de conditionnement d'air est aussi appelé FRL (Filtre Régulateur Librifiant). Une unité de conditionnement d'air délivre un air conditionné, c'est à dire, filtré et huilé ou non. En aucun cas, un FRL ne délivrera un air "sec" (il faut un assécheur pour cela). La désignation "marketing" de l'air sec, signifie que l'air n'est pas huilé.

II.2.1.6. Les vérins :

II.2.1.6.1. Principe:

Un vérin pneumatique est un actionneur qui permet de transformer l'énergie de l'air comprimé en un travail mécanique. Un vérin pneumatique est soumis à des pressions d'air comprimé qui permettent d'obtenir des mouvements dans un sens, puis dans l'autre. Les mouvements obtenus peuvent être linéaires ou rotatifs [14].

II.2.1.6.2. Applications:

Cet actionneur de conception robuste et de simplicité de mise en œuvre est utilisé dans toutes les industries manufacturières. Il permet de reproduire les actions manuelles d'un opérateur telles que pousser, tirer, plier, serrer, soulever, poinçonner, positionner, etc...

Les croquis ci-dessous (voir Figure I. 14) évoquent les principaux emplois des vérins pneumatiques en automatisation de production [14].

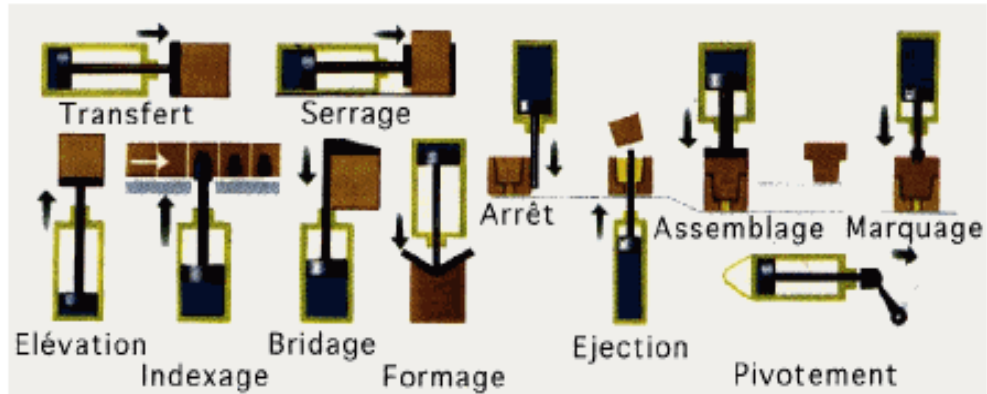


Figure II. 14: Emplois des vérins [14]

II.2.1.6.3. Constitution d'un vérin:

Un piston muni d'une tige se déplace librement à l'intérieur d'un tube. Pour faire sortir la tige, on applique une pression sur la face avant du piston, et sur la face arrière pour faire rentrer la tige (voir Figure I. 15) [14].

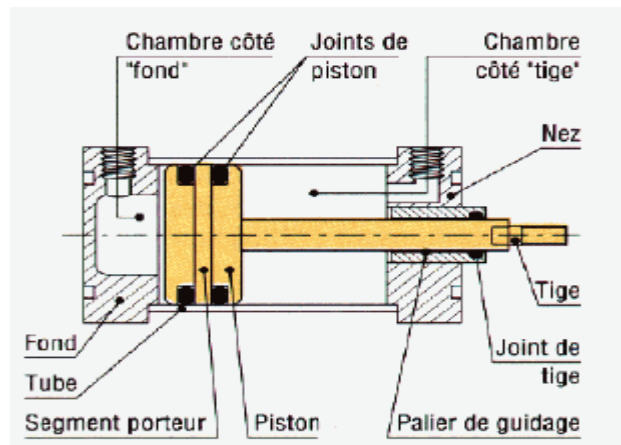


Figure II. 15: Vue en coupe d'un vérin pneumatique [14]

• Amortissement :

Certains vérins disposent d'amortisseurs afin d'obtenir un ralentissement en fin de mouvement de façon à éviter un choc du piston sur le nez ou le fond du vérin [14].

• Auxiliaires implantés sur les vérins :

Il est possible d'équiper les vérins de dispositifs de contrôle de mouvement tels que régulateurs de vitesse et capteurs de position magnétique [14].

II.2.1.6.4. Les différents types de vérins pneumatiques :

1. Vérin double effet :

Un vérin double effet a deux directions de travail. Il comporte deux orifices d'alimentation et la pression est appliquée alternativement de chaque côté du piston ce qui entraîne son déplacement dans un sens puis dans l'autre (voir Figure I. 16).

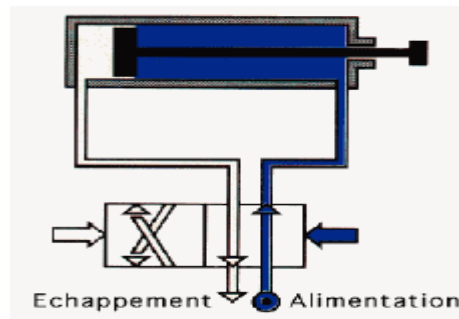


Figure II. 16: Vérin double effet avec son distributeur [14]

L'air comprimé est distribué par un distributeur à deux sorties [14].

2. Vérin simple effet :

Un vérin simple effet ne travaille que dans un sens. L'arrivée de la pression ne se fait que sur un seul orifice d'alimentation ce qui entraîne le piston dans un seul sens, son retour s'effectuant sous l'action d'un ressort (voir Figure I. 17).

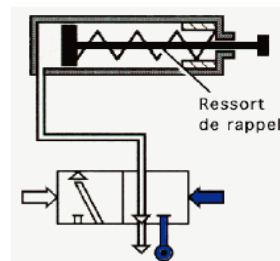


Figure II. 17: Vérin simple effet avec son distributeur [14]

Un distributeur à une seule sortie est donc suffisant. L'emploi de ces vérins reste limité aux faibles courses [14].

II.2.1.6.5. Caractéristiques d'un vérin :

Un vérin est déterminé par sa **course** et par son **diamètre** [14]:

- de sa course dépend la longueur du déplacement à assurer,
- de son diamètre et de la pression de l'air dépend l'effort à développer.

II.2.1.6.6. Les vérins dans La bande transporteuse Industrial Control Work-Cell :

La bande transporteuse Industrial Control Work-Cell contient trois vérins pneumatiques à simple effet (V.S.E.) (voir Figure I. 18)

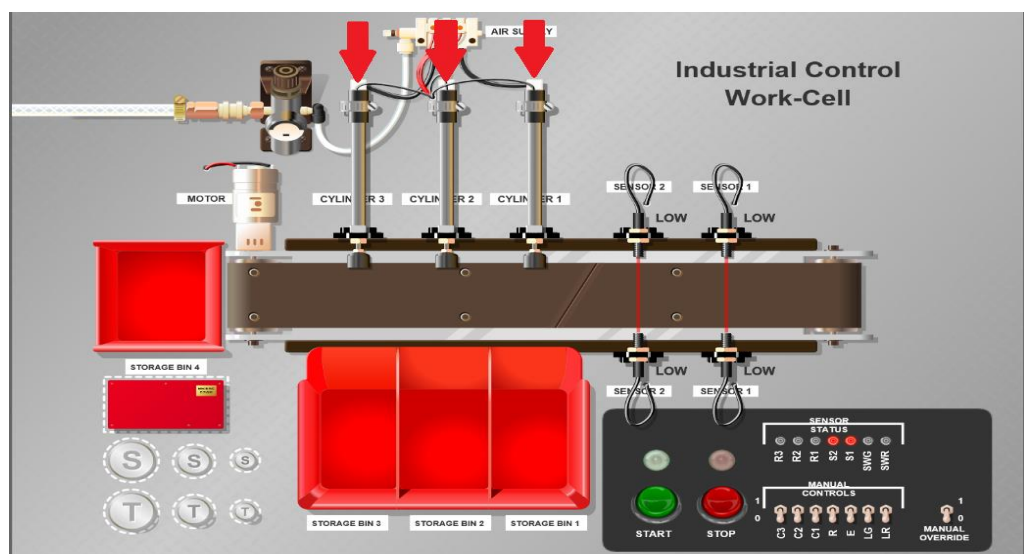


Figure II. 18: vérins pneumatiques à simple effet (V.S.E.) [10]

II.2.1.7. Connexions PLC :

Pour que le PLC puisse contrôler les différents éléments du banc d'essai de bande transporteuse, ses entrées et sorties doivent être connectées aux bornes appropriées du bande transporteuse. En outre, les besoins en puissance du PLC sont pris à partir de la source de tension. Les sections suivantes (voir Figure I. 19) identifient les connexions individuelles pour des PLC spécifiques [10].

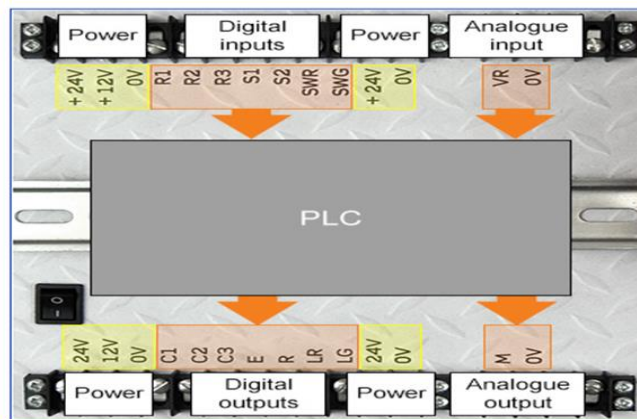


Figure II. 19: Connexions PLC [10]

II.2.1.8. Mise en marche et arrêt de l'unité API:

Le commutateur sur le panneau supérieur contrôle l'alimentation de l'unité PLC attachée (voir Figure I. 20). Avec un PLC attaché et ce commutateur en position de marche, le PLC contrôlera l'appareil (sauf si le commutateur de priorité manuelle a été activé) [10].

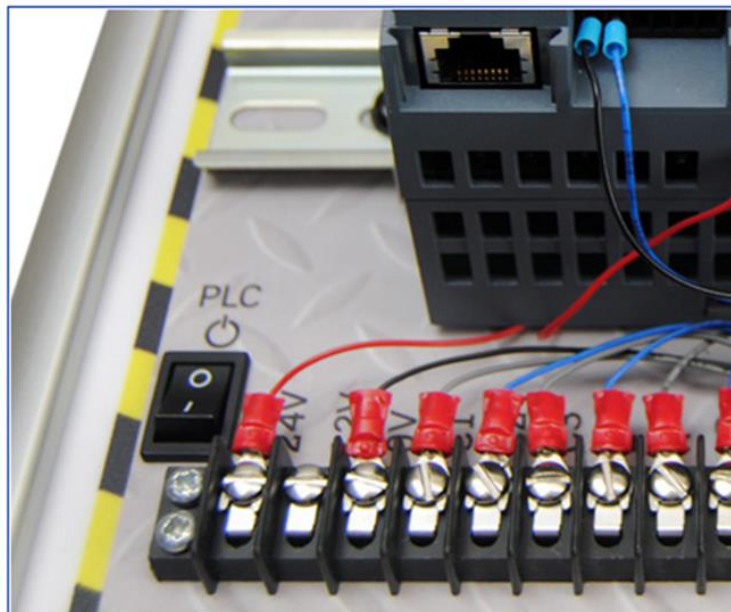


Figure II. 20: Interrupteur marche / arrêt de l'unité PLC [10]

II.2.1.9. Les modes De Contrôle :

II.2.1.9.1. Mode manuel :

Lorsque le commutateur de commande manuelle est réglé sur la position 1 Comme nous l'avons dit précédemment, la bande transporteuse est contrôlée uniquement à partir du jeu de commutateurs dans la section du panneau de commande [10].

Tableau 3 : Commutateurs de commande Manuels

Manuel Control	Action (Anglais)	Action (Français)
SWG	Green pushbutton	Bouton poussoir vert
SWR	Red pushbutton	Bouton poussoir rouge
S1	Optical sensor 1	Capteur optique 1
S2	Optical sensor 2	Capteur optique 2
R1	Cylinder 1 extended	Vérin 1 allongé
R2	Cylinder 2 extended	Vérin 2 allongé
R3	Cylinder 3 extended	Vérin 3 allongé
LG	Illuminate green lamp	Lampe verte allumée
LR	Illuminate red lamp	Lampe lampe rouge
E	Enable convoyer Motors	Activer les moteurs de convoyeur
R	Reverse convoyer Motors	Moteurs de convoyeur inversé
C1	Extend cylinder 1	Prolonger le cylindre 1
C2	Extend cylinder 2	Prolonger le cylindre 2
C3	Extend cylinder 3	Prolonger le cylindre 3

Le panneau de commande du simulateur affiche l'état actuel du capteur [S1, S2] en tout temps à l'aide d'une série de lampes [LG, LR]. En mode manuel, le cadran VR contrôle la vitesse de la bande transporteuse (voir Figure I. 21).



Figure II. 21: variateur de vitesse d'une bande transporteuse (VR) [10]

Lorsque la commande manuelle est désactivée (0), les positions de la commande sont ignorées et la commande revient au PLC [10].

II.2.1.9.2. Mode automatique :

Lorsque le commutateur de commande manuelle est réglé sur la position 0, la bande transporteuse est contrôlée à partir de l'automate. Les programmes sont créés et exécutés avec une application de simulation API.

Un logiciel de logique à relais simple prend un signal d'entrée, prend une décision logique sur la base des informations fournies et exécute la décision en envoyant un signal au déclencheur de sortie (voir Figure I. 22) [10].



Figure II. 22: PLC programs [10]

II.2.1.10. Zone de stockage :

Une zone de stockage est un lieu de stockage au sein d'un entrepôt pouvant contenir plusieurs emplacements de stockage. Une zone peut être associée à des caractéristiques de stockage spécifiques.

Une zone de stockage permet de spécifier dans le système l'emplacement physique des articles (voir Figure I. 23). Vous pouvez préciser les caractéristiques de stockage de ces zones, comme l'emplacement physique. Les caractéristiques de la zone sont comparées à celles des articles pour déterminer l'emplacement de stockage approprié. Pour créer et gérer ces caractéristiques [7].

Il existe quatre types de zones de stockage dans Le prototype « Industrial control work-cell » :

- SB1
- SB2
- SB3
- SB4

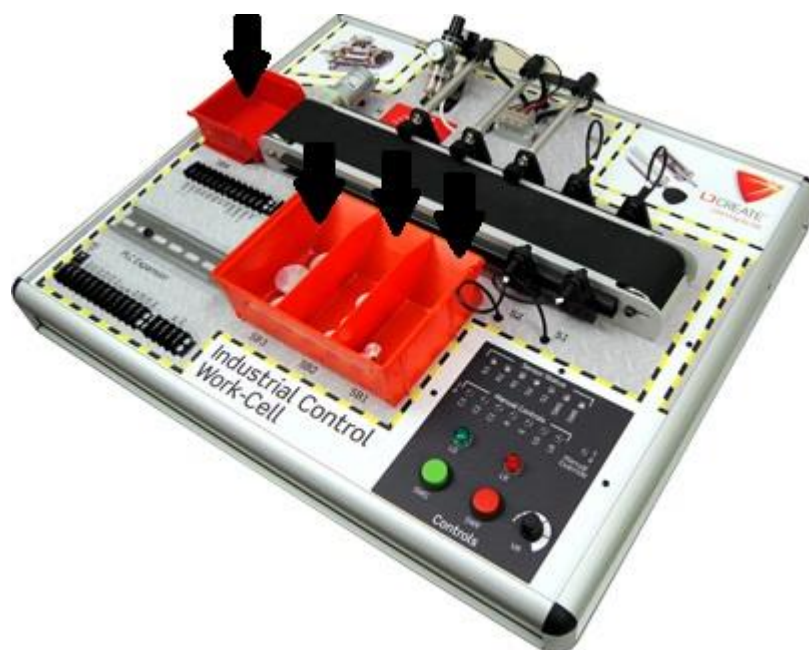


Figure II. 23: les zones des stockages [10]

II.2.2. En niveau du logiciel (software) :

L'application Programmable Logic Control permet d'explorer de nombreuses fonctionnalités d'un PLC industriel.

L'application Programmable Logic Control prend en charge l'entrée de programmes de logique d'échelle pour contrôler soit un simulateur matériel connecté, ou elle peut être utilisée en mode de simulation où les fonctions du formateur sont simulées à l'écran (voir Figure I.24).

L'application peut être utilisée en conjonction avec un bande transporteuse physique compatible, comme l'unité de cellule de travail de contrôle industriel PC61.

En plus de L'application Programmable Logic Control, nous avons d'autres applications de contrôle qui sont:

- TIA Portal ou Totally integrated automation c'est un environnement de développement tout en un permettant de programmer non seulement des automates mais aussi des afficheurs industriels (HMI). Il est constitué de :
- Step7 fait partie de la suite TIA Portal et permet seulement de programmer des automates.
- Win cc (permettant de programmer des afficheurs Siemens).
- TIA Portal intègre aussi la gestion des fonctionnalités motion, comptage etc...[17].

L'application doit être installée sur chaque poste de travail et nécessite :

Système d'exploitation Microsoft Windows ou version ultérieure Autorisations suffisantes pour effectuer des installations d'applications [10].

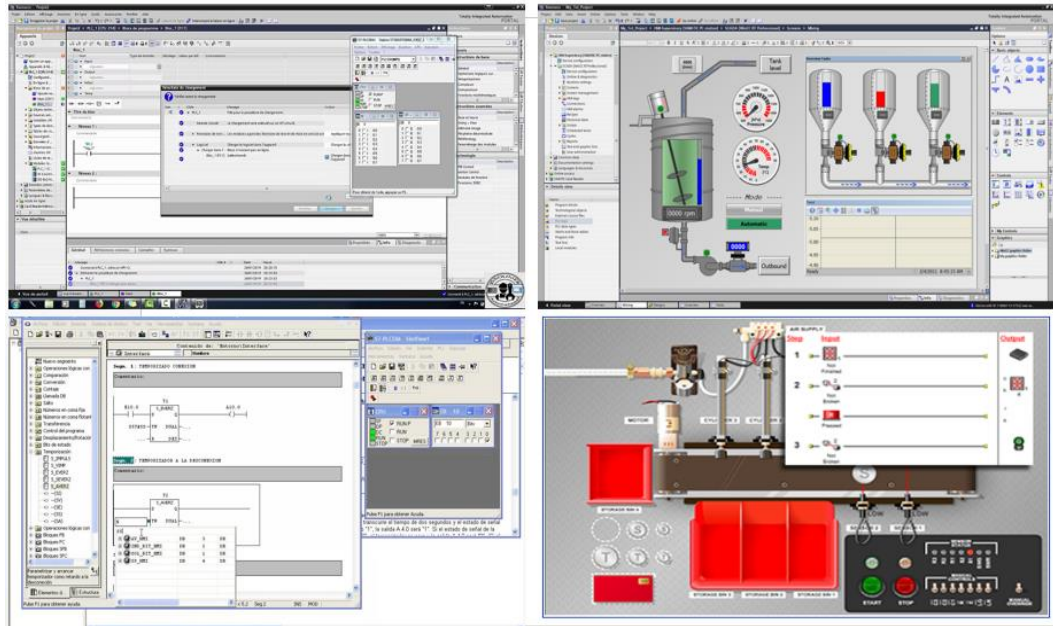


Figure II. 24: Les logiciels des contrôles logiques programmables du prototype «Industrial control work-cell »

II.3. Conclusion:

Dans ce chapitre, nous avons étudié la conception de banc d'essai de la bande transporteuse, les processus les plus importants qu'elle contient, le principe de fonctionnement et différents méthodes de contrôle.



Chapitre III:
Présentation logiciel
TIA Portal & WinCC

III.1. Introduction :

La plateforme Totally Integrated Automation Portal (TIA Portal) est le nouvel environnement de travail Siemens qui permet de mettre en œuvre des solutions d'automatisation avec un système d'ingénierie intégré comprenant les logiciels SIMATIC STEP 7 V15.1 et SIMATIC WinCC V15.1.

III.1.1. Les interfaces de logiciel :

Lorsque l'on lance TIA Portal, l'environnement de travail se décompose en deux types de vue:

III.1.1.1. Interface du portail:

L'interface du portail fournit une vue d'ensemble du projet et un accès aux outils qui permettent de l'élaborer. Vous pouvez trouver rapidement ce que vous souhaitez faire, et appeler l'outil qui servira à accomplir la tâche voulue. Si vous le souhaitez, un changement vers la vue du projet s'effectue automatiquement pour la tâche sélectionnée. Cette vue simplifie donc principalement la préparation et la mise en place du projet.

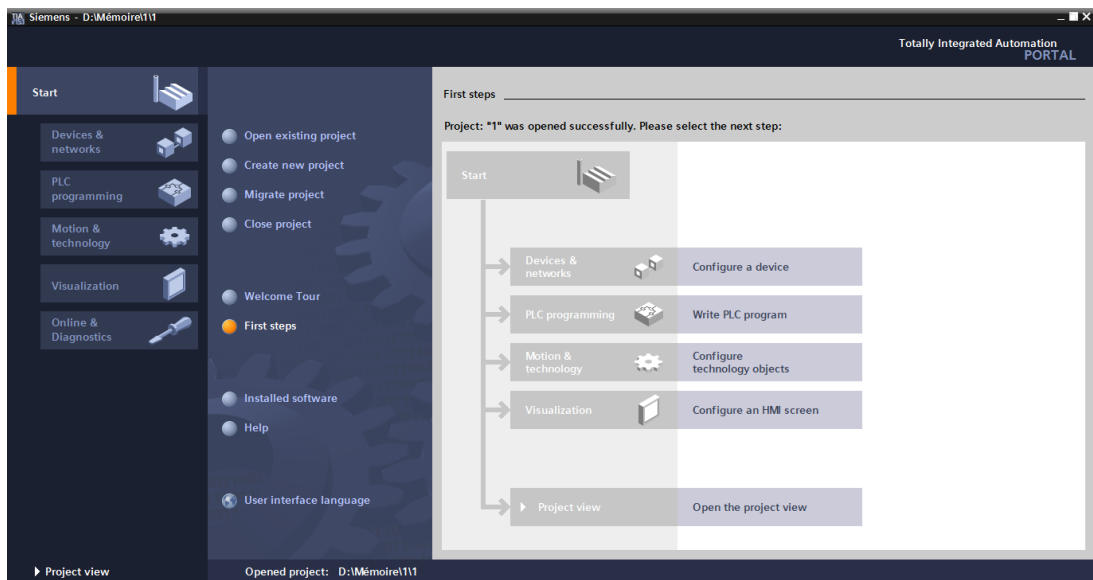


Figure III. 1: Interface du portail

III.1.1.2. Interface du projet :

L'interface du projet est une vue structurée de tous les éléments constituant le projet. La barre de menu avec les barres de fonction est située en haut de la fenêtre, le navigateur du projet

et tous les éléments du projet sont sur la gauche. Les menus associés aux différentes tâches avec les instructions et les bibliothèques sont sur la droite de la fenêtre.

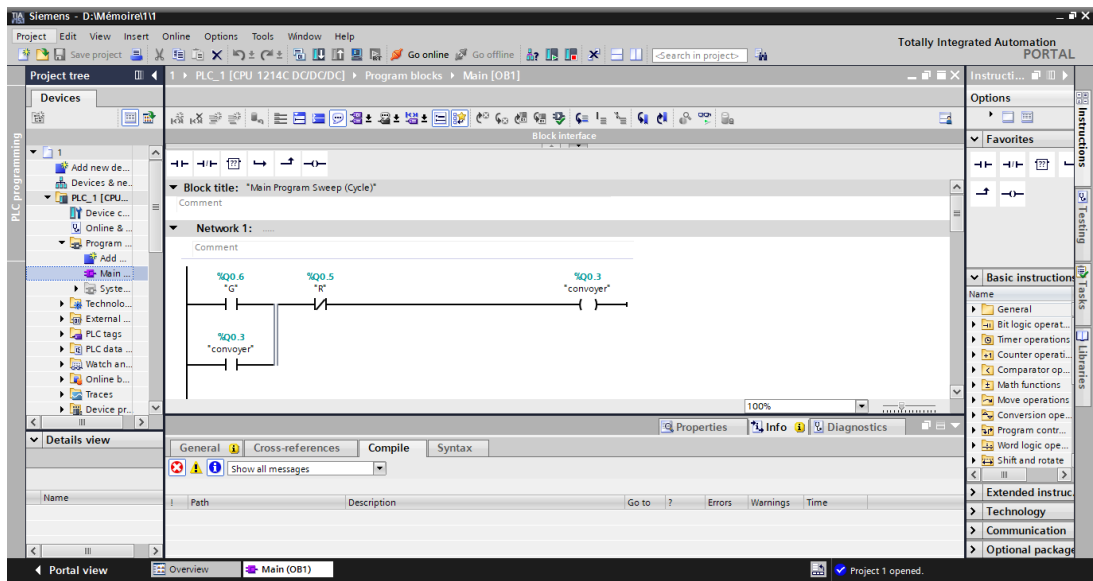


Figure III. 2: L'interface du projet

III.1.2. Création d'un projet et configuration d'une station de travail :

III.1.2.1. Création d'un projet :

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

- 1) L'outil que nous allons utiliser est « Totally Integrated Automation Portal », que l'on appelle ici d'un double-clic.



Figure III. 3: icône program Tia portal dans l'ordinateur

- 2) Les programmes pour SIMATIC S7-1200 sont gérés sous forme de projets. Nous allons maintenant créer un nouveau projet via la vue portail « **Create new projet** > **Nom** : Initiation > **Create** »

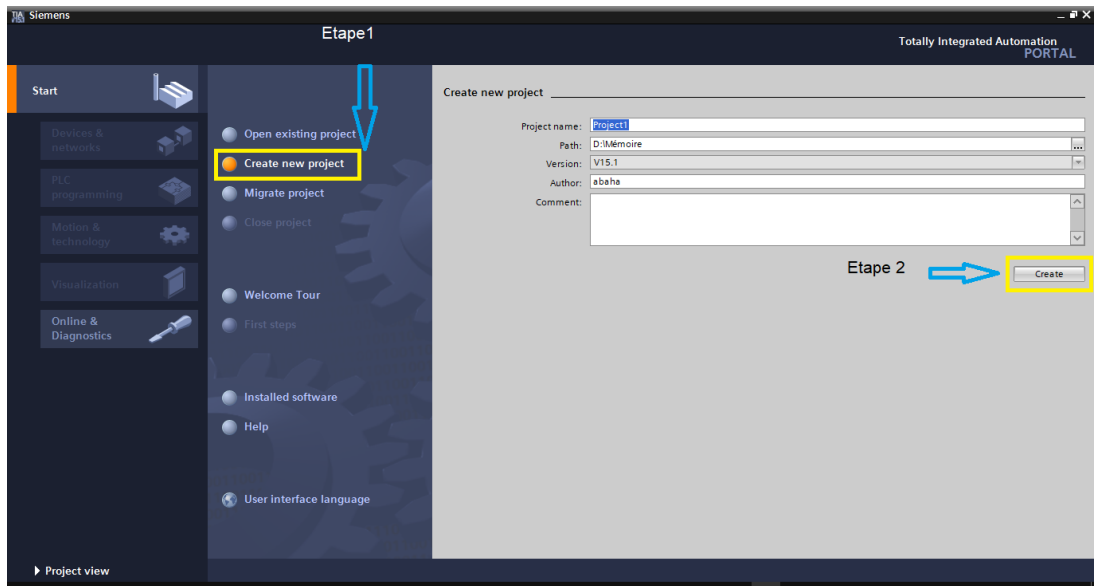


Figure III. 4 : Création d'un projet TIA Portal

« **Mise en marche** » est recommandée pour le début de la création du projet. Premièrement, nous voulons configurer un appareil par la commande « **Configure à device** »

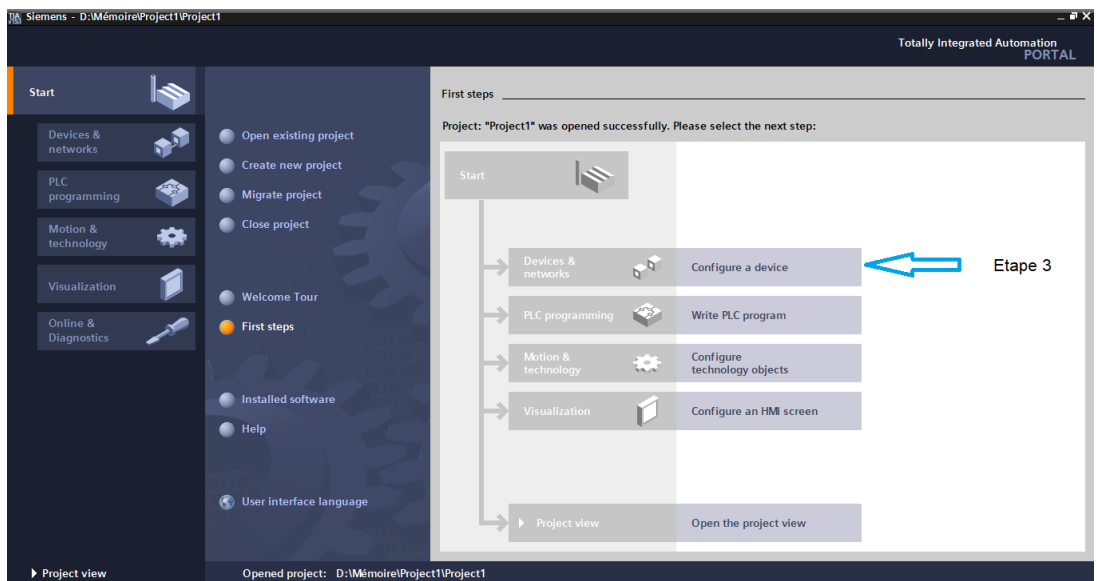


Figure III. 5: Configuration matériel

Puis « **Add new device** ». Choisissez alors dans le catalogue la commande « **Controllers** ».

« **Add new device** > **Controllers** > **SIMATIC S7-1200** > **CPU 1214C DC/DC/DC** > **6ES7 214-1AG40-0XB0** > **Add** ».

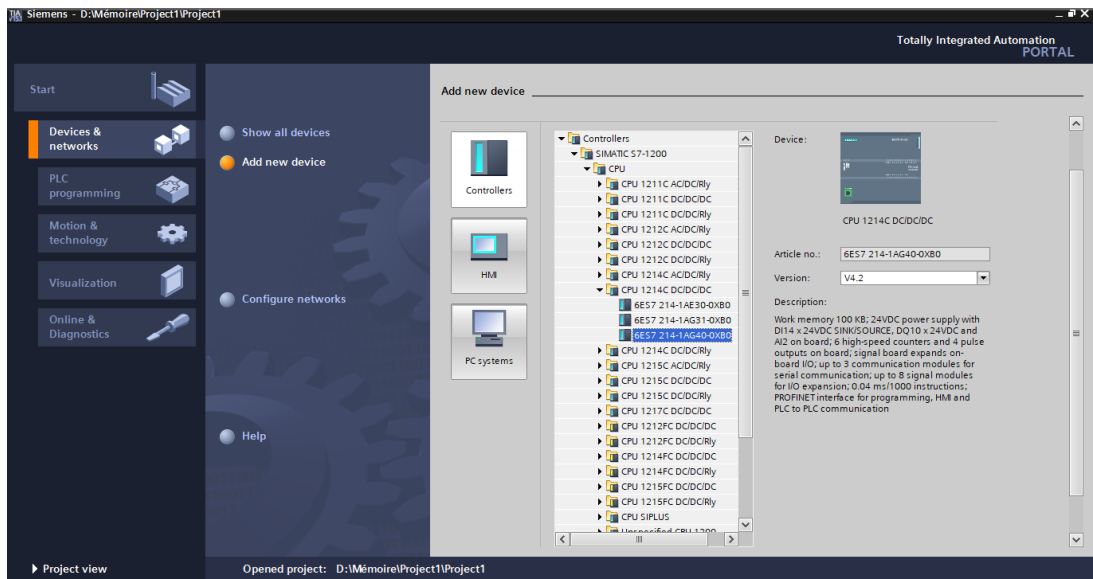


Figure III. 6: Ajout d'API

III.1.3. Configuration et paramétrage du matériel

Le logiciel bouge automatiquement vers la vue du projet avec la configuration matérielle ouverte. Ici, on peut ajouter des modules supplémentaires depuis le Catalogue du matériel (fenêtre de droite), et dans la Vue d'ensemble des appareils, les adresses d'entrée/sortie peuvent être visualisées. Dans notre cas, les entrées intégrées à la CPU ont des adresses allant de %I 0.0 à %I 1.5 (soit 14 entrées) et les sorties intégrées des adresses allant de %Q 0.0 à %Q 1.1 (soit 10 sorties).

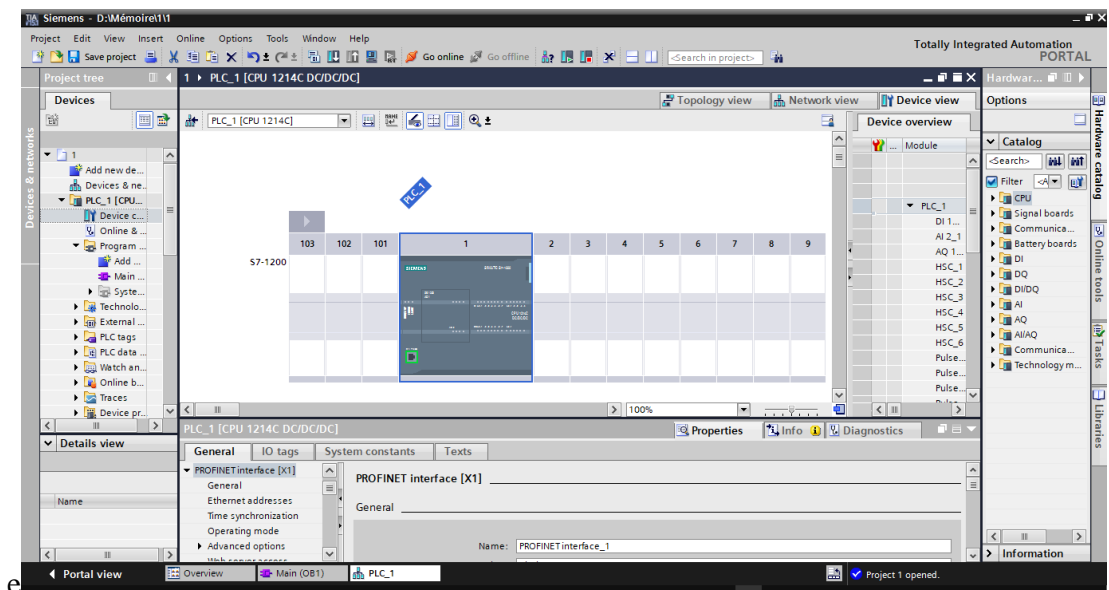


Figure III. 7: Vue de l'appareil

III.1.4. Adressage des E/S :

Pour connaître l'adressage des entrées et sorties présentés dans la configuration matériel, il faut aller dans « **IO tags** » dans le navigateur du projet.

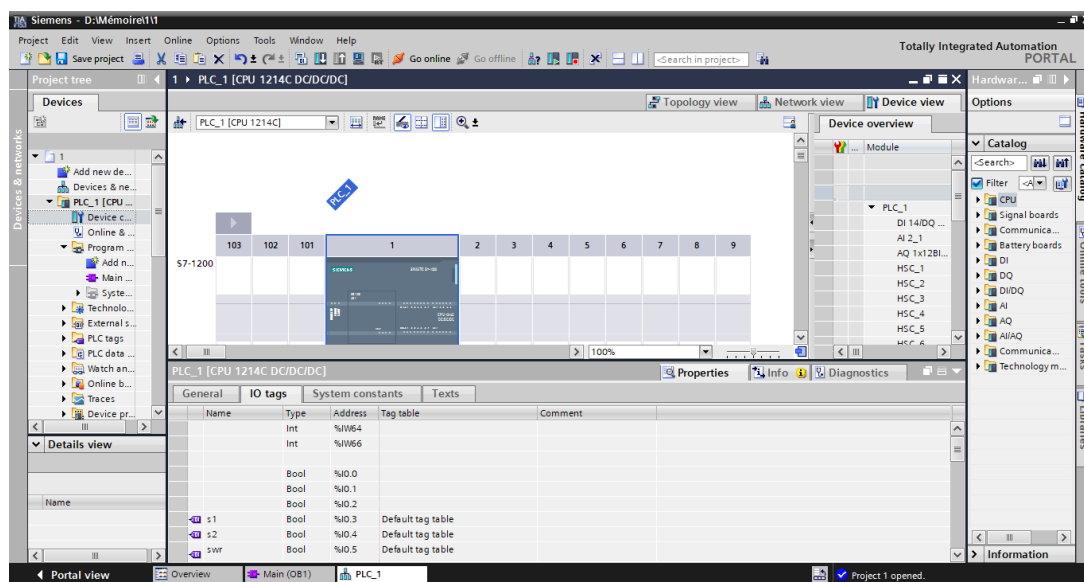


Figure III. 8: Adressage des entrées/sorties

III.1.2.1. Adresse Ethernet de la CPU :

Afin que le logiciel puisse accéder à la bonne CPU, son adresse IP et le masque de sous-réseau doivent être paramétrés « **Properties** > **General** > **PROFINET Interface** > **Ethernet addresses** > **Adresse IP**.

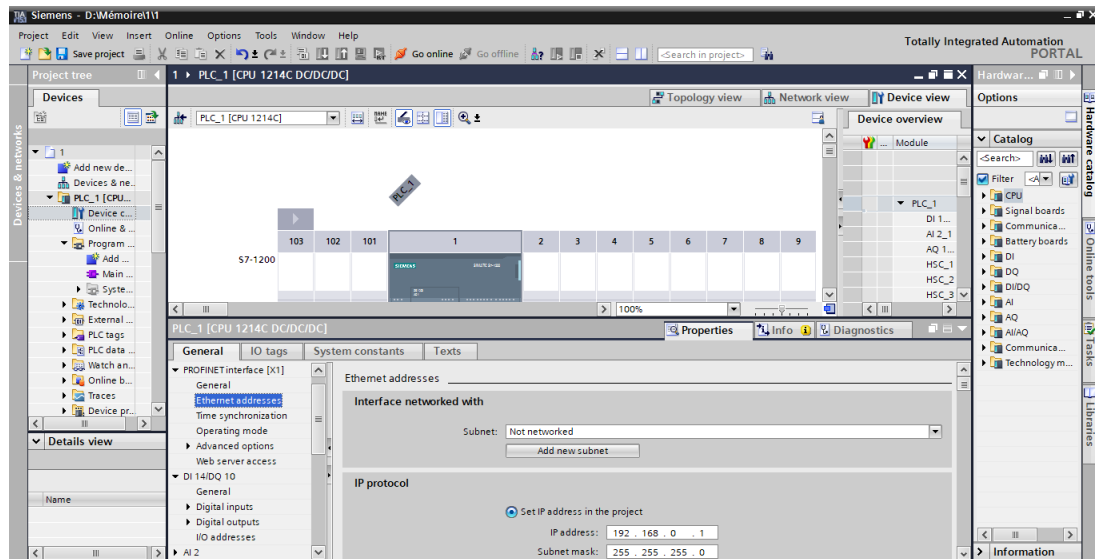


Figure III. 9: Adresse Ethernet de CPU

III.1.5. Les variables API :

III.1.5.1. Adresses symbolique et absolue :

Dans TIA Portal, toutes les variables globales (entrées, sorties, mémentos) possèdent une adresse symbolique et une adresse absolue.

- **L'adresse absolue** : représente l'identificateur d'opérande (I, Q, M,...) et son adresse et numéro de bit.
- **L'adresse symbolique** : correspond au nom que l'utilisateur a donné à la variable.

Le lien entre les adresses symbolique et absolue se fait dans la table des mnémoniques. Lors de la programmation, on peut choisir d'afficher les adresses absolues, symboliques ou encore les deux simultanément.

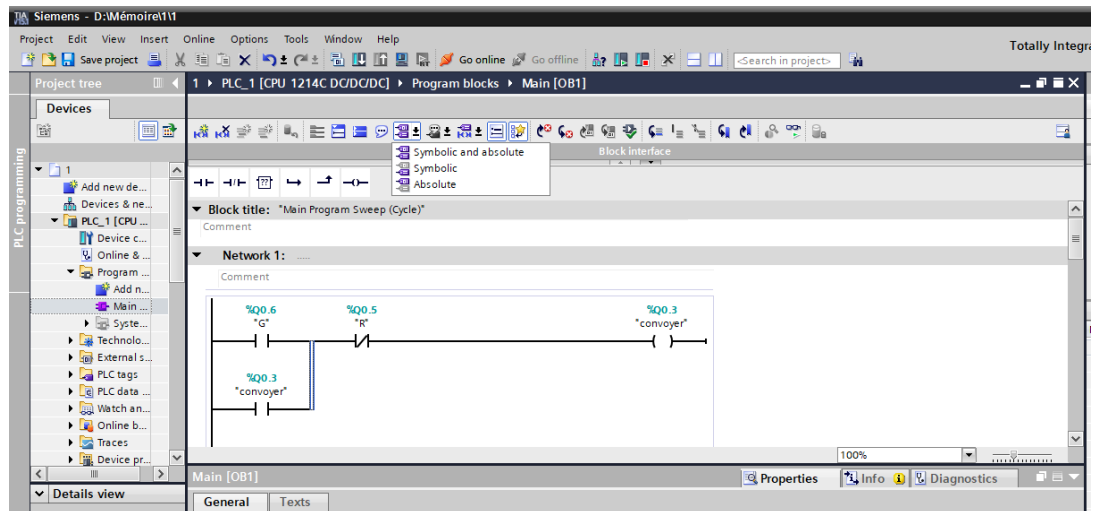


Figure III. 10: Vue l'adresse de variable

III.1.5.2. Table des mnémoniques :

C'est dans la table des mnémoniques que l'on va pouvoir déclarer toutes les variables et les constantes utilisées dans le programme.

Lorsque l'on définit une variable API, il faut définir :

- Un nom : c'est l'adressage symbolique de la variable.
- Le type de donnée : BOOL, INT,...
- L'adresse absolue : par exemple Q1.5

On peut également insérer un commentaire qui nous renseigne sur cette variable. Le commentaire peut être visible dans chaque réseau utilisant cette variable.

	Name	Data type	Address	Retain	Acces...	Writa...	Visibl...	Comment
1	S1	Bool	%I0.3		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
2	S2	Bool	%I0.4		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
3	SWR	Bool	%I0.5		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
4	SWG	Bool	%I0.6		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
5	C1	Bool	%Q0.0		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
6	C2	Bool	%Q0.1		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
7	C3	Bool	%Q0.2		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
8	E	Bool	%Q0.3		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
9	LR	Bool	%Q0.5		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
10	LG	Bool	%Q0.6		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
11	Tag	Time	%MD10		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
12	GD	Bool	%M0.0		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
13	PD	Bool	%M0.1		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
14	MD	Bool	%M0.2		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
15	Tag 1	Int	%MM8		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
16	Tag 2	Int	%MM24		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
17	Tag 3	Int	%MM26		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
18	<Add new>				<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	

Figure III. 11: L'éditeur des mnémoniques

III.1.6. L'éditeur de programme :

L'éditeur de programme est utilisé pour créer les blocs du programme. Une fois qu'on a déclaré les variables locales, on peut commencer à programmer. Pour une meilleure vue d'ensemble, on programme sur des réseaux. Un nouveau réseau peut être insérer en cliquant sur le symbole « **Insérer réseau** ». Comme le bloc lui-même, chaque réseau doit être titré. Si une longue description est nécessaire, on peut le faire dans la partie « **Commentaire** ».

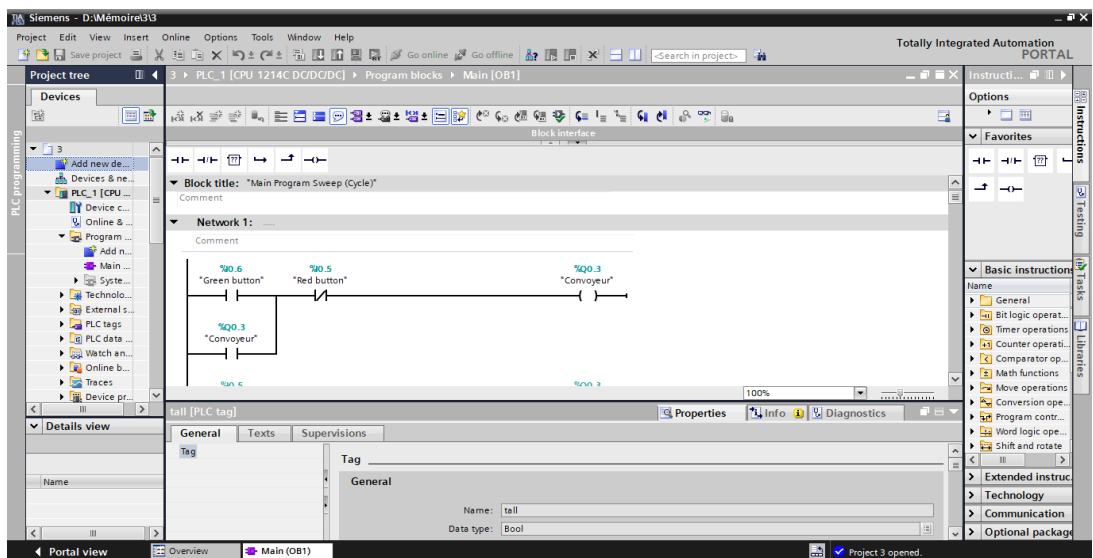


Figure III. 12: Vue de l'éditeur de programme

III.1.7. Compilation et chargement de la configuration matérielle :

Une fois la configuration matérielle est réalisée, il faut qu'elle soit compiler et charger dans l'automate. La compilation se fait à l'aide de l'icône « **compiler** » de la barre de tâche. On sélectionne l'API dans le projet puis on clique sur l'icône « **compiler** ».

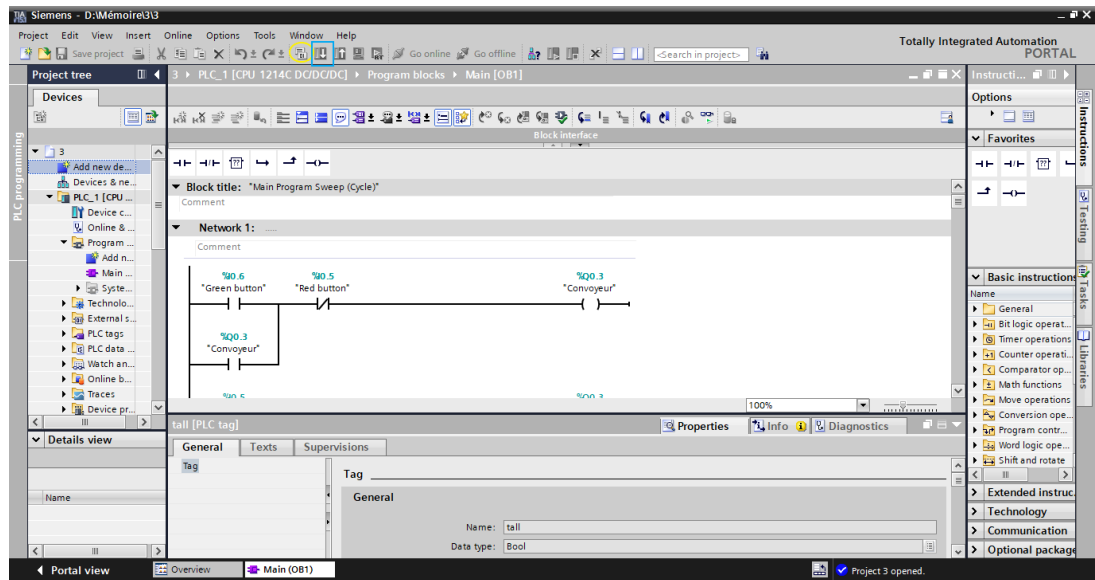


Figure III. 13: Vue de compilation et chargement

Pour charger la configuration dans l'automate, on effectue un clic sur l'icône « **Download to device** ». La fenêtre ci-dessous s'ouvre et vous devez faire le choix du mode de connexion (PN/IE, Profibus, MPI). Si vous choisissez le mode PN/IE, l'API doit posséder une adresse IP.

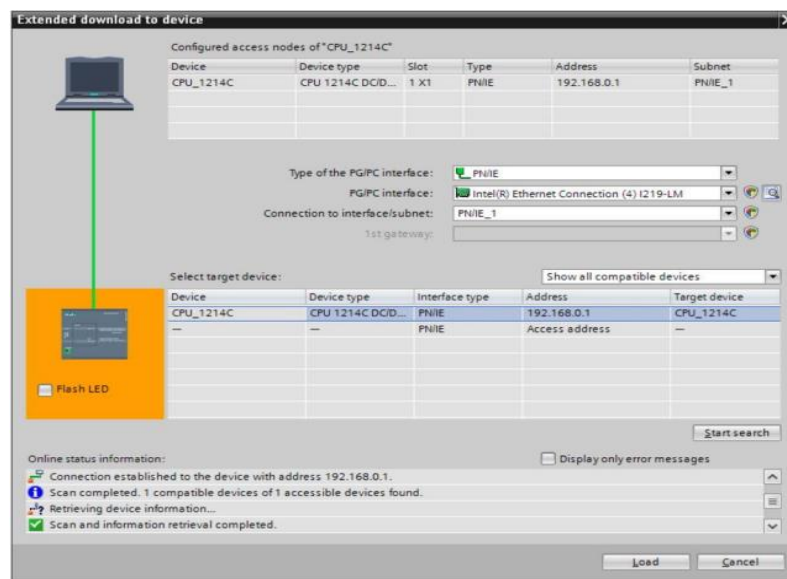


Figure III. 14: La fenêtre de choix de mode et l'interface de communication

Si le programme trouve un appareil, il figure dans la liste en bas de la fenêtre. La touche « **Clign. DEL** » permet de faire clignoter une LED sur la face avant de l'appareil afin de s'assurer que l'on est connecté à l'appareil désiré.

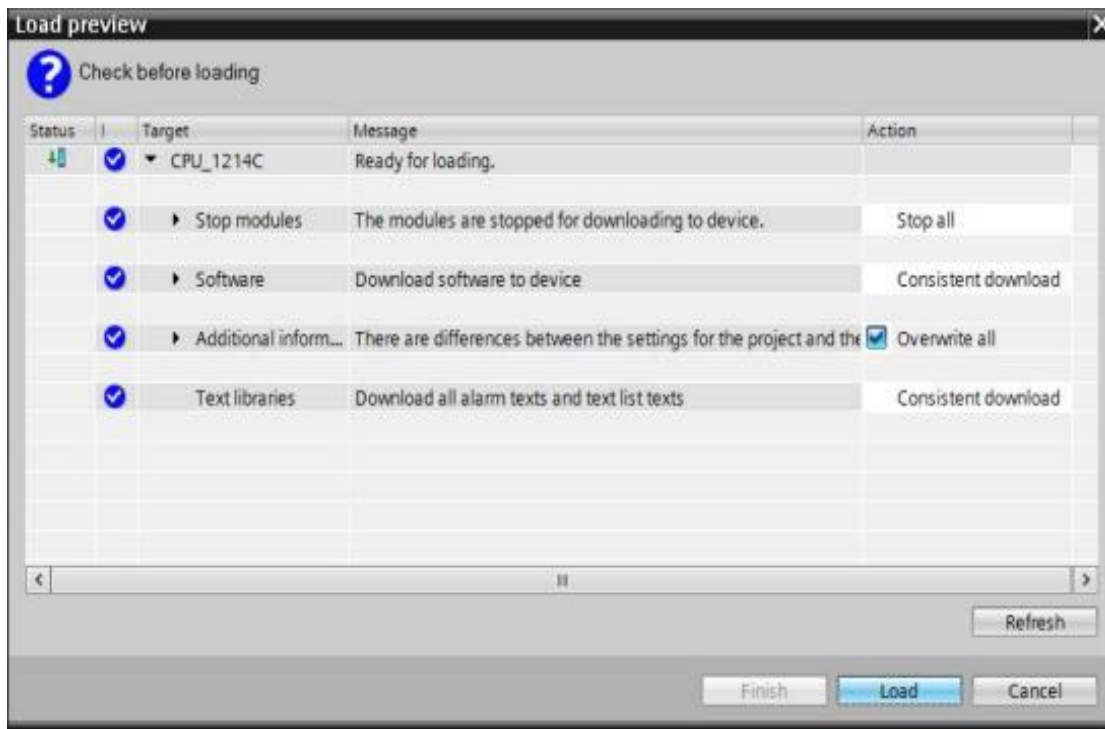


Figure III. 15: Charge de program

III.1.8. Simulation :

En absence de l'automate et des moyens pour réaliser notre processus et de tester réellement l'exécution du programme ; on peut utiliser le simulateur PLCSIM intégrée dans TIA PORTAL. Il permet de simuler un automate de la famille SIEMENS avec tous ces modules.

III.1.8.1. Le simulateur PLCSIM :

Le simulateur présente une interface simple et accessible. Pour changer l'état d'une entrée, il suffit de cocher la case correspondante. Les états des sorties changent automatiquement selon l'évolution du programme. Ceci permet de suivre l'évolution du programme en détails. La simulation va nous permettre de tester les différentes situations que peut affronter le système.



Figure III. 16: Simulateur PLCSIM

III.2. Présentation du logiciel WinCC :

III.2.1. SIMATIC WINCC :

WinCC (Windows Control Center), est le logiciel qui permet de créer une Interface Homme Machine (IHM) graphique, qui assure la visualisation et la supervision du procédé. Elle permet aussi la saisie, l’affichage et l’archivage des données, tout en facilitant les tâches de conduite et de surveillance aux exploitants. Elle offre une bonne solution de supervision, car elle met à la disposition de l’opérateur des fonctionnalités adaptées aux exigences d’une installation industrielle.

III.2.2. Wincc Runtime :

WinCC Runtime est le logiciel de visualisation de procédé dans Runtime, nous exécutons le projet en mode procédé. Au Runtime, l’opérateur peut réaliser le contrôle-commande du procédé. Les tâches suivantes sont alors exécutées :

- Communication avec les automates.
- Affichage des vues à l’écran.
- Commande du procédé.
- Archivage des données de Runtime actuelles.

III.2.3. Ecran de supervision IHM :

Avec TIA PORTAL, on peut configurer une grande gamme des écrans de supervision de SIEMENS, de différentes tailles, type de communication, type de l’écran (tactile ou clavier)

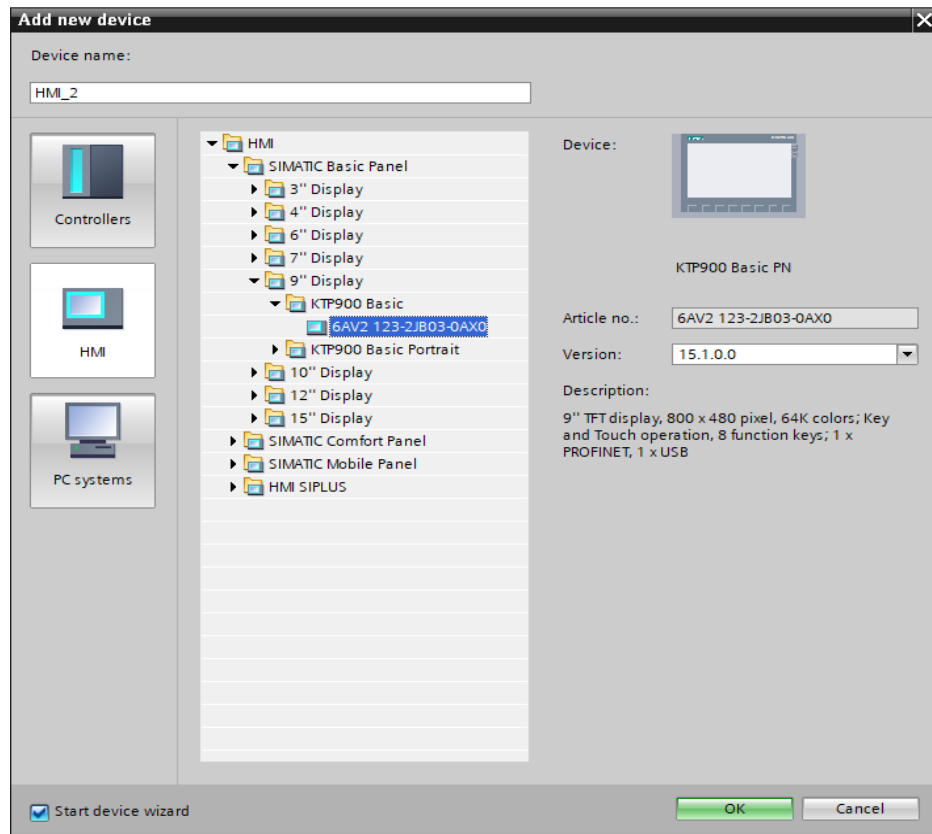


Figure III. 17: Ajoute d'IHM

III.2.4. Configuration de la connexion IHM avec l'API :

Dans l'interface de l'appareil, sélectionnez l'interface Ethernet du pupitre par un double-clic, ouvrez sous («**Propriétés** > **Général** > **Interface PROFINET** > **Adresses Ethernet** > **Adresse IP**).

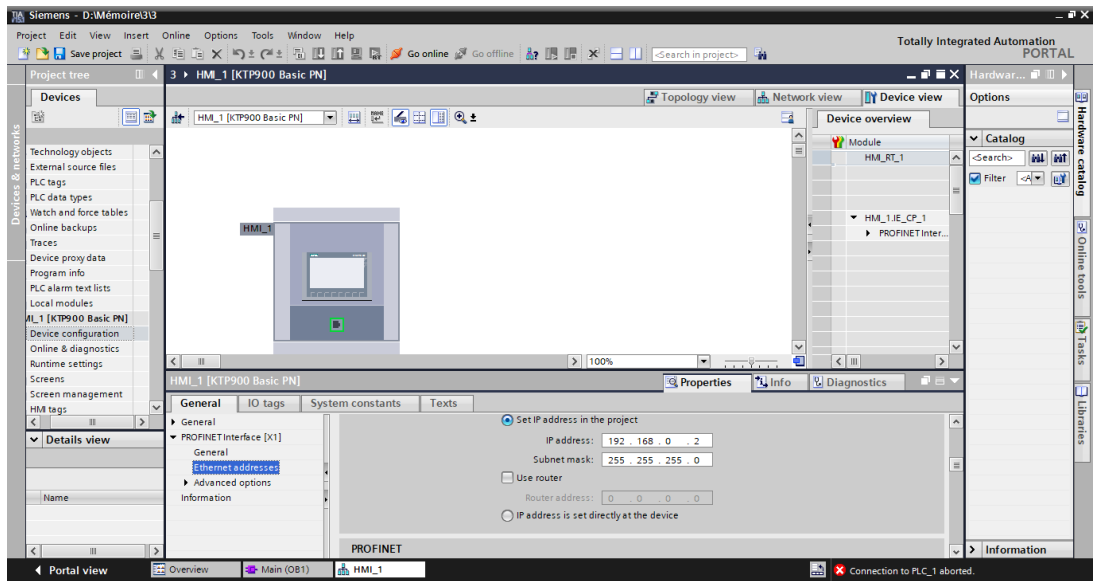


Figure III. 18: Vue de l'IHM

La liaison IHM ("HMI connection") entre la CPU et le pupitre créée au préalable dans l'assistant s'affiche

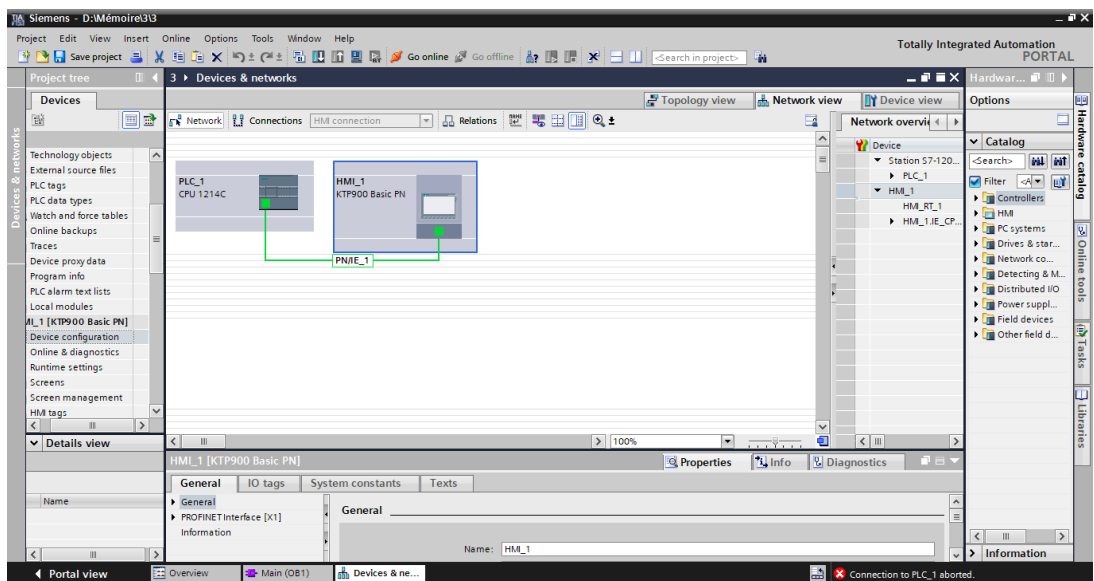


Figure III. 19: Création de la liaison de communication



III.3. Conclusion:

Dans ce chapitre, nous avons donné une description générale du logiciel « TIA Portal » et logiciel de supervision «WinCC», car ces outils sont fournis aux différentes fonctionnalités de test, telles que la simulation du programme, la visualisation ainsi la commande afin de remédier aux erreurs éventuelles et les modifications appropriées avant de procéder à la mise en œuvre dans l'automate.



Chapitre IV

*Commande et supervision du
système*

IV.1. Introduction :

Dans ce chapitre, nous avons développé des projets qui visent à identifier la taille des pièces et à les trier à l'aide de l'automate Siemens S7 1200. Le travail est organisé en trois projets, dans chaque projet on programme l'automate différemment afin de trier les pièces selon leurs tailles et les stocker dans différentes unités de stockage.

Ensuite, nous avons présenté et réalisé une interface de supervision pour compter les pièces de tailles différentes. Cette interface a été créée et simulée à l'aide du logiciel WinCC du TIA Portal.

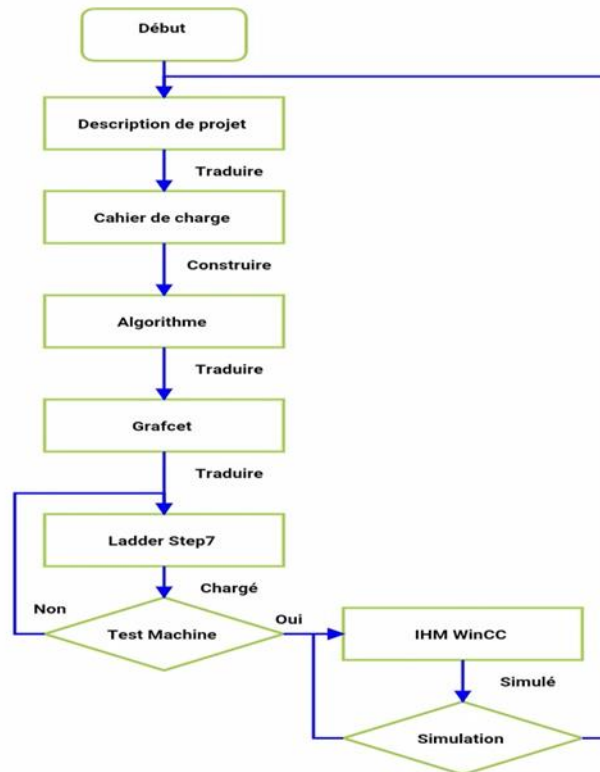


Figure IV. 1: Étapes de développement du système.

IV.2. Projet 1 : Sélection des pièces selon la hauteur

IV.2.1. But de projet :

Le but de ce projet est de contrôler la sélection de pièces selon de la hauteur. On a deux tailles différentes : pièces de court hauteur et pièces de longue hauteur.

IV.2.2. Cahier des charges :

Au départ, si on appuie sur le bouton poussoir SWG le moteur de la bande transporteuse est démarré est donc le convoyeur seras en mouvement et les trois vérins C1, C2 et C3 sont en position arrière. Par la suite, on résume les étapes suivant on ce qui suit :

- 1) Si la pièce est de courte hauteur, elle est identifiée par le capteur S1, alors cette pièce seras poussée par la sortie du vérin C1 vers la zone de stockage SB1, puis le vérin C1 revient à sa position initiale.
- 2) Si la pièce est de longue hauteur, elle est identifiée par le capteur S2 avec la valeur du capteur S1 mémorisée, alors cette pièce seras poussée par la sortie du vérin C2 vers la zone de stockage SB2, puis le vérin C2 revient à sa position initiale.

IV.2.3. Modélisation de projet :

IV.2.3.1. Table des mnémoniques :

Dans tous les programmes il faut définir la liste des variables qui vont être utilisées lors de la programmation. Pour cela, la table des variables mnémoniques suivante est défini :

Tableau 4 : Table des mnémoniques

Symbole	Description	Symbole	Description
SWG	Bouton poussoir vert	Tag(x)	Zone de mémoire
SWR	Bouton poussoir rouge	P	Pièce
E	Convoyeur active	PH	Petit hauteur
S1	Capteur 1	GH	Grande hauteur
S2	Capteur 2	PD	Petit diameter
C1	Vérin 1	MD	Moyen diameter
C2	Vérin 2	GD	Grande diameter
C3	Vérin 3	MP	Moyen diamètre et petite hauteur
LG	Lampe vert	MG	Moyen diamètre et grande hauteur
LR	Lampe rouge	GP	Grande diamètre et petite hauteur
T(x)	Temporisateur	GG	Grande diamètre et grande hauteur

IV.2.3.2. Modélisation par l'Algorithme et l'Organigramme :

L'organigramme d'un algorithme est une représentation graphique normalisée utilisée pour analyser ou décoder un problème. Dans ce qui suit, nous présentons un organigramme pour remédier au problème susvisé.

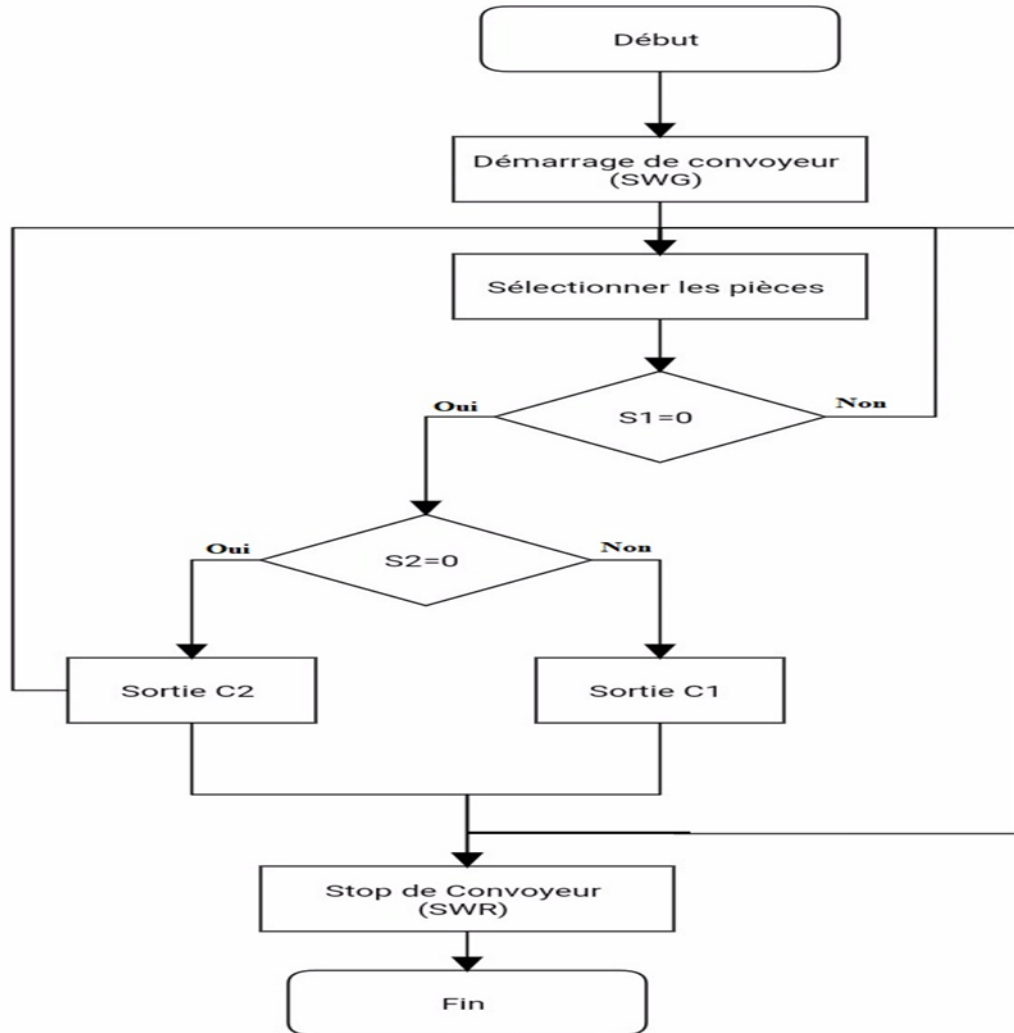


Figure IV. 2: Organigramme du projet 1.

IV.2.3.3. Élaboration de Grafcet :

Dans ce qui suit, nous allons élaborer un Grafcet traduisant l'algorithme et le cahier de charge définis précédemment.

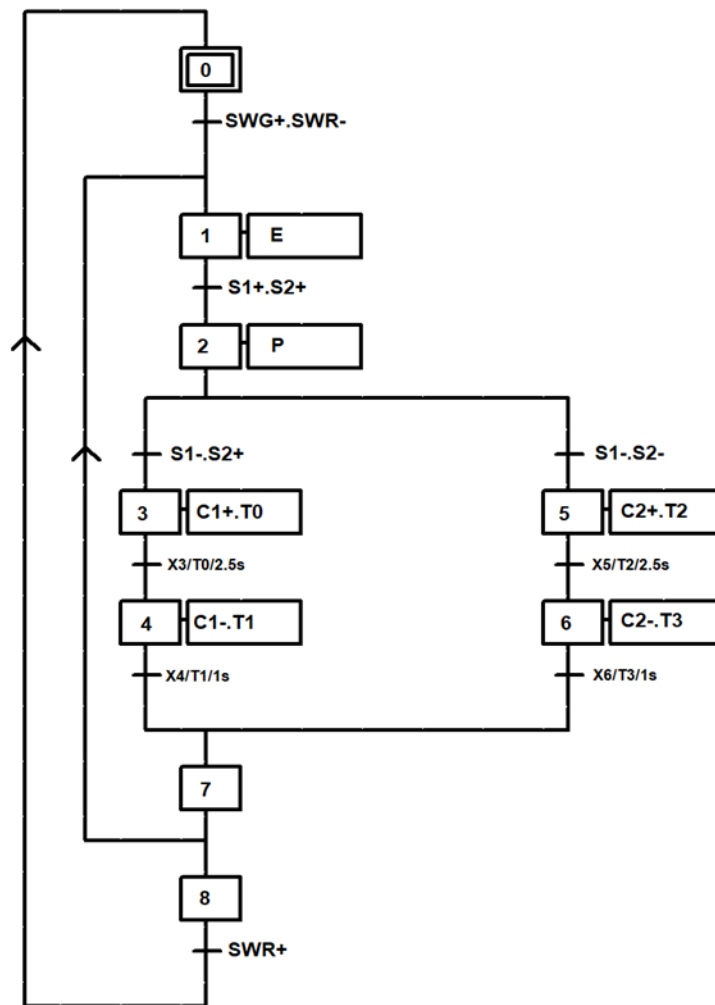


Figure IV. 3: Grafcet du projet 1

IV.2.3.4. Programmation par TIA Portal :

- Réseaux 1 : Démarrage de Convoyeur

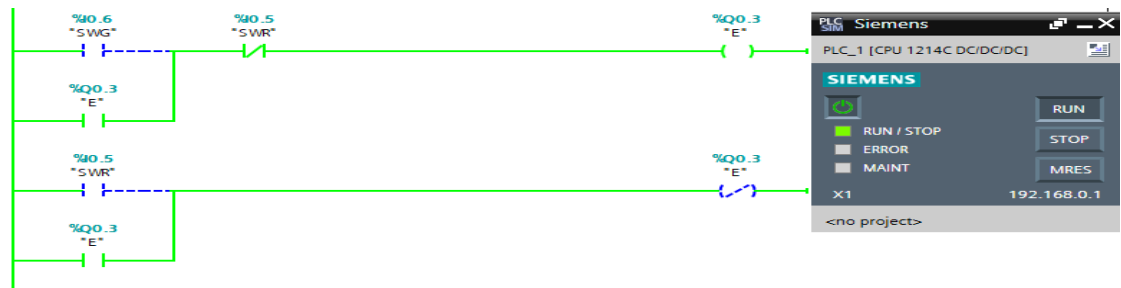


Figure IV. 4: Réseaux 1 du projet 1

- Réseaux 2 : Allumer les LED associés aux boutons poussoirs (LED vert et rouge)

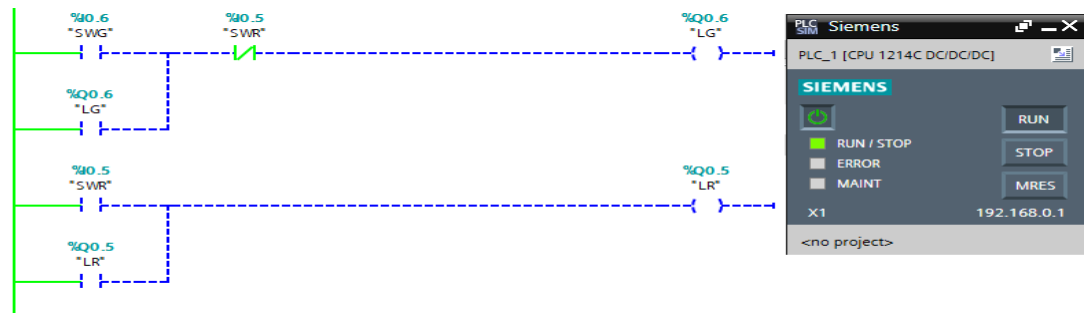


Figure IV. 5: Réseaux 2 du projet 1

- Réseaux 3 : Identification de courtes pièces

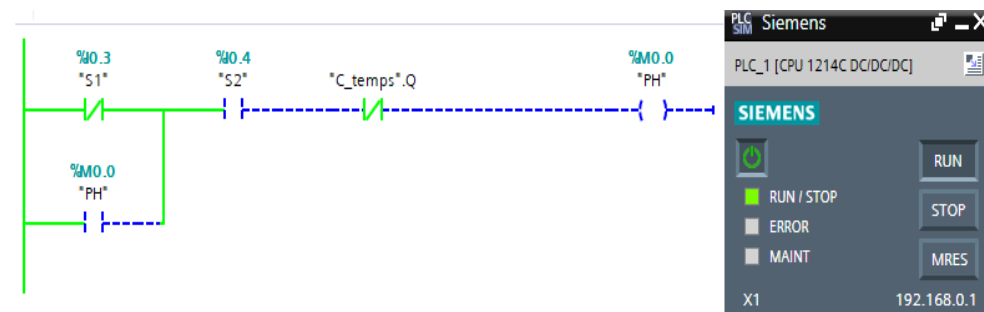


Figure IV. 6: Réseaux 3 du projet 1

- Réseaux 4 : Identification de longues pièces

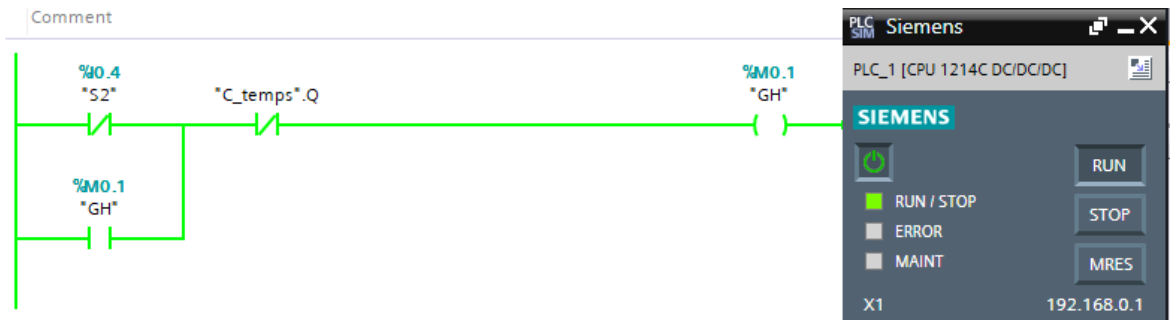


Figure IV. 7: Réseaux 4 du projet 1

- Réseaux 5 : Lancement d'une temporisation pour le tri des pièces dans SB1 et SB2



Figure IV. 8: Réseaux 5 du projet 1

- Réseaux 6 : Définition du temps de sortie de Vérin C1 et C2

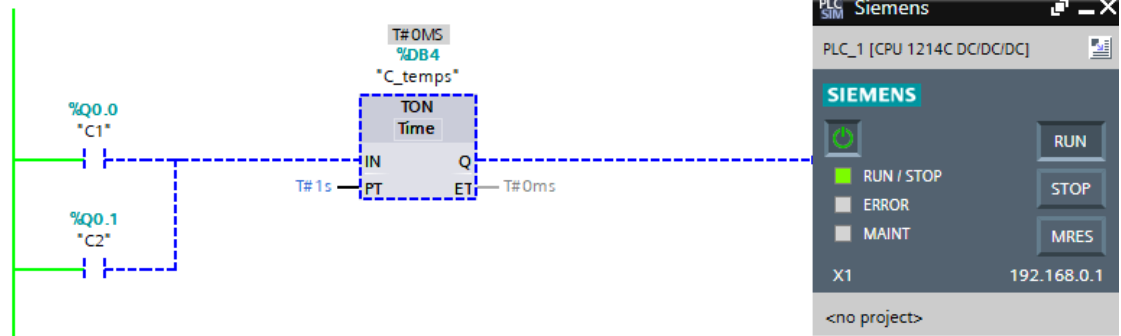


Figure IV. 9: Réseaux 6 du projet 1

Résultat d'implémentation du program

Pour consulter le fonctionnement résultant de ce programme veuillez visiter Ce [Lien](#)

IV.3. Projet 2 : Sélection des pièces selon largeur (diamètre)

IV.3.1. But de projet :

Le but de ce projet est de trier les pièces selon les différentes largeurs. On a trois largeurs différentes du diamètre : le petit diamètre, le moyen diamètre et le grand diamètre.

IV.3.2. Cahier des charges :

Au départ, si on appuie sur le bouton poussoir SWG le moteur de la bande transporteuse est démarré est donc le convoyeur seras en mouvement et les trois vérins C1, C2 et C3 sont en position arrière. Par la suite, on résume les étapes suivant on ce qui suit :

- Si la pièce est de petit diamètre, alors on doit donner l'ordre au vérin C1 pour sortir afin de la stocker dans la zone SB1.
- Si la pièce est de moyen diamètre, alors on doit donner l'ordre au vérin C2 pour sortir afin de la stocker dans la zone SB2.
- Si la pièce est de grand diamètre, alors on doit donner l'ordre au vérin C3 pour sortir afin de la stocker dans la zone SB3.

IV.3.3. Modélisation par l'Algorithme et l'Organigramme :

L'idée du programme consiste à lancer une temporisation lorsque la pièce traverse le détecteur de présence. Le temps de chaque type de pièce sera différent. Comme ça on peut

distinguer les trois types (chaque type doit avoir une différente valeur de temporisation). Par la suite, on donne l'ordre au vérin approprié pour sortir afin de stocker la pièce de ca zone.

Pour une vitesse imposé de la bande transporteuse on calcule les temporisations est on suit les étapes suivants :

- Si la pièce est de petit diamètre, elle est identifiée par sa durée de temporisation qui est inférieur ou égal à 260 ms, puis la pièce est poussée par la sortie du vérin C1 vers la zone de stockage SB1, puis le vérin C1 revient à sa position initiale.
- Si la pièce est de moyen diamètre, elle est identifiée par sa durée de temporisation qui est supérieur à 260 ms, et inférieur à 600 ms, puis la pièce est poussée par la sortie du vérin C2 vers la zone de stockage SB2, puis le vérin C2 revient à sa position initiale.
- Si la pièce est de grande diamètre, elle est identifiée par sa durée de temporisation qui est supérieur à 600 ms, puis la pièce est poussée par la sortie du vérin C3 vers la zone de stockage SB3, puis le vérin C3 revient à sa position initiale.

L'organigramme suivant résume ces étapes :

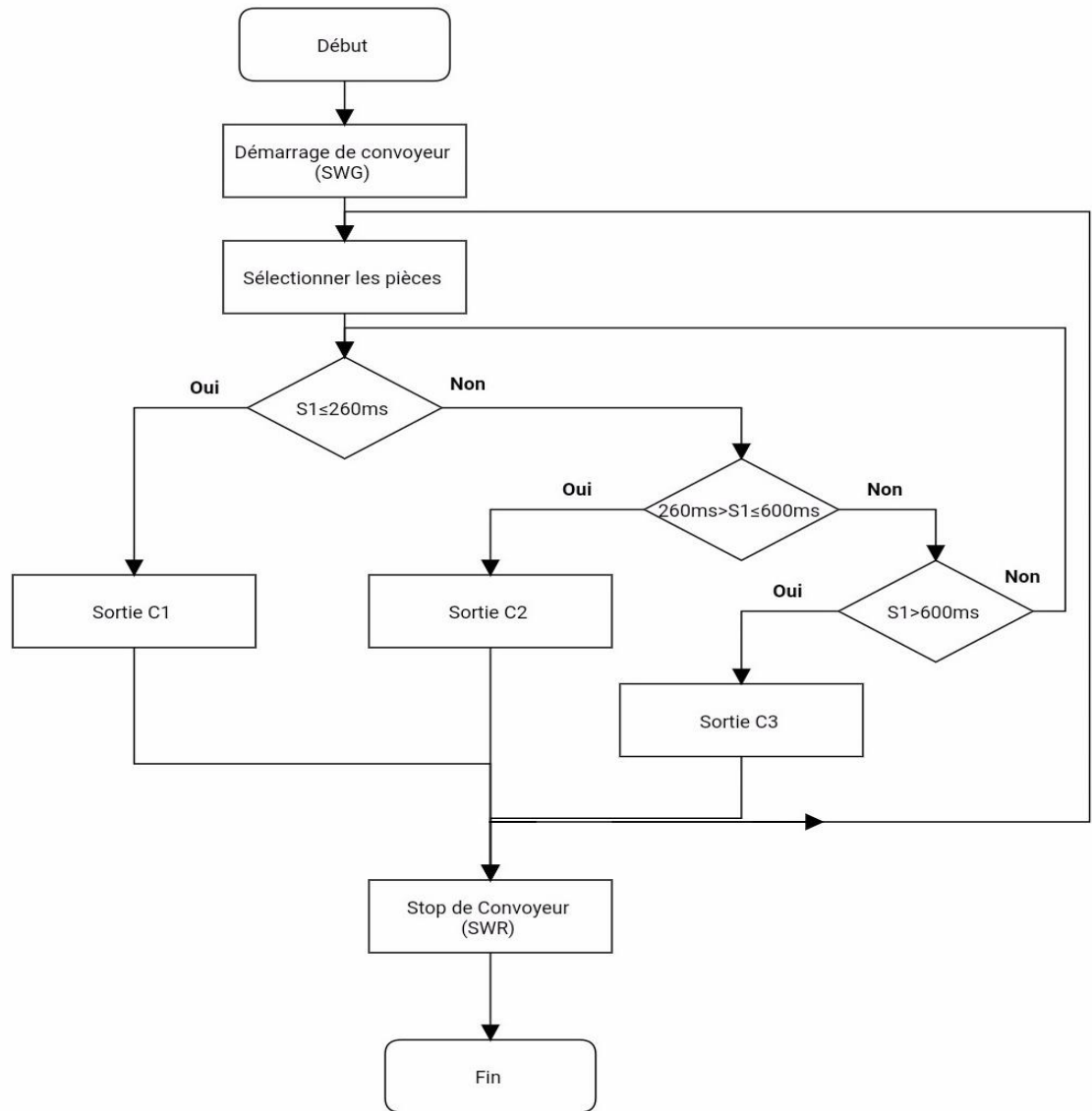


Figure IV. 10: Organigramme de projet 2.

IV.3.4. Élaboration de Grafcet :

Dans ce qui suit, nous allons élaborer les Grafcet traduisant l'algorithme et le cahier de charge définis précédemment.

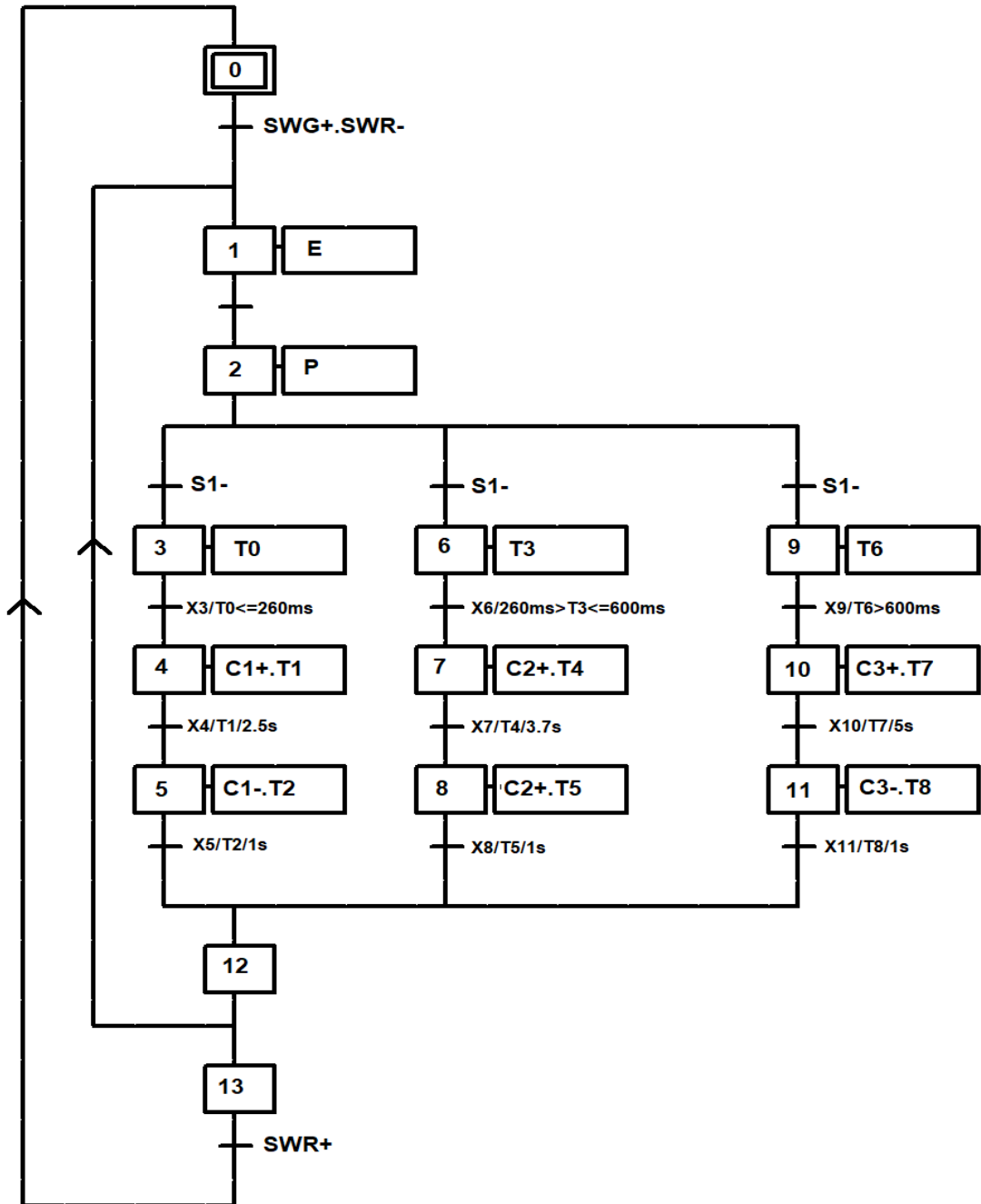


Figure IV. 11: Grafcet de sélection de pièces par différente largeur.

IV.3.5. Programmation par TIA Portal :

- Réseaux 1 : Démarrage de Convoyeur

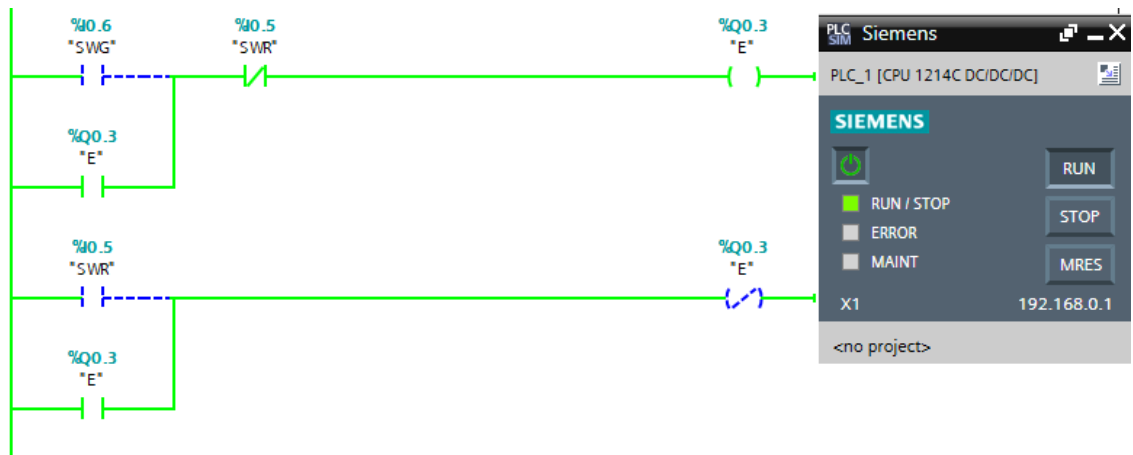


Figure IV. 12: Réseaux 1 du projet 2

- Réseaux 2 : Allumer les LEDS associés aux boutons poussoirs (LED vert et rouge)

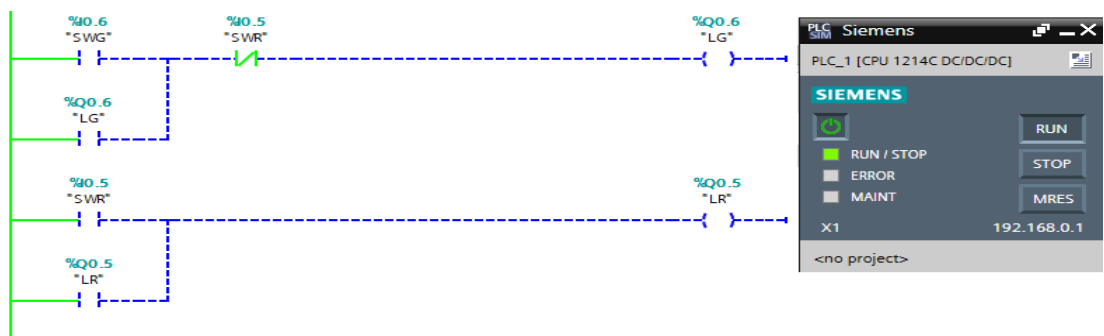


Figure IV. 13: Réseaux 2 du projet 2

- Réseaux 3 : Déterminer le nombre de pièces par un compteur

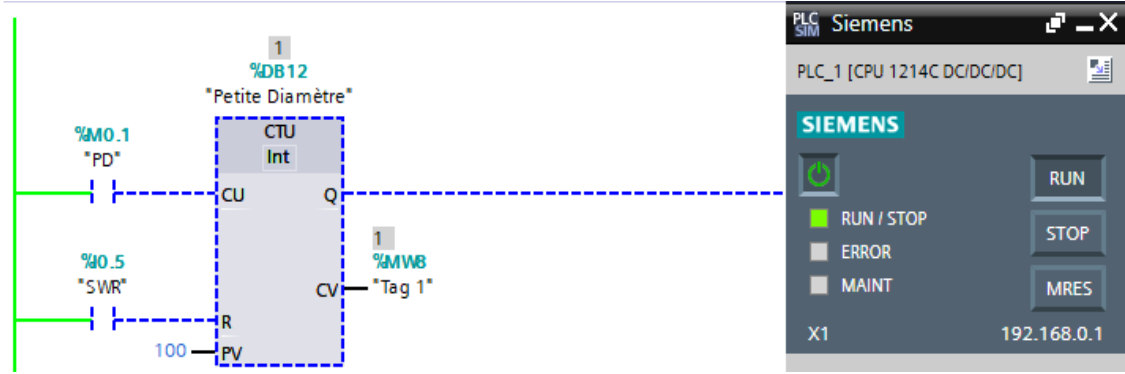


Figure IV. 14: Réseaux 3 du projet 2

- **Réseaux 4 :** Calcul de temps de petit pièce

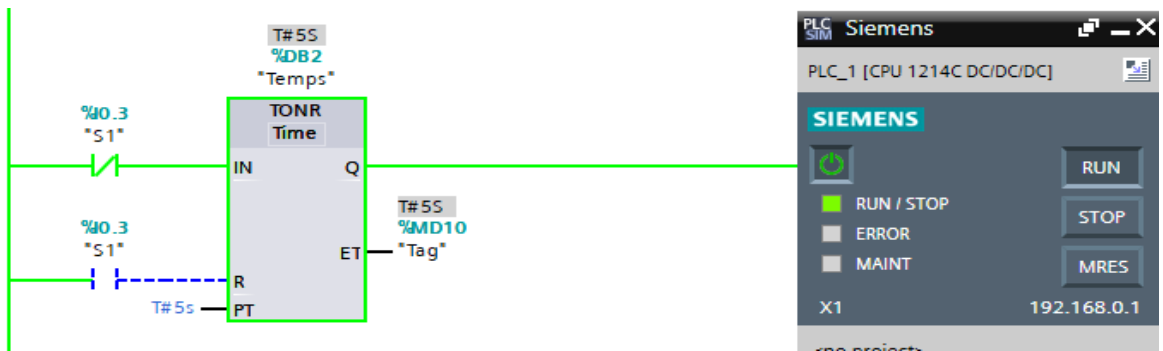


Figure IV. 15: Réseaux 4 du projet 2

- **Réseaux 5 :** Identification de courtes pièces

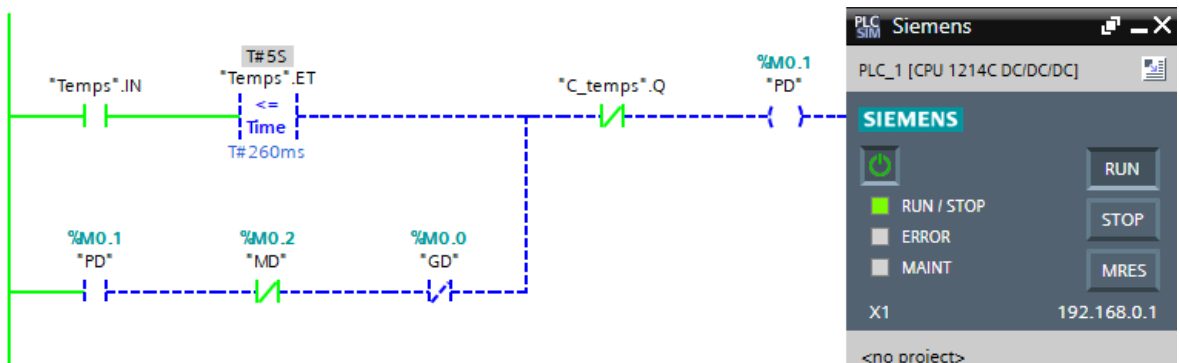


Figure IV. 16: Réseaux 5 du projet 2

- **Réseaux 6 :** Identification de Moyen pièces

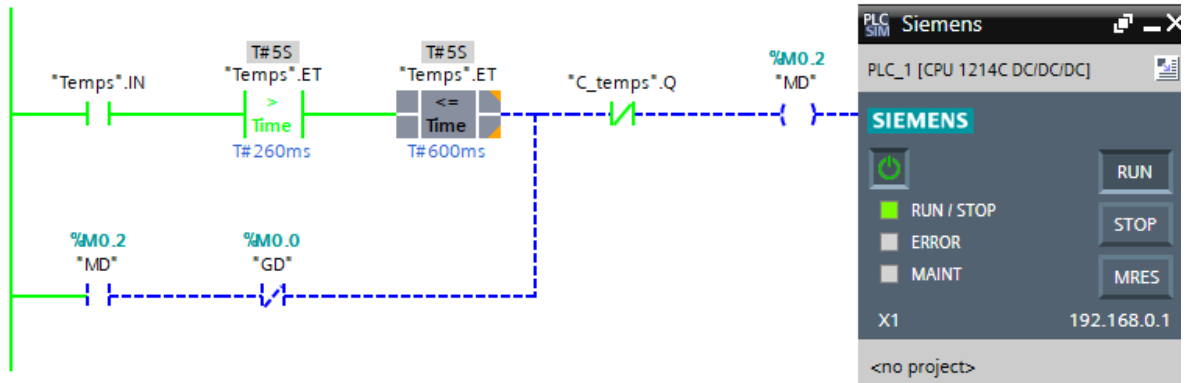


Figure IV. 17: Réseaux 6 du projet 2

- **Réseaux 7** : Identification de longues pièces

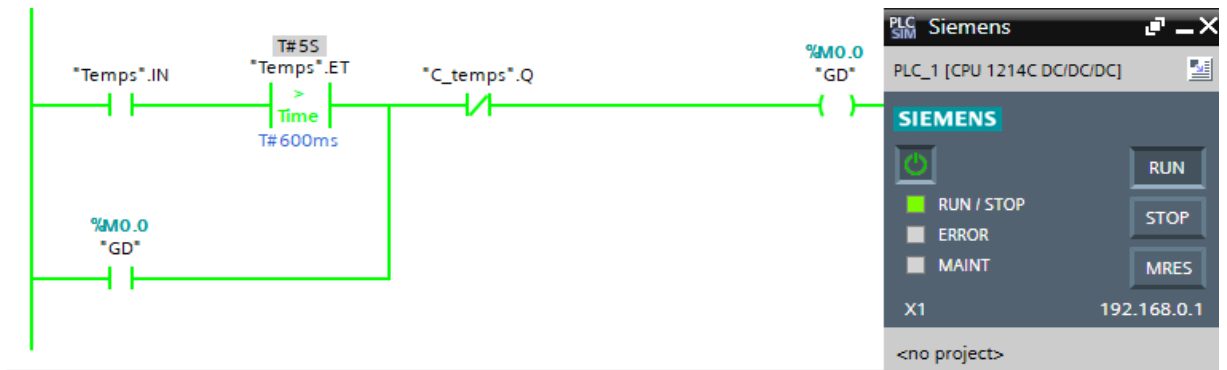


Figure IV. 18: Réseaux 7 du projet 2

- **Réseaux 8** : Définition du temps de sortie de Vérin C1 et C2 et C3

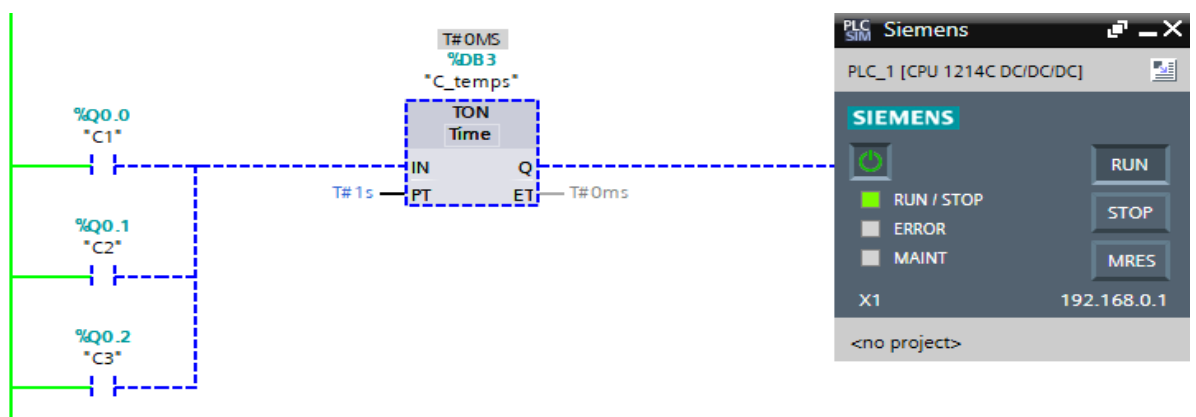


Figure IV. 19: Réseaux 8 du projet 2

- **Réseaux 9** : Lancement d'une temporisation pour le tri des pièces dans SB1et SB2 et SB3

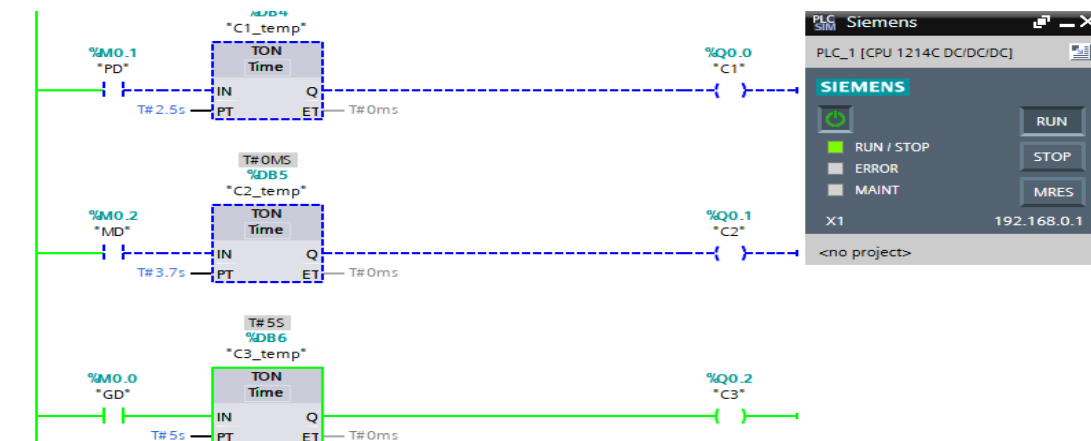


Figure IV. 20: Réseaux 9 du projet 2

Résultat d'implémentation du program

Pour consulter le fonctionnement résultant de ce programme veuillez visiter Ce [Lien](#)

IV.4. Projet 3 : Sélection des pièces selon largeur et la hauteur à la fois

IV.4.1. But de Projet :

Le but de ce projet est de trier les pièces selon les différentes largeurs et Longueurs. On a trois largeurs différentes du diamètre : le petit diamètre, le moyen diamètre et le grand diamètre, On a deux Longueur différentes : court et longue.

IV.4.2. Cahier des charges :

Au départ, si on appuie sur le bouton poussoir SWG le moteur de la bande transporteuse est démarré est donc le convoyeur seras en mouvement et les trois vérins C1, C2 et C3 sont en position arrière. Par la suite, on résume les étapes suivant on ce qui suit :

- Si la pièce de moyen diamètre et court hauteur, alors on doit donner l'ordre au vérin C1 pour sortir afin de la stocker dans la zone SB1.
- Si la pièce de moyen diamètre et longue hauteur, alors on doit donner l'ordre au vérin C2 pour sortir afin de la stocker dans la zone SB2.
- Si la pièce de grand diamètre et court hauteur, alors on doit donner l'ordre au vérin C3 pour sortir afin de la stocker dans la zone SB3.
- Si la pièce de grand diamètre et longue hauteur, puis elle est passée directement sans aucune poussée vers la zone de stockage SB4.



IV.4.3. Modélisation par l'Algorithme et l'Organigramme :

L'idée du programme consiste à lancer une temporisation lorsque la pièce traverse le détecteur de présence. Le temps de chaque type de pièce sera différent. Comme ça on peut distinguer les pièces de grand diamètre aux pièces de petit diamètre. A l'aide des capteurs S1 et S2 on distingue les pièces de court hauteur aux pièces de longue hauteur. Par la suite, on donne l'ordre au vérin approprié pour sortir afin de stocker la pièce de ca zone.

Pour une vitesse imposé de la bande transporteuse on calcule les temporisations est on suit les étapes suivants sont adoptés:

- Si la pièce est de moyen diamètre et court hauteur, elle est identifiée par sa durée de temporisation qui est inférieur ou égale à 260 ms, puis la pièce est poussée par la sortie du vérin C1 vers la zone de stockage SB1, puis le vérin C1 revient à sa position initiale.
- Si la pièce est de moyen diamètre et longue hauteur, elle est identifiée par sa durée de temporisation qui est supérieur à 260 ms, et inférieur à 600 ms, Et par le capteur S2 pour la différence de longueur, puis la pièce est poussée par la sortie du vérin C2 vers la zone de stockage SB2, puis le vérin C2 revient à sa position initiale.
- Si la pièce est de grands diamètres et court hauteur, elle est identifiée par sa durée de temporisation qui est supérieur à 600 ms, puis la pièce est poussée par la sortie du vérin C3 vers la zone de stockage SB3, puis le vérin C3 revient à sa position initiale.
- Si la pièce est de grande diamètre et longue hauteur, elle est identifiée par sa durée de temporisation qui est supérieur à 600 ms, Et par le capteur S2 pour la différence de longueur, Il est transféré directement dans la zone de stockage SB4.

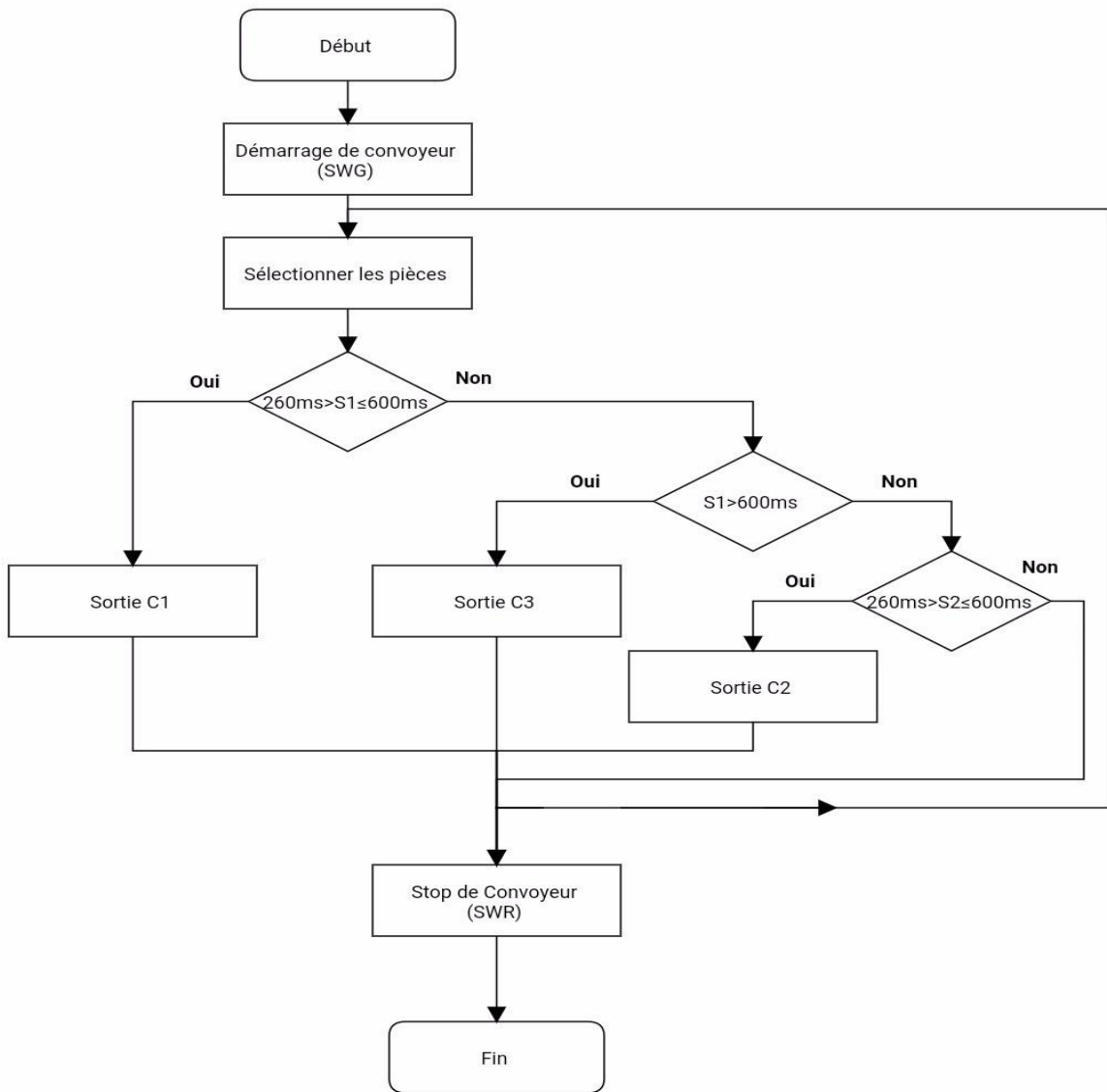


Figure IV. 21: Organigramme de projet 3.

IV.4.4.Elaboration de Grafcet:

Dans ce qui est, nous allons élaborer les Grafcet traduisant l’algorithme et le cahier de charge définis précédemment.

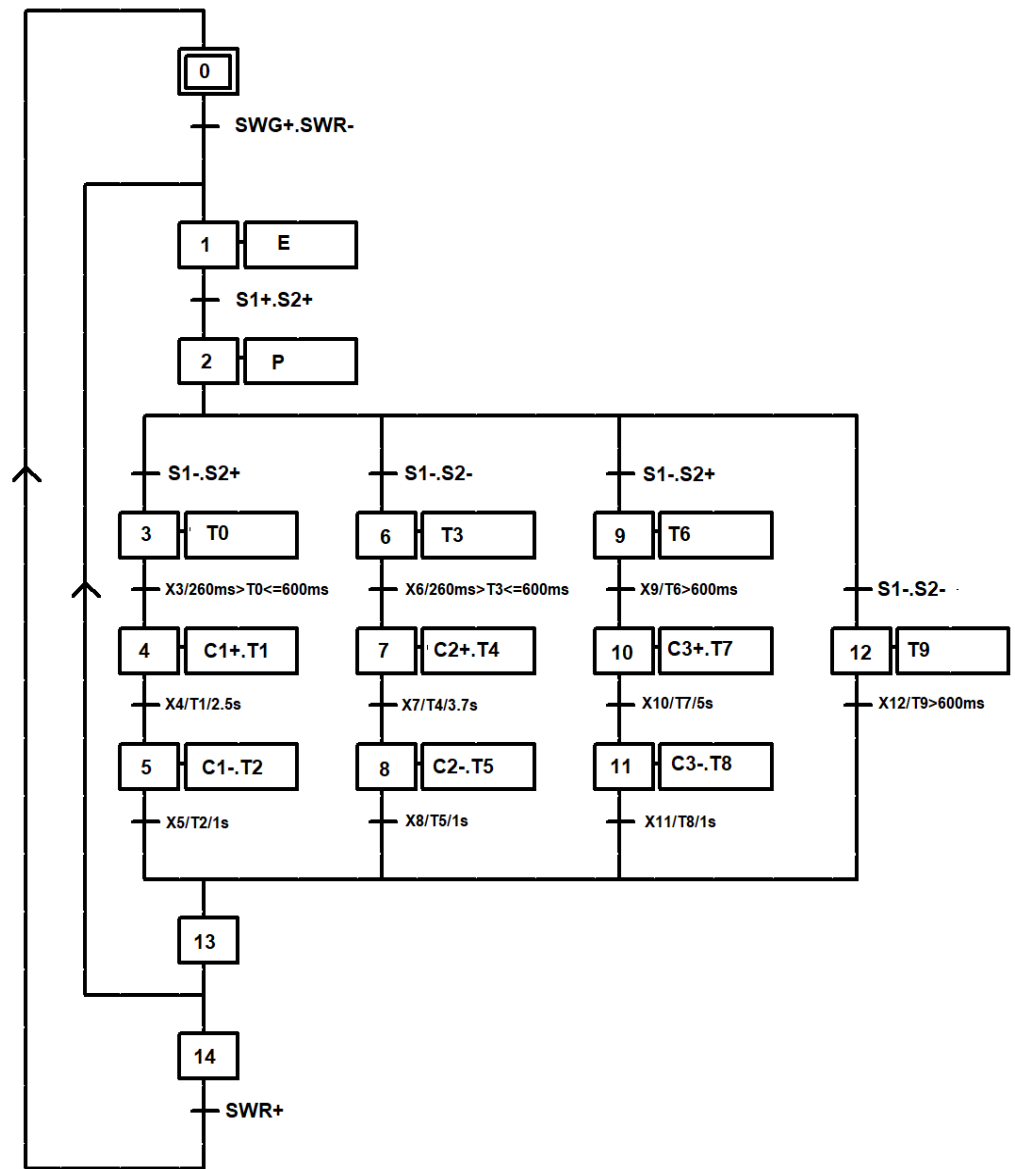


Figure IV. 22: Grafcet de sélection de pièces par différente hauteur et largeur.

IV.4.5. Programmation par TIA Portal :

- Réseaux 1 : Démarrage de Convoyeur

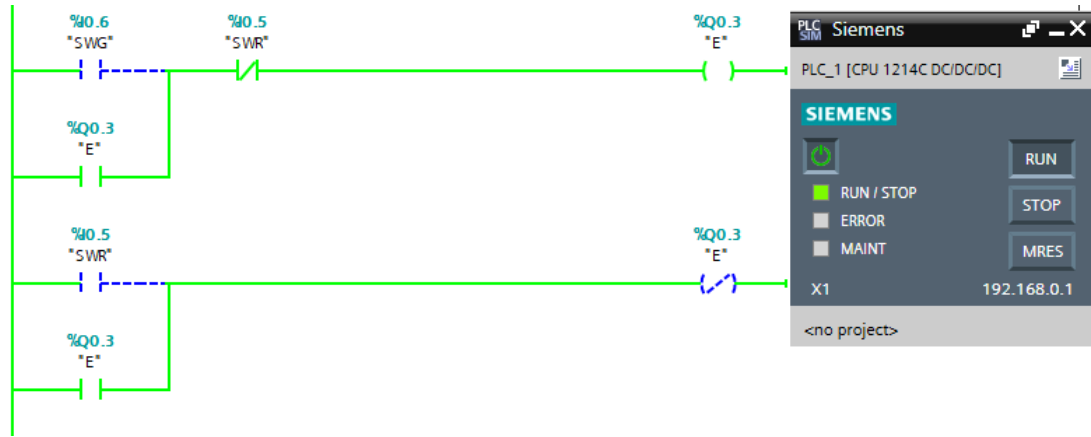


Figure IV. 23: Réseaux1 du projet 3

- Réseaux 2 : Allumer les LED associés aux boutons poussoirs (LED vert et rouge)

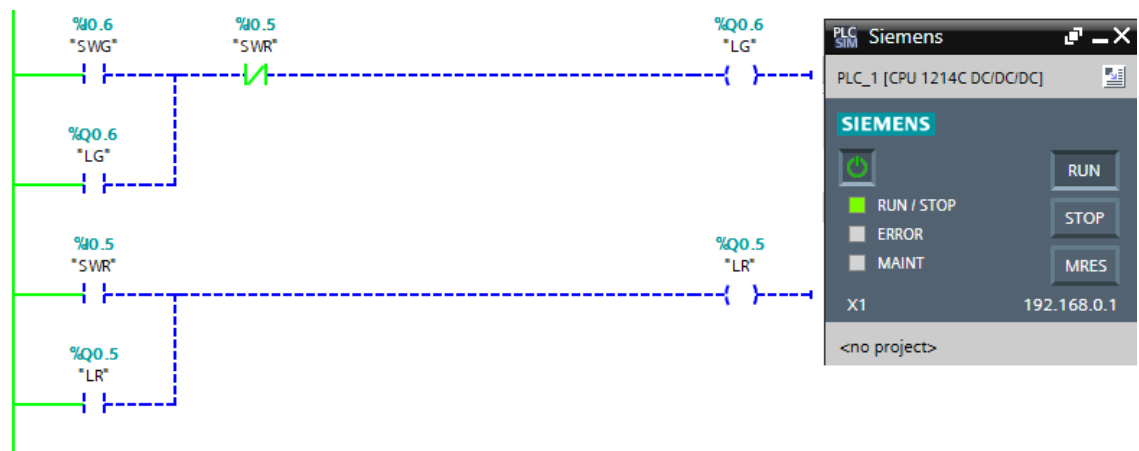


Figure IV. 24: Réseaux 2 du projet 3

- **Réseaux 3** : Calcul le temps des Pièces par Capteur S1

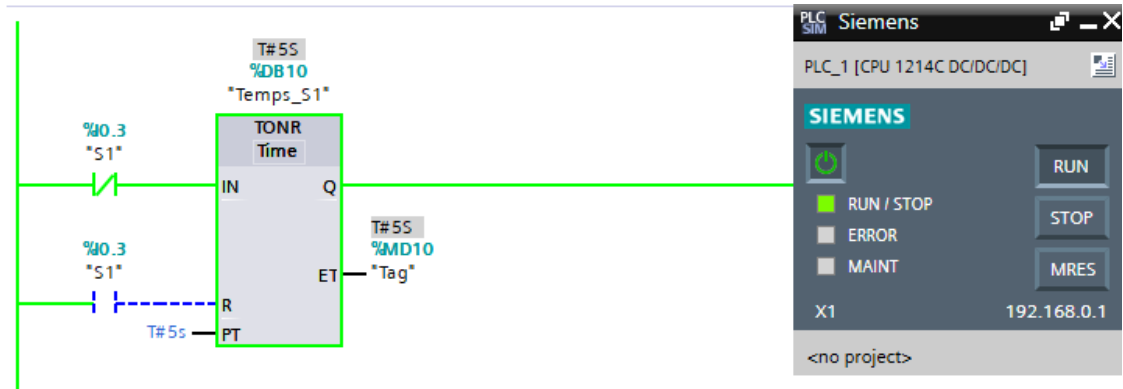


Figure IV. 25: Réseaux 3 du projet 3

- **Réseaux 4** : Calcul le temps des Pièces par Capteur S2

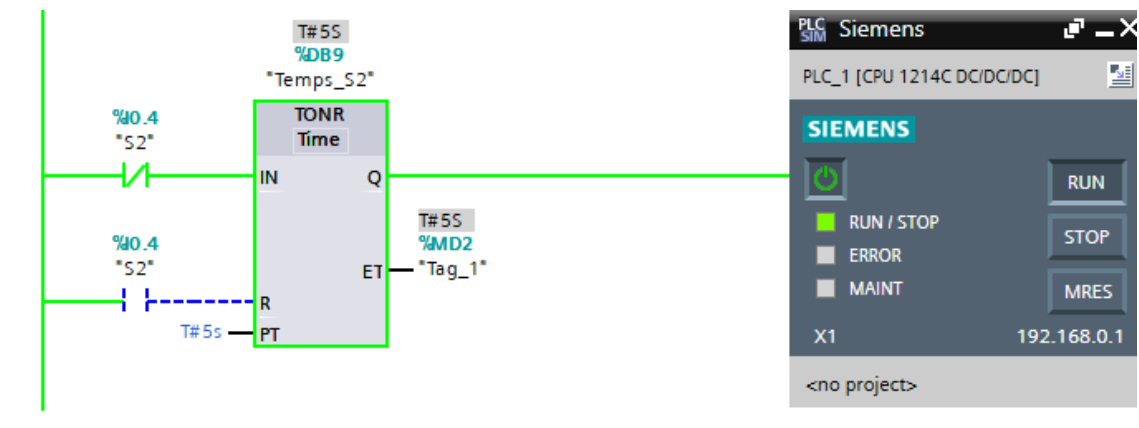


Figure IV. 26: Réseaux 4 du projet 3

- **Réseaux 5** : Identification des pièces des moyens diamètres et courts longueurs

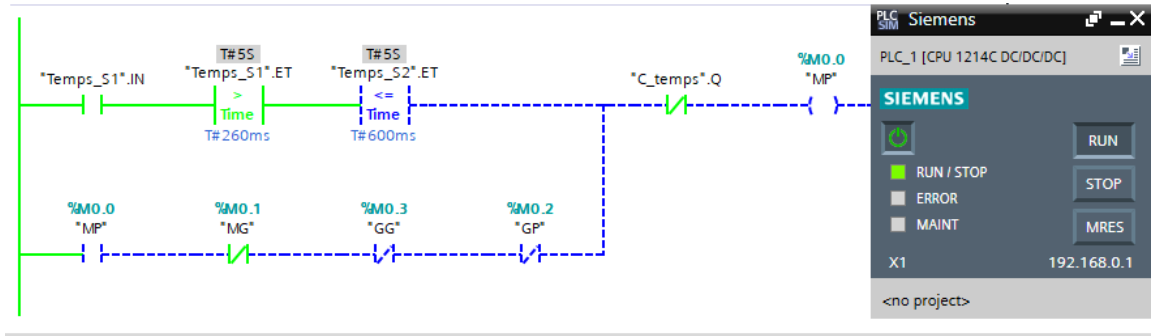


Figure IV. 27: Réseaux 5 du projet 3

- **Réseaux 6** : Identification des pièces des moyens diamètres et longues longueurs

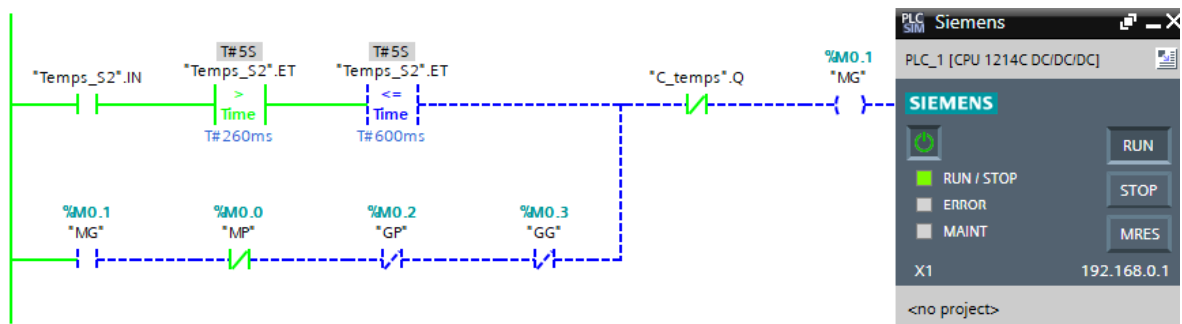


Figure IV. 28: Réseaux 6 du projet 3

- **Réseaux 7** : Identification des pièces des grands diamètres et courts longueurs

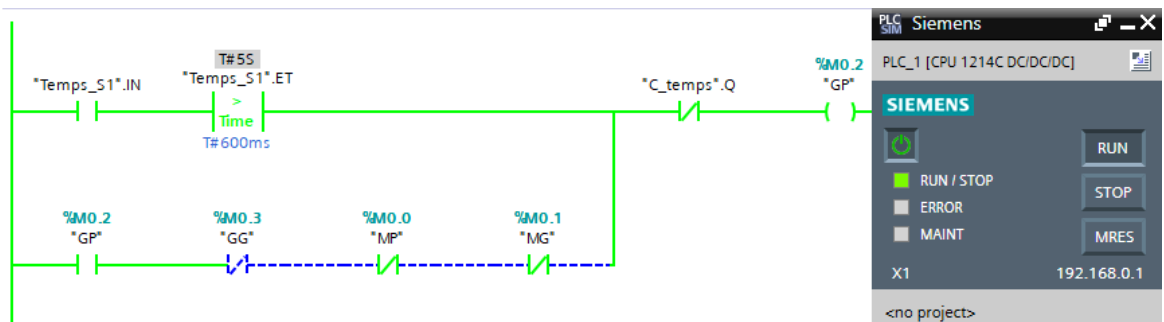


Figure IV. 29: Réseaux 7 du projet 3

- Réseaux 8 : Identification des pièces des grands diamètres et longues longueurs

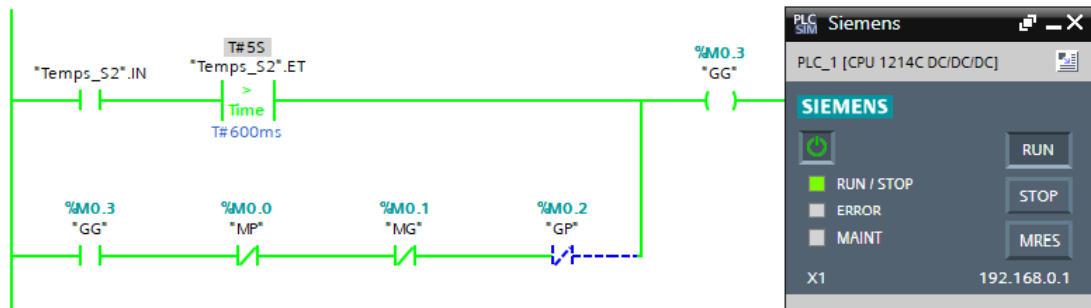


Figure IV. 30: Réseaux 8 du projet 3

- Réseaux 9 : Définition du temps de sortie de Vérin C1 et C2 et C

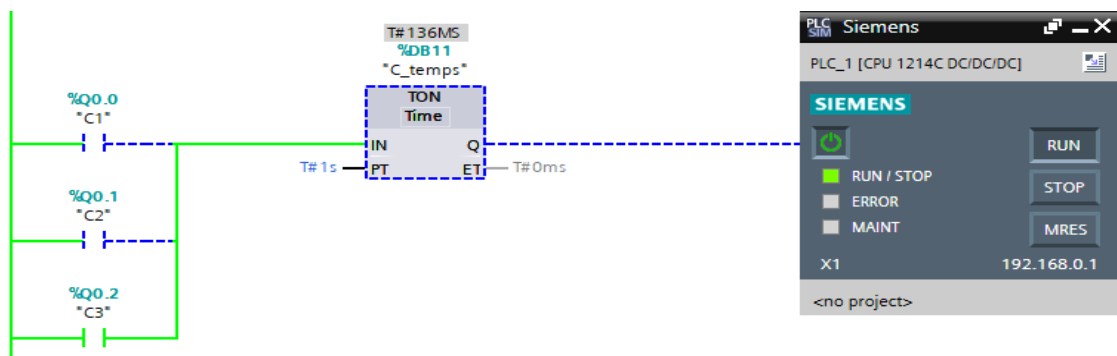


Figure IV. 31: Réseaux 9 du projet 3

- Réseaux 10 : Lancement d'une temporisation pour le tri des pièces dans SB1 et SB2 et SB3 et SB4



Figure IV. 32: Réseaux 10 du projet 3



Résultat d'implémentation du program

Pour consulter le fonctionnement résultant de ce programme veuillez visiter Ce [Lien](#)

IV.5. Projet 4 : Supervision de système sous WinCC

IV.5.1. Description de projet:

L'objectif de cette projet est la supervision et la visualisation de trois dernières projets, Pour cela nous avons créé un IHM utilisent logiciel WinCC, et programmée des compteurs pour calculer le nombre et le type de pièces passent sur le bande de transporteuse.

IV.5.2. Cahier de charge :

L'ordre de départ du cycle est donné par un bouton poussoir que l'on note SWG. Par l'appuie sur le bouton poussoir SWG le moteur est démarré puis le convoyeur [E] est avancé, et les pièces sont ensuite chargés sur le convoyeur :

- Si la pièce de petit diamètre passe, elle est affichée par compteur {Small}.
- Si la pièce de moyen diamètre passe, elle est affichée par compteur {Medium}.
- Si la pièce de grands diamètres passe, elle est affichée par compteur {Big}.

L'ordre de fin cycle est donné par bouton poussoir SWR, Si on appuie sur le bouton poussoir SWR, Le moteur est arrêté, puis le convoyeur [E] est désactivant.

Les différentes étapes de réalisation d'une application WINCC sont les suivantes :

- Choix de type d'IHM (KTP900 BASIC PN) et configuré l'interface graphique.
- Détermination de la liaison WinCC-API.
- Définition des variables de l'application (Éditeur des mnémoniques).
- Paramétrage des différents outils de commande (boutons et l'écran d'affichage) en fonction de l'état des variables.
- Exécution de Runtime et validation de l'interface.

IV.5.3. Choix de l'Interface Homme-Machine(IHM) :

Nous avons choisi le pupitre KTP 900 BASIC de la famille SIEMENS. Ce terminal d'exploitation (Interface Homme Machine) offre une facilité, une maniabilité et surtout des fonctionnalités

avancées. La mise en œuvre de cette interface de supervision nécessite le développement des programmes en utilisant des outils de programmation tel que WinCC Comfort.

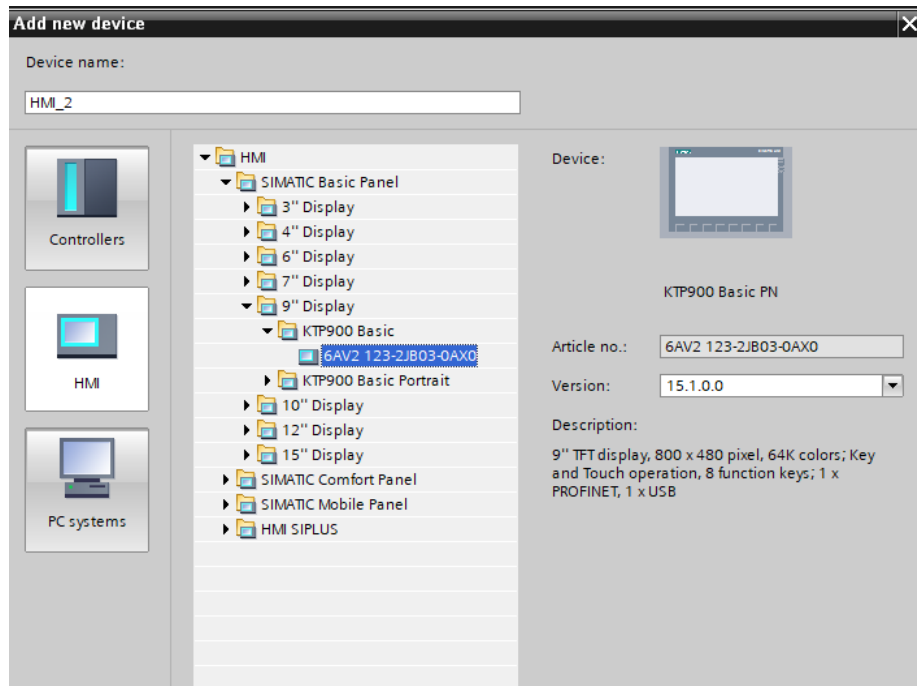


Figure IV. 33: Ajout d'IHM.

IV.5.4. Caractéristiques du SIMATIC IHM KTP900 BASIC PN :

- 1 x Ecran tactile de 9" TFT.
- Résolution 800 x 480 pixels (64K couleurs)
- 1 x RJ 45 for PROFINET/interface Industriel Ethernet.
- 1 x USB

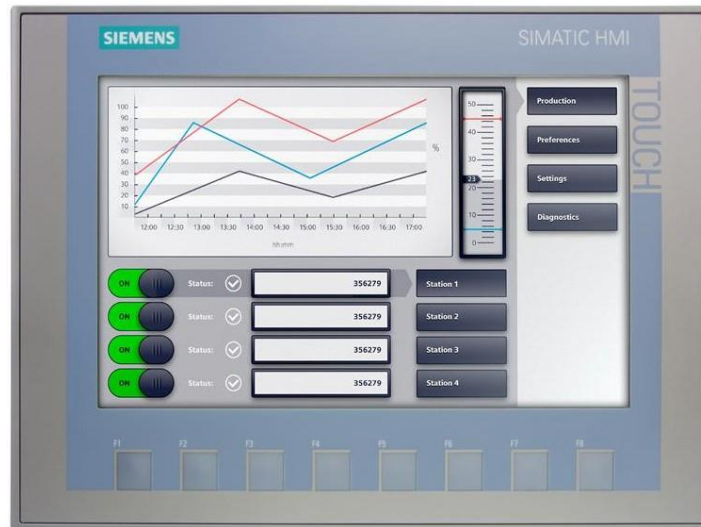


Figure IV. 34: L'interface de supervision KTP900 BASIC PN.

IV.5.5. Liaison créés par le système lors de l'intégration :

A l'ouverture de WINCC, on enregistre le projet, puis on l'intègre au projet de programmation conçu dans 'TIA portail' afin d'introduire les variables manipulées. Par la suite, on définit la liaison entre le pupitre et l'automate. La communication entre l'automate S7-1214C et l'écran de supervision « KTP900 BASIC PN » se fait via PROFINET

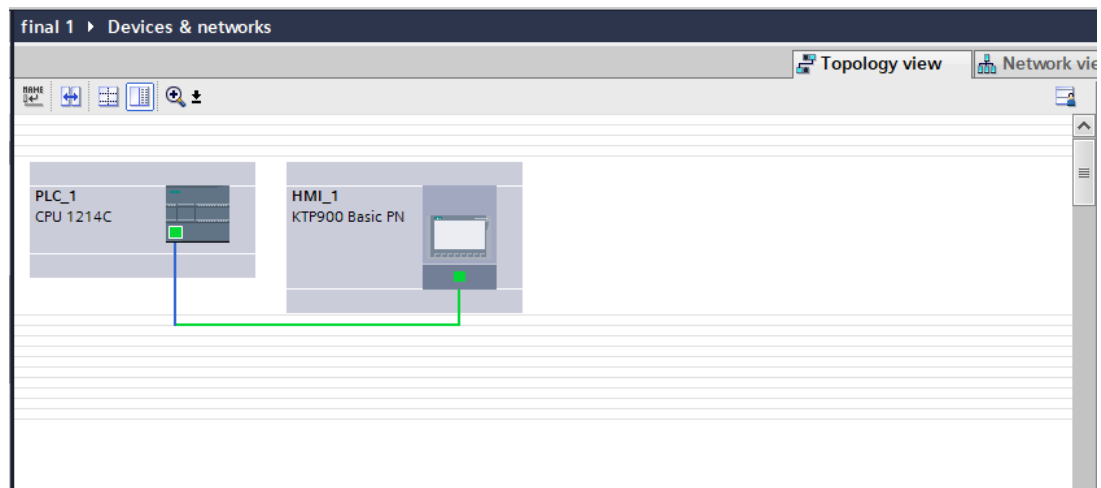


Figure IV. 35: Les paramètres de liaison d'une IHM.

IV.5.6. Editeur de mnémoniques IHM :

C'est une table qui contient les variables de l'API utilisées dans la supervision par l'HMI, et qui représentent le type de données (bool, int, reel...) la liaison matérielle (pc avec HMI). Aussi il nous permet de choisir le mode d'accès (accès symbolique ou absolu) et modifier le temps d'acquisition.

Name	Data type	Connection	PLC name	PLC tag
SWR	Bool	HMI_Connectio...	PLC_1	SWR
SWG	Bool	HMI_Connectio...	PLC_1	SWG
Compteur 1	Int	HMI_Connectio...	PLC_1	*Tag 1*
Compteur 2	Int	HMI_Connectio...	PLC_1	*Tag 2*
Compteur 3	Int	HMI_Connectio...	PLC_1	*Tag 3*
<Add new>				

Figure IV. 36: Editeur de mnémoniques.

IV.5.7. Exécution de Runtime :

L'exécution de Runtime sont les éléments principaux du validation de l'interface. Elles permettent de visualiserait de contrôler le système

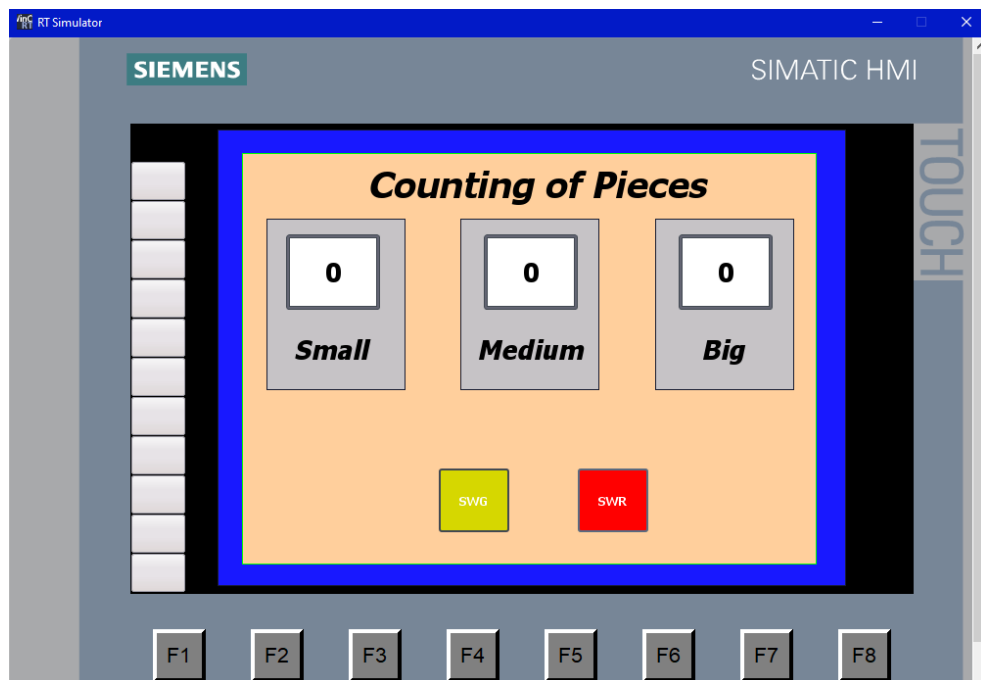


Figure IV. 37: Vue les compteurs des pièces.

La (Figure IV.37) représente Les compteurs des pièces de mon interface, elle contient :

- {SWG} : Bouton poussoir pour exécuter la machine
- {SWR} : Bouton poussoir pour éteindre la machine
- {Small} : Compteur pour calculer petit pièce
- {Medium} : Compteur pour calculer moyen pièce
- {Big} : Compteur pour calculer grande pièce

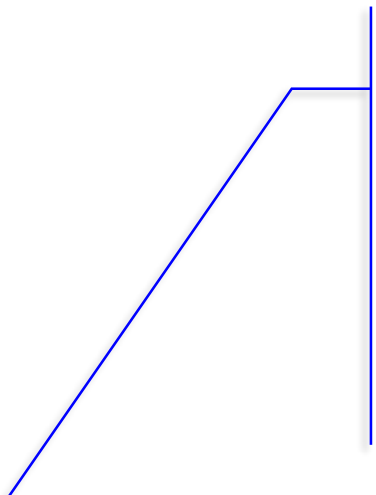
Résultat d'implémentation du program

Pour consulter le fonctionnement résultant de ce programme veuillez visiter Ce [Lien](#)

IV.6. Conclusion :

Dans ce chapitre nous avons pu choisir un API de cette projet d'étude selon le cahier de charge que nous avons établi les différents programmes et simulations sont réalisés par les différents logiciels de Siemens et développé, une interface de commande /contrôle (supervision) pour le système à l'aide du logiciel WinCC Comfort.

Les tests notamment faits sur le simulateur de WinCC Runtime Advanced ont montré que le système de supervision répond au programme de l'automate.



Conclusion
Générale



Conclusion générale :

L'ingénieur automaticien a pour mission principale de spécifier le cahier des charges en tenant compte des désirs des clients, ainsi que des besoins des responsables fabrication, maintenance et méthodes. La présentation du cahier de charges permet de bien comprendre ce que nous devons faire pour la suite de notre travail.

L'objectif de notre travail consiste à utiliser logiciel TIA Portal pour l'automatisation et la commande de système à bande transporteuse par l'automate s7-1200 de Siemens, Ensuite la supervision de système par logiciel WinCC. Pour cela, Nous avons procédé étapes par étapes.

Après avoir présenté la description de système à bande transporteuse, nous avons réalisé une étude sur le système et l'automate s7-1200 et ceci afin de mieux comprendre son fonctionnement.

Ensuit, On a proposé la gestion du prototype de système à bande transporteuse réalisé au sein de laboratoire de l'automatique par l'automate s7-1200, la solution que nous avons proposé pour l'automatisation de prototype était basée sur les problèmes à la sélection des pièces selon largeur et la hauteur, nous avons développé trois projets pour résoudre ce problèmes.

Enfin, nous avons réalisé une interface qui permet de compter les pièces et lancer ou arrêter le système. Cette interface est réalisé par le logiciel WinCC du siemens.

Références bibliographiques

- [1]. GONZAGA, A. (2004). Les automates programmables industriels. PDF téléchargé du www.Geea.Org.
- [2]. HADJAÏSSA, A. (2019). Automates Programmables Industriels.
- [3]. (HAMMOUMI).Automatismes logiques Industriels.PDF
- [4]. Mémoire de Master " Les langages de programmation de l'automate programmable Industriel (application pilotage d'un ascenseur)", UNIVERSITE d'ADRAR, Mai 2017.
- [5].Initiation-1-TIA-Portal-MS1.doc (Programmation de l'API SIMATIC S7-1200 avec TIA Portal VX).PDF
- [6].CHAOUI Mohamed. (2015-2016). Automatisation et supervision d'un ascenseur.PDF
- [7]. Automate programmable S7-1200 (A5E02486682-05). (11/2011).PDF
- [8]. Hardware Configuration S7-1200.FR. (2018).PDF
- [9]. SIMATIC S7-1200.PDF
- [10] cloud.ljcreate.com [Student Access: DRL1-4692(290-00: Industrial Control Trainer - Siemens (GB))]
- [11] Article moteur à courant continu 25/09/2007 site : <https://energiepluslesite.be/techniques/ascenseurs7/moteur-a-courant-continu/#Generalite>
- [12] Mémoire de Master « Vérification et contrôle des fonctions des Bandes Transporteuses en Caoutchouc et en PVC utilisées dans l'industrie » UNIVERSITE BADJI MOKHTAR ANNABA, 2015/2016
- [13] LES CAPTEURS EN INSTRUMENTATION INDUSTRIELLE , Georges Asch Bernard Poussery, 8e édition.
- [14] les composants pneumatiques coursbis, Author : laurent, 4/28/2006
- [15] Les Capteurs, Author :MICHEL 12/29/2014
- [16] Infor SunSystems 6.3.x, Date de publication : 16 juillet 2018, Code document : ss_6.3.x_ssolh__fr-fr_dhtmlhelp6.
- [17] MODULE A3 Initiation á la programmation d'API avec STEP 7 , Edition : 05/2004 ; fr : 05/2005