

الجمهورية الجزائرية الديمقراطية الشعبية
République Algérienne Démocratique et Populaire
وزارة التعليم العالي والبحث العلمي
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique

المركز الجامعي لعين تموشنت
Centre Universitaire de Ain Témouchent - BELHADJ Bouchaib
Institut de Technologie
Département de Génie Electrique



Projet de Fin d'Etudes

Pour l'obtention du diplôme de Master en :

Domaine : Sciences & Technologies

Filière : Electronique

Spécialité : Instrumentation

Thème

Etude et Simulation d'une Station de Lavage

Présenté Par :

Mr HOURIA Ibrahim

Devant le jury composé de :

Dr AISSOU Massinissa	M.C.A	C.U.A.T	Président
Dr FLITTI Mohammed	M.C.A	C.U.A.T	Examineur
Dr ZEBENTOUT A. Djawad	M.C.B	C.U.A.T	Encadrant

Sommaire

<i>Liste des tableaux</i>	<i>v</i>
<i>Liste des figures</i>	<i>vi</i>
<i>Listes des abréviations</i>	<i>viii</i>
 <i>Introduction Générale</i>	 <i>1</i>

Chapitre I

Généralité sur les Automatismes

I.1 Introduction.....	2
I.2 Système automatisé et automatisme	2
I.3 Avantages et inconvénients d'un système automatisé	3
I.4 Classification des systèmes automatique :	4
I.4.1 Les systèmes de grande diffusion :	4
I.4.2 Les systèmes de diffusion limitée :	4
I.5 Structure d'un système automatisé	5
I.5.1 Partie opérative	6
I.5.2 Partie commande.....	9
I.6 Communication dans un système automatisé	10
I.6.1 Le niveau 0 :	11
I.6.2 Le niveau 1 :	11
I.6.3 Le niveau 2 :	12
I.7 Lavage automobile.....	12
I.7.1 Évolution du monde de lavage.....	12
I.7.2 Lavage manuel	13
I.7.3 Lavage à haute pression	13
I.7.4 Lavage automatique	14
I.8 Conclusion :	15

Chapitre II

Les Automates Programmables (API)

II.1 Introduction	16
II.2 Historique sur les API.....	16
II.2.1 La logique câblée.....	16
II.2.2 La logique programmée.....	17
II.3 Définition d'un automate programmable	18
II.4 Les fonctions d'API.....	20
II.4.1 Un rôle de commande :.....	20
II.4.2 Un rôle de communication :	20
II.4.3 Nature des informations traitées par l'automate :	21
II.5 Structure matérielle des automates	21
II.5.1 L'unité centrale de traitement (CPU : Central Processing Unit)	22
II.5.2 Mémoires :.....	23
II.5.3 Interfaces d'entrées /sorties	23
II.6 Module d'alimentation :	24
II.7 Les bus.....	25
II.8 Autres composants spécifiques :.....	25
II.9 Caractéristiques techniques	26
II.10 Cycle d'API.....	27
II.11 Critères de choix d'un automate.....	29
II.12 Conclusion :.....	29

Chapitre III

Le S7-300 et Logiciel de Programmation (TIA PORTAL V12)

III.1 Introduction :.....	30
III.2 L'automate S7-300 :.....	30
III.2.1 Caractéristiques de l'API S7-300 :.....	30
III.2.2 Unité centrale (CPU) :.....	31
III.2.3 Connexion MPI :	32
III.2.4 Logement pour la carte mémoire :	32
III.2.5 Logement de pile :.....	33

III.2.6 Coupleurs (IM) :.....	33
III.2.7 Modules de signaux (SM) :	33
III.2.8 Module de fonction (FM) :.....	33
III.2.9 Processus de communication (CP) :.....	33
III.3 Console de programmation (PG).....	33
III.4 Logiciel de programmation STEP7 (TIA Portal).....	34
III.4.1 Description du logiciel TIA Portal	35
III.4.2 Vue du portal et vue du projet.....	35
III.4.3 Adressage des E/S	38
III.4.4 Les variables API	38
III.4.5 Adresse Ethernet de la CPU	39
III.4.6 Programme utilisateur	39
III.4.7 Structure de programme :	40
III.4.8 Les Blocs	40
III.4.9 Mémentos	42
III.4.10 Mnémoniques	42
III.5 Langages de programmation :	43
III.5.1 GRAFCET ou SFC.....	43
III.5.2 Schéma par blocs ou FBD	44
III.5.3 Schéma à contact ou CONT :.....	44
III.5.4 Texte structuré ou ST	45
III.5.5 Liste d'instructions ou IL	45
III.6 Conclusion :.....	46

Chapitre IV

Programmation et simulation de la station de lavage

IV.1 Introduction.....	47
IV.2 Schéma de principe de la station de lavage.....	47
IV.3 Présentation de la partie opérative :	49
IV.4 Présentation de la partie commande :.....	53
IV.4.1 Les entrée/ sorties du programme	54
IV.4.2 La table des mnémoniques	55

IV.5 Simulation du système de lavage	56
IV.6 La conclusion :	67
<i>Conclusion Générale</i>.....	66
<i>Bibliographie</i>	67
<i>Webographie</i>	69
<i>Annexes</i>	71

Liste des tableaux

Tableau III.1 : différents types de variables contenues dans le Step 7.....	43
Tableau IV-1 : pré-actionner et actionneur de la station.....	50
Tableau IV-2 : Capteurs de la station de lavage.....	51

Liste des figures

-----Chapitre I-----	
Figure I-1 : Représentation fonctionnelle d'un système de production	3
Figure I-2 : Distributeur de billet	4
Figure I-3 : Système robotisé pour la vérification automatique des cartes de circuits imprimés	5
Figure I-4 : Structuration en deux parties d'un système automatisé.....	6
Figure I-5 : Description d'un capteur.....	7
Figure I-6 : Capteurs TOR	7
Figure I-7 : Exemple d'un capteur analogique (capteur de vibration).	7
Figure I-8 : Exemple d'un actionneur électrique (une vanne motorisée).....	8
Figure I-9 : Exemple d'un actionneur pneumatique.....	8
Figure I-10 : Exemple d'un actionneur hydraulique	9
Figure I-11 : Exemple des différents types des pré-actionneurs	9
Figure I-12 : Une figure montrant des liaisons par réseaux.	11
Figure I-13 : Lavage manuelle	13
Figure I-14 : Nettoyeur haute pression.....	14
Figure I-15 : Exemple d'une station de lavage aux rouleaux	14
-----Chapitre II-----	
Figure II-1 : Exemple d'une armoire électrique (logique câblée)	17
Figure II-2 : Exemple d'une armoire d'Api.....	18
Figure II-3 : Situation de l'automate dans un système automatisé de production.	19
Figure II-4 : Exemple d'API (siemens S7-300/200).....	20
Figure II-5 : Structure interne d'un API.....	22
Figure II-6 : Exemple d'un CPU (CPU 313)	22
Figure II-7 : Exemple d'un module d'entrées (Sm321)	24
Figure II-8 : Exemple d'une interface de sortie	24
Figure II-9 : Exemple d'un module d'alimentation (ABB)	25
Figure II-10 : Les différents types de bus	25
Figure II-11 : Exemple d'une carte PID	26
Figure II-12 : Un automate monobloc (logo siemens)	26
Figure II-13 : Un automate modulaire (Allen Bradley)	27
Figure II-14 : Cycle de traitement d'information d'un API.....	28

-----Chapitre III-----

Figure III-1 : S7-300.	30
Figure III-2 : Les constituants du S7-300.....	31
Figure III-3 : La CPU 314.....	32
Figure III-4 : Console de programmation PG	34
Figure III-5 : Vue du portale	36
Figure III-6 : Vue du Projet	37
Figure III-7 : Adressage de la CPU	38
Figure III-8 : Les différents types de bloc.....	40
Figure III-9 : Classification des langages de programmation	43
Figure III-10 : Exemple d'un langage SFC.....	43
Figure III-11 : Exemple d'un langage FBD.....	44
Figure III-12 : Exemple d'un langage CONT.....	44
Figure III-13 : Exemple d'un langage ST.....	45
Figure III-14 : Exemple d'un langage IL.....	46

-----Chapitre IV-----

Figure IV-1 : Organisation d'une station automatisée de lavage de voiture.....	47
Figure IV-2 : Actigramme de la station de lavage.	48
Figure IV-3 : Processus de la station de lavage	48
Figure IV-4 : Schéma de puissance de la station de lavage.	50
Figure IV-5 : Schéma de principe de la station de lavage (point de vue commande)	51
Figure IV-6 : Schéma de commande de la station de lavage.	53
Figure IV-7 : Configuration matériels du système sur le logiciel TIA PORTAL.....	54
Figure IV-8 : Table de mnémonique du système dans le logiciel TIA PORTAL.....	55
Figure IV. 9 : CPU et modules d'entrées/sorties en ligne.....	56
Figure IV-10 : Démarrage du cycle du système de lavage	59
Figure IV.11. : Sortie de rouleau horizontal.....	60
Figure IV.12. : Sortie des trois rouleaux	61
Figure IV.13 : Prélavage	62
Figure IV.14 : Savonnage.....	63
Figure IV.15 : Rinçage	64
Figure IV.16 : Sortie Moteur M2	65
Figure IV.17 : Séchage et fin d'opération de lavage.....	66

Listes des abréviations

SAP : Système Automatisés de Production
API : Automate Programmable Industrielle
PLC : Programming Logic Control
P.O : Partie Opérative
P.C : Partie Commande
CPU : Unité Centrale de Processus
RAM: Random Access Memory
ROM: Read only Memory
E/S : Entrée, Sortie
TOR : Tout ou Rien
S7 : STEP
OB : Bloc Organisation
FB : Bloc de Fonction
FC : fonction
DB : Bloc de Données
SDB : Données système
CONT : Langage a Contact (Ladder)
FBS : Langage en Bloc Fonctionnels
SFC : Séquentiel fonction charte ou GRAFCET.
M : Moteurs
K : Contacteur
PS : Power Supply
PID : Proportionnelle intégrale dérivée
MPI : Multi point interface
PG : Console de programmation
IHM : Interface homme machine
INT : Entier
EV : électrovanne.

Introduction
Générale

Introduction Générale

L'automatisation sert à remplacer un système à logique câblé par un appareil électronique programmable ou bien on dit une logique programmée, adapté à l'environnement industriel, les automatismes sont devenus indispensables car ils permettent d'augmenter la productivité et la flexibilité, assurent l'amélioration du qualité ainsi que les conditions de travail. Ce phénomène assuré la réalisation des fonctions d'automatisme simple ou complexe pour commande l'ensemble des constituants d'une chaîne de production automatisé a l'aïd d'un ou des informations logiques, analogiques ou numériques, et garante la surveillance ainsi que la sécurité des processus industriel.

Les automates programmables industriels (API), en anglais programmable logic controllers (PLC), apportent la solution sur mesure pour les besoins d'adaptation de nombreuses activités actuelles. Les API sont devenus aujourd'hui les constituants les plus répandus des installations automatisées, où ils représentent l'outil de base de l'automatisation des systèmes de production.

Ce mémoire a été organisé en quatre chapitres :

- Dans le premier chapitre on va donner un bref aperçu sur les automatismes et sur la structure générale des systèmes automatisés et enfin nous allons donner un petit historique sur les stations de lavage automobile objet de notre étude ;
- Le deuxième chapitre sera consacré à la présentation des automates programmables ainsi que les différents types d'information traitées par l'API et les critères essentiels pour réponde aux besoins souhaités ;
- Dans le troisième chapitre on va décrire l'automate S7-300 qu'on a choisi et le logiciel TIA PORTAL V12 associé pour faire la programmation et concrétiser notre étude ;
- Enfin, le quatrième chapitre sera consacré essentiellement à la description du cahier des charges de notre station et à la programmation et simulation via le logiciel qu'on a choisi et présenté dans le 3^{ème} chapitre.

On terminera enfin notre projet de fin d'études par une conclusion générale.

Chapitre I
Généralité sur
Les Automatismes

I.1 Introduction

Pour assouvir les besoins vitaux des premiers âges, l'homme a utilisé les produits de la nature. Très tôt, il a inventé des outils et des techniques, pour sauvegarder ses forces et économisé son temps ainsi que ses énergies pour d'autres activités.

Au XX^{ème} siècle à travers l'émergence du concept d'information, est apparu l'automatisation et l'informatisation qui permettent de réaliser des actions de plus en plus complexes, plus rapide et plus commode.

Dans ce 1^{er} chapitre nous donnons un bref aperçu sur la structure et la classification des systèmes automatisés ainsi que les différents modes de communication de ces systèmes, et à la fin nous parlons de la station de lavage automobile, qui fait l'objet de notre étude.

I.2 Système automatisé et automatisme

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

L'automatisation conduit à une très grande rapidité, une meilleure régularité des résultats et évite à l'homme des tâches pénibles et répétitives.

- Réalisant leurs fonctions en relative autonomie.
- Assurant un contrôle des performances par la mise en place possible d'une chaîne de retour.

Les objectifs principaux d'automatisation des systèmes de production ont été développés afin de réduire le coût et la complexité de l'installation, de minimiser et simplifier l'intervention de l'homme dans le processus de fabrication et d'assurer une plus grande précision avec le maximum d'économie de ressource donc une ergonomie [4].

Ainsi, un système automatisé est un ensemble d'éléments en interaction, organisés dans un but précis, il agit sur une matière d'œuvre afin de lui donner une valeur ajoutée. Le système automatisé est soumis à des contraintes : énergétiques, de configuration, de réglage et d'exploitation qui interviennent dans tous ces modes de marche et d'arrêt.

Dans ce domaine, l'automatisation tient une place très importante. Aujourd'hui, il serait difficile de concevoir un système de production sans avoir recours aux différentes technologies et composants qui forment les systèmes automatisés de production. Il faut cependant noter que les moyens de production malgré les avantages qu'ils présentent, peut engendrer certains inconvénients qu'il ne faut pas négliger.

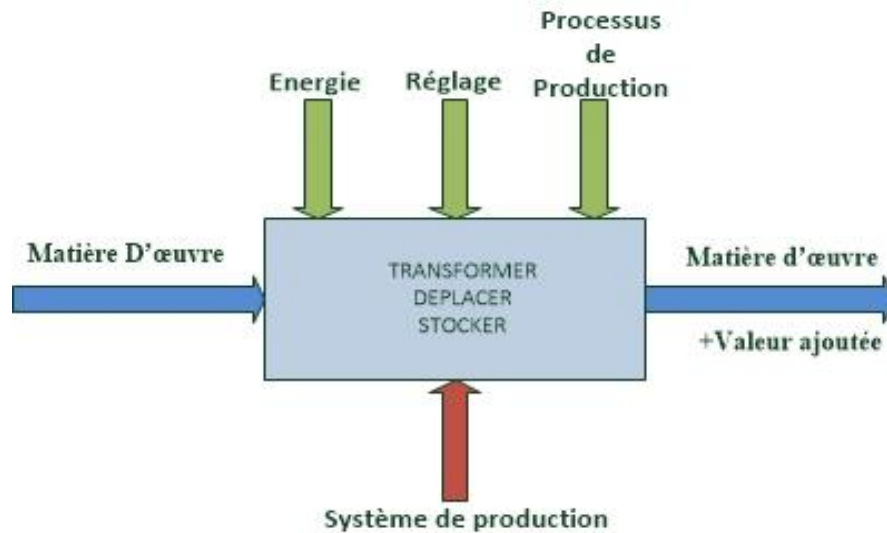


Figure I-1 : Représentation fonctionnelle d'un système de production [2].

La matière d'œuvre peut se présenter sous plusieurs formes à savoir :

- Produit : liquide, solide, gazeux.
- Énergie : électrique, thermique, mécanique, etc.
- Information : physique, audiovisuel, etc.

La valeur ajoutée est caractérisée par sa nature, sa quantité et sa qualité. Elle peut être soit :

- Une modification physique : conversion d'énergie, mécanique, etc.
- Un arrangement particulier : montage, assemblage, etc.
- Un prélèvement d'information : mesure, contrôle, etc.

Environnement ou contexte : c'est le contexte physique, social, économique, etc. qui joue un rôle essentiel dans le fonctionnement du système et influe sur la qualité et/ou la quantité de la valeur ajoutée. [1]

I.3 Avantages et inconvénients d'un système automatisé

Les avantages :

- La capacité de production accélérée.

- L'aptitude à convenir à tous les milieux de production ;
- La souplesse d'utilisation.
- La création de postes d'automaticiens.
- Gain du temps.

Les inconvénients :

- Le coût élevé du matériel, principalement avec les systèmes hydrauliques.
- La maintenance doit être structurée.
- La suppression d'emplois.

I.4 Classification des systèmes automatique :

On peut distinguer, deux grandes familles des SAP :

I.4.1 Les systèmes de grande diffusion :

Encore appelés « grand public », couvrent un usage général. Ils affectent notre existence quotidienne au travers de l'électroménager, de la domotique, de la distribution (distributeur de billets...), du secteur médical du transport, etc. produits en grand nombre, leur conception vise à la recherche de solution optimales fortement intégrées, compatible avec la production de série et les contraintes de la concurrence.



Figure I-2 : Distributeur de billet [W7].

I.4.2 Les systèmes de diffusion limitée :

Ils couvrent un usage spécifique. On rencontre, notamment, ce type d'équipement dans le domaine de la production industrielle : ce sont les systèmes automatiques de production. Le nombre limité, souvent unitaire, de ces systèmes interdit des études longues et coûteuses qui

seraient difficiles à amortir. Ils sont en règle générale, réalisés par l'assemblage de composants et composants standard, pour la partie opérative et la partie commande ils doivent être évolutifs pour s'adapter aux évolutions des besoins des utilisateurs. [2]



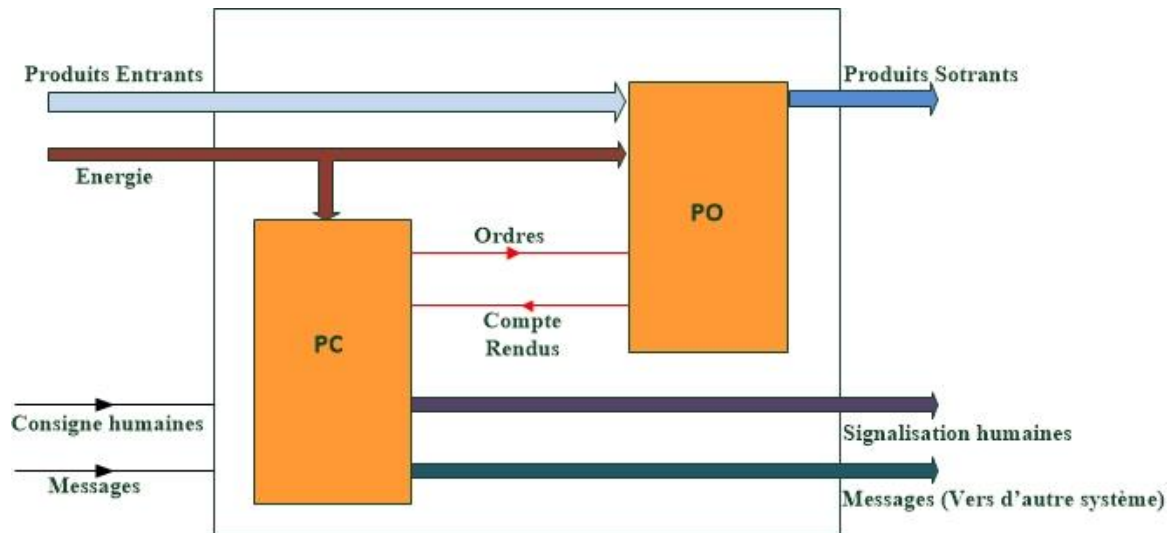
Figure I-3 : Système robotisé pour la vérification automatique des cartes de circuits imprimés [8]

I.5 Structure d'un système automatisé

L'automatisation de la production consiste à transférer tout ou partie des tâches de coordination, auparavant exécutées par des opérateurs humains, dans un ensemble d'objet technique appelé Partie Commande (PC).

La partie commande mémorise le savoir-faire des opérateurs pour obtenir la suite des actions à effectuer sur les matières d'œuvre afin d'élaborer la valeur ajoutée. Elle exploite un ensemble d'informations prélevées sur la partie opérative (PO) pour élaborer la succession des ordres nécessaires pour obtenir les actions souhaitées. [1]

Les informations échangées entre ces deux parties sont des informations internes au systèmes, alors que les consignes, signalisation et autres messages proviennent ou sont destinés à l'extérieure.



FigurI-4 : Structuration en deux parties d'un système automatisé.[2]

I.5.1 Partie opérative

La partie opérative, également appelée processus, constitue l'unité de travail qui exécute la tâche dictée par la partie commande. C'est elle qui assure la transformation des produits entrants permettant d'élaborer la valeur ajoutée recherchée ; elle est le siège des conversions énergétique.

C'est en règle générale, un ensemble mécanisme bien qu'il existe des systèmes dont la partie opérative n'est pas animée, les principales technologies des chaînes d'action, mises en œuvre, utilisent les énergies générées par l'électricité, de l'air comprimé (pneumatique) et des fluides hydrauliques. [2]

Elle se compose généralement de 3 types d'éléments : les capteurs et les actionneurs et les pré-actionneurs.

I.5.1.1 Capteurs

Un capteur est un organe de prélèvement d'informations qui élabore à partir d'une grandeur physique, une autre grandeur physique de nature différente (souvent électrique).

Les capteurs servent à détecter des positions physiques, des pressions, des températures, des forces, des vitesses, etc. L'information captée par la partie opérative est transmise à la partie commande par l'intermédiaire d'une interface d'entrée. On peut classer les capteurs en 2 groupes en fonction de la nature de l'information délivrée en sortie. [3]

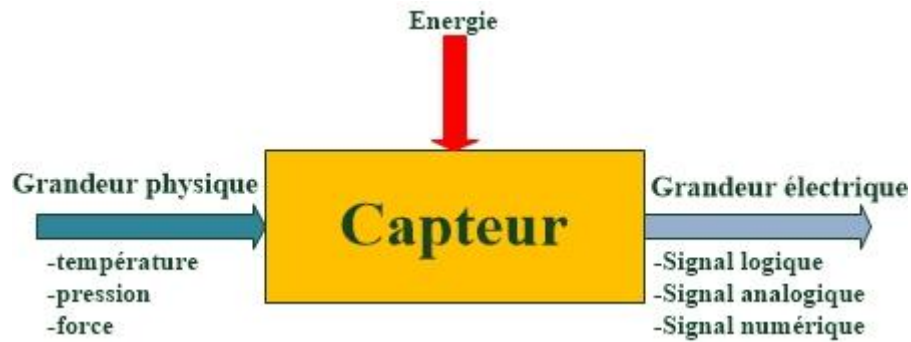


Figure I-5 : Description d'un capteur.

Les capteurs tout ou rien ou bien TOR comme illustré dans la Figure I-6 fournissent une information électrique binaire (0 ou 1), le type de signal de sortie donne 0V ou 5V.



Figure I-6 : Capteurs TOR [W1]

Les capteurs analogiques (Fig I-7) fournissent une information électrique à intensité variable en courant (4-20 mA) ou en tension (0-10 V). Cette information est convertie en valeur numérique et stockée dans un registre de l'automate pour être ensuite traitée par la partie commande.



Figure I-7 : Exemple d'un capteur analogique (capteur de vibration). [W2]

I.5.1.2 Actionneurs

Les actionneurs sont les éléments de puissance permettant d'engendrer les mouvements de rotation ou de translation dans la partie opérative. On distingue 3 catégories d'actionneurs pour répondre aux besoins variés des machines [3] :

Les actionneurs électriques (Fig I-8), qui transforment l'énergie électrique en énergie mécanique sous forme de mouvement de rotation (moteur) ou de translation (électroaimant).



Figure I-8 : Exemple d'un actionneur électrique (une vanne motorisée) [W3].

Les actionneurs pneumatiques (Fig. 9), qui transforment l'énergie pneumatique en mouvement de rotation (moteur pneumatique) ou de translation (vérin).



Figure I-9 : Exemple d'un actionneur pneumatique [W4]

Les actionneurs hydrauliques (Fig I-10), qui transforment l'énergie hydraulique en mouvement de rotation (moteur) ou de translation (vérin). Les actionneurs hydrauliques sont utilisés pour engendrer des mouvements nécessitant beaucoup de force.



Figure I-10 : Exemple d'un actionneur hydraulique [W5].

I.5.1.3 Les pré-actionneurs

Les pré-actionneurs sont des éléments qui, sur réception d'un ordre de la partie commande, fournit à l'actionneur qui lui est associé l'énergie requise pour fonctionner. Les pré-actionneurs utilisés avec les actionneurs électriques sont principalement les contacteurs magnétiques (Fig I-11 2), et les variateurs de vitesse (Fig I-11-3), Pour les actionneurs pneumatiques et hydrauliques, les pré-actionneurs les mieux adaptés à la tâche sont les distributeurs électropneumatiques (Fig I-11 1), ou électrohydrauliques, selon le cas [3].



Figure I-11 : Exemple des différents types des pré-actionneurs [W6]

I.5.2 Partie commande

Elle commande la partie opérative pour obtenir les effets voulus, par l'émission d'ordres en fonction d'information disponibles elle peut aussi échanger des informations avec des humaines ou d'autre système.

Les parties commandes sont construites à partir de constituants électronique et électrique et s'appuient essentiellement sur des technologies programmées (automate programmable,

micro-contrôleurs, etc.). Les parties commandes sont ainsi de plus en plus réparties et distribuées et sont alors reliées à l'aide de réseaux [2].

La partie commande peut avoir à traiter des éléments de logique combinatoire et séquentielle, des opérations logiques et numériques en effectuant des opérations sur des valeurs alphanumériques ou numériques, et des fonctions d'asservissement en boucle fermée

Le traitement d'information est appelé à coordonner 3 dialogues :

Le dialogue entre la partie commande et la partie opérative commande les actionneurs au moyen des pré-actionneurs.

Le dialogue homme-machine permet d'exploiter la machine en émettant des consignes de marche ou d'arrêt (par l'intermédiaire de bouton-poussoir, de boutons sélecteurs, de roues codeuses, etc.) et en recevant de l'information sur l'état de la machine (visualisé à l'aide de lampes témoins, d'afficheurs numériques à 7 segments, etc.).

Le dialogue entre les parties commande des machines rend possible l'échange d'information entre plusieurs machines pouvant participer à une même production [3].

I.6 Communication dans un système automatisé

Un système automatisé doit être mis en liaison avec ses composants afin de permettre des échanges d'information de différents niveaux. Un système de production automatisé utilise à cette fin 3 niveaux de commande assurant des fonctions définies ; ce sont les niveaux de commande 0, 1 et 2.

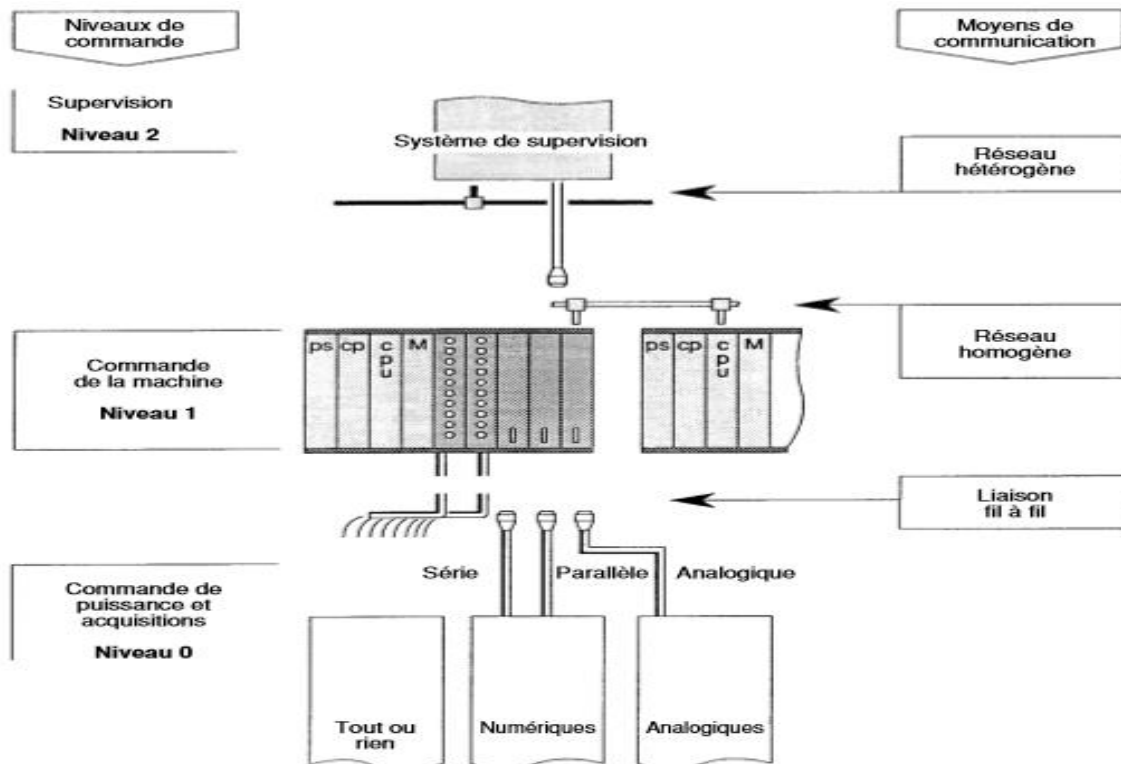


Figure I-12 : Une figure montrant des liaisons par réseaux.[3]

I.6.1 Le niveau 0 :

Permet l'acquisition d'information et la commande des actions de la partie opérative. Les capteurs et les pré-actionneurs Tout Ou Rien sont reliés à l'automate programmable par une liaison fil à fil.

I.6.2 Le niveau 1 :

Permet la coordination au moyen d'échanges d'information entre les éléments de traitement. Un câble d'interconnexion raccorde les automates les uns aux autres par l'intermédiaire de connecteurs prévus à cette fin. Lorsque l'on branche un terminal de programmation sur le réseau, on peut avoir un accès direct aux programmes résidant dans les différents automates. Dans les applications industrielles, on distingue 2 types de réseaux.

Les réseaux homogènes sont constitués par des automates programmables de même nature et compatibles.

Les réseaux hétérogènes utilisent des éléments interconnectés de nature différente (automates programmables, machines à commande numérique, robots) [3].

I.6.3 Le niveau 2 :

Permet la supervision des différentes unités de production. On y assure la centralisation de l'information et le dialogue avec d'autres organes de communication. Le dialogue de supervision assure la conduite du système de production. Il permet :

- De lancer des campagnes de production ;
- D'effectuer des réglages ;
- D'afficher des messages ;
- De gérer les défauts ;
- De produire des statistiques sur la production ;
- De gérer l'entretien préventif ;

Dans notre étude on s'intéresse au système de lavage de voiture qui fait partie des systèmes de grande diffusion comme il a été défini précédemment ; dans ce qui suit on va donner un bref aperçu sur l'historique et les différents types de lavage automobile existants.

I.7 Lavage automobile

L'entretien d'un véhicule passe obligatoirement par le lavage de celui-ci de manière régulière afin d'éviter le dépôt de poussières et la dégradation de la peinture de votre automobile par la pollution et les intempéries. Pour nettoyer votre voiture, deux solutions s'offrent à vous : le lavage automatique et le lavage manuel

I.7.1 Évolution du monde de lavage

Le lavage automobile est passé par plusieurs étapes et a suivi également l'évolution du domaine de l'automobile et de d'automatisme. En 1963, le premier portique de lavage à haute pression a vu le jour ; Dix ans plus tard le premier portique de lavage et séchage automatique avec brosses a été inventé ; en 1988 le portique automatique a été relié avec un panneau de commande programmable, quelques années plus tard exactement en 1991 le portique de lavage est devenu multi-programme et le développement de ces automatisés est toujours à l'ordre du jour. [15]

I.7.2 Lavage manuel

Lorsque votre auto a besoin d'être lavée de fond en comble, rien ne vaut le lavage à la main qui est le seul pouvant vous garantir une propreté parfaite. Toutefois, cette méthode reste la plus longue, contraignante et il est nécessaire de savoir s'y prendre pour laver la voiture.

Les brosses, le seau, les éponges, les chiffons, les lave-vitres sont autant de produits indispensables pour cette tâche, mais il est également possible d'y ajouter par exemple une brosse à dents afin de frotter les endroits les plus inaccessibles.



Figure I-13 : Lavage manuelle [W9]

Laver sa voiture à grande eau, dans le jardin ou dans la rue, est néfaste pour l'environnement. Les résidus s'écoulent dans le circuit des eaux de ruissellement, qui n'est pas toujours équipé pour recevoir ces polluants, ou s'infiltrent dans le sol ou les cours d'eau, risquant ainsi de contaminer les nappes phréatiques. De plus, ce type de lavage consomme beaucoup d'eau.

I.7.3 Lavage à haute pression

Vous pouvez en premier choix opter pour le lavage à haute pression. Celui-ci consiste tout simplement à asperger votre voiture grâce à un appareil de nettoyage qui propulse de l'eau à haute pression. Cette méthode convient parfaitement pour les lavages rapides et permet notamment de nettoyer de manière précise les endroits peu accessibles de votre auto tels que les bas de caisses, les jantes ou encore les pare-chocs avant et arrière.



Figure I-14 : Nettoyeur haute pression [W10].

Néanmoins certaines tâches robustes pourraient vous obliger à réaliser une finition de nettoyage à l'éponge. De plus, il est conseillé de ne pas mettre le jet trop près de la carrosserie afin d'éviter les micro-rayures sur les voitures peu résistantes ou très vieilles. Faire laver sa voiture en haute pression dans une station consomme moins d'eau que de la nettoyer à la main.

I.7.4 Lavage automatique

Plus pratique, plus rapide et plus fréquente, la méthode qui consiste à se rendre dans un lave-auto afin de procéder à un lavage automatique est utilisée par la majorité des conducteurs. La technique est simple et consiste à se rendre dans une station de lavage où votre véhicule sera lavé de manière automatique ou semi-automatique selon la méthode de lavage choisie. Les stations de lavage ont l'obligation de rétablir une qualité sanitaire et écologique compatible avec le milieu naturel. Les plus récentes possèdent donc toutes un dispositif de traitement des eaux usées.[14]



Figure I-15 : Exemple d'une station de lavage aux rouleaux [W11].

En prend l'exemple d'un lavage aux rouleaux. Cette technique qui ne demande aucun effort, est très rapide et reste la moins coûteuse.

Le véhicule passe tout simplement à travers un mécanisme de rouleaux qui tournent sur eux-mêmes afin de la laver. Il faut penser à bien fermer les vitres et à baisser l'antenne avant le passage.

Cependant, ce système a la mauvaise réputation de laisser des micro-rayures sur la carrosserie à cause des fibres synthétiques des brosses. Cela est plus ou moins vrai selon le type de rouleaux utilisé (les plus récents pallient plus ou moins le problème). [5]

I.8 Conclusion :

Dans ce chapitre on a défini le système automatisé de production et ces différents éléments ainsi que les avantages qui assurent une meilleure production dans le milieu industriel et conduit de production d'une installation industrielle, et dernièrement on a parlé sur la station de lavage en général.

Chapitre II
Les Automates
Programmables (API)

II.1 Introduction

Dans le monde industriel, les exigences attendues de l'automatisation ont bien évolué. Parmi les éléments les plus répandus dans un système automatisé est API (l'automate programmable industriel).

Ils sont apparus aux Etats-Unis vers la fin des années soixante, à la demande de l'industrie automobile américaine qui réclamait plus d'adaptabilité de leurs systèmes de commande. Les ingénieurs américains ont résolu le problème en créant un nouveau type de produit nommée automates programmables. L'API est la première machine à langage c'est-à-dire un des calculateurs logiques dont le jeu d'instruction est orienté vers les problèmes de logique et des systèmes a évolution Séquentielles.

Ils n'étaient rentables que pour des installations d'une certaine complexité, mais la situation a très vite changé, ce qui a rendu les systèmes câblés obsolètes. De nombreux modèles d'automates sont aujourd'hui disponibles ; depuis les nano automates bien adaptés aux machines et aux installations simples avec un petit nombre d'entrées/sorties, jusqu'aux automates multifonctions capables de gérer plusieurs entrées/sorties et destinés au pilotage de processus complexes. [6]

Ce chapitre sera consacré aux automates programmables d'une façon générale et les différents constituants d'un API.

II.2 Historique sur les API

Avant : utilisation de relais électromagnétiques et de systèmes pneumatiques pour la réalisation des parties commandes \Rightarrow logique câblée

II.2.1 La logique câblée

La technologie câblée consiste à raccorder des modules par des liaisons matérielles selon un schéma fourni par la description. Ces modules peuvent être électromagnétiques, électriques, pneumatiques ou fluidiques.

En électricité ou en électronique, les liaisons sont faites par câble électrique. En pneumatique et fluide, il s'agit de canalisations reliant les différents composants.



Figure II-1 : Exemple d'une armoire électrique (logique câblée) [W12]

Les outils câblés sont utilisés dans l'industrie où l'on apprécie ; à savoir la rapidité et le parallélisme. Ils souffrent cependant d'un certain nombre de limitations parmi lesquelles nous citons :

- Leur encombrement (poids et volume).
- Leur manque de souplesse vis-à-vis de la mise au point des commandes et de l'évolution de celles-ci (améliorations, nouvelles fonctions, modification, etc.) : Toute modification impose la modification de câblage voire un changement de composants.
- Leur difficulté de maîtriser des problèmes complexes.
- La complexité de recherche des pannes et donc du dépannage.
- Leur coût élevé pour les systèmes complexes.

Solution : utilisation de systèmes à base de microprocesseurs permettant une modification facile et pratique des systèmes à exemple automatisé \Rightarrow logique programmée.

II.2.2 La logique programmée

La technologie programmable consiste c'est-à-dire des machines destinées à traiter de l'information. Leur utilisation en gestion et en calcul scientifique est connue. Alors, les applications techniques relèvent de l'informatique à substituer le fonctionnement de l'automatise par un programme chargé sur un constituant programmable industrielle.

L'informatique industrielle est une discipline conjuguant les théories de l'automatique et les moyens de l'informatique dans le but de résoudre des problèmes de nature industrielle.



Figure II-2 : Exemple d'une armoire d'Api [W13]

L'informatique offre donc une alternative technologique à l'automaticien et lui ouvre des possibilités nouvelles liées à la puissance de traitement et aux facilités de mémorisation de l'information. En termes d'avantage, nous citons :

- Moins de câble et d'encombrement.
- Fiabilité de l'automatisme.
- Facilité de modification.
- Flexibilité.
- Résolution des problèmes complexes.

Le constituant programmable peut être soit un micro-ordinateur, soit une carte électronique ou bien un automate programmable.

II.3 Définition d'un automate programmable

Un API est une machine électronique spécialisée dans la conduite et la surveillance en temps réel de processus industriels. Il exécute une suite d'instructions introduites dans ses mémoires sous forme de programme, et s'apparente par conséquent aux machines de traitement d'information.

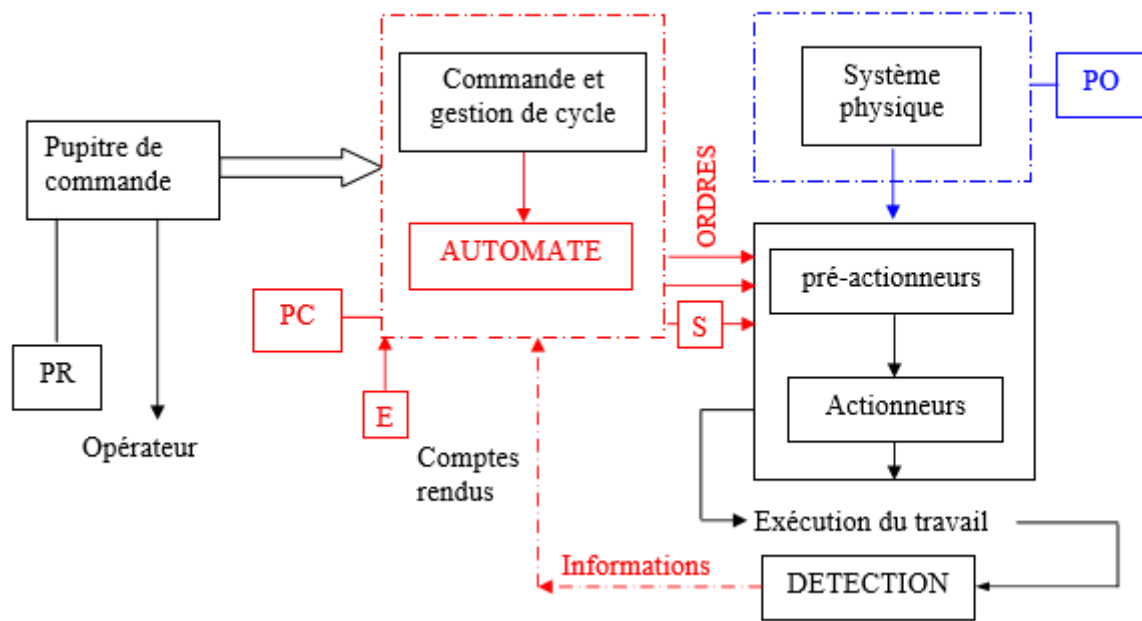


Figure II-3 : Situation de l'automate dans un système automatisé de production. [4]

Trois caractéristiques fondamentales le distinguent des outils informatiques tels que les ordinateurs utilisés dans les entreprises :

- Il peut être directement connecté aux capteurs et pré-actionneurs grâce à ses entrées/sorties.
- Il est conçu pour fonctionner dans des ambiances industrielles sévères (température, vibrations, microcoupures de la tension d'alimentation, parasites, etc.).
- Enfin, sa programmation à partir des langages spécialement développés pour le traitement des fonctions d'automatismes facilitent son exploitation et sa mise en œuvre.

[7]

Selon la norme française EN 61131-1, un API est un système électronique fonctionnant de manière numérique, destiné à être utilisé dans un environnement industriel, qui utilise une mémoire programmable pour le stockage interne des instructions orientées utilisateur aux fins de mise en œuvre des fonctions spécifiques, telles que des fonctions de logique, de mise en séquence, de temporisation, de comptage et de calcul arithmétique, pour commander au moyen d'entrées et de sorties Tout ou Rien ou analogiques divers types de machines ou de processus. L'API et ses périphériques associés sont conçus pour pouvoir facilement s'intégrer

à un système d'automatisme industriel et être facilement utilisés dans toutes leurs fonctions prévues. [7]

Donc on peut résumer que L'API : est un appareil électronique de traitement de l'information, il est construit autour d'un processeur numérique et doté de possibilités d'échanges avec d'autres processeurs.

Il fonctionne grâce à une protection adaptée dans des conditions industrielles et peut être relié à de nombreux signaux physiques.

Effectue des fonctions d'automatisme programmées telles que : logique combinatoire, logique séquentiel, temporisation, comptage, calculs numériques, asservissement, régulation, Commande, mesure et contrôle au moyen de signaux d'entrées et de sorties toutes machines et processus, en environnement industriel. [9]



Figure II-4 : Exemple d'API (siemens S7-300/200) [14].

II.4 Les fonctions d'API

L'automate doit remplir :

II.4.1 Un rôle de commande :

Où il est un composant d'automatisme, élaborant des actions, suivant un algorithme approprié, à partir des informations que lui fournissent des détecteurs (Tout ou Rien) ou des capteurs (analogiques ou numériques).

II.4.2 Un rôle de communication :

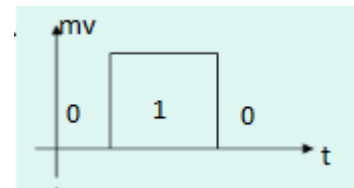
Où il est capable de communiquer avec :

- Des opérateurs humains : c'est le dialogue d'exploitation,
- D'autres processeurs hiérarchiquement :
- Supérieurs (calculateur de gestion de production),
- Égaux (autres automates intervenant dans la même chaîne)
- Inférieurs (instruments intelligents).

II.4.3 Nature des informations traitées par l'automate :

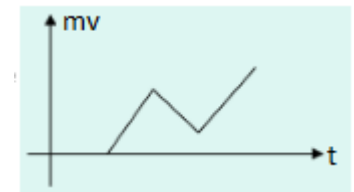
Tout ou rien (T.O.R.) :

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



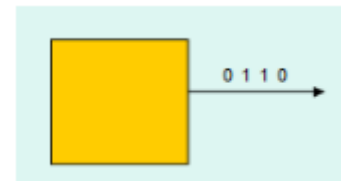
Analogique :

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.



II.5 Structure matérielle des automates

Il se compose de plusieurs parties et notamment d'une mémoire programmable dans laquelle l'opérateur écrit, dans un langage propre à l'automate, des directives concernant le déroulement du processus à automatiser.

Son rôle consiste donc à fournir des ordres à la PO en vue d'exécuter un travail précis comme par exemple la sortie ou la rentrée d'une tige de vérin, l'ouverture ou la fermeture d'une vanne. La PO lui donnera en retour des informations relatives à l'exécution du travail [8]. L'API est composé quatre parties principales :

La structure interne d'un API est représentée comme suit :

- Un processeur.
- Une mémoire.
- Des interfaces d'Entrées/Sorties.
- Une alimentation (240 Vac \rightarrow 24 Vcc).

Ces quatre parties sont reliées entre elles par des bus (ensemble câble autorisant le passage de l'information entre ces 4 secteurs de l'API).[7]

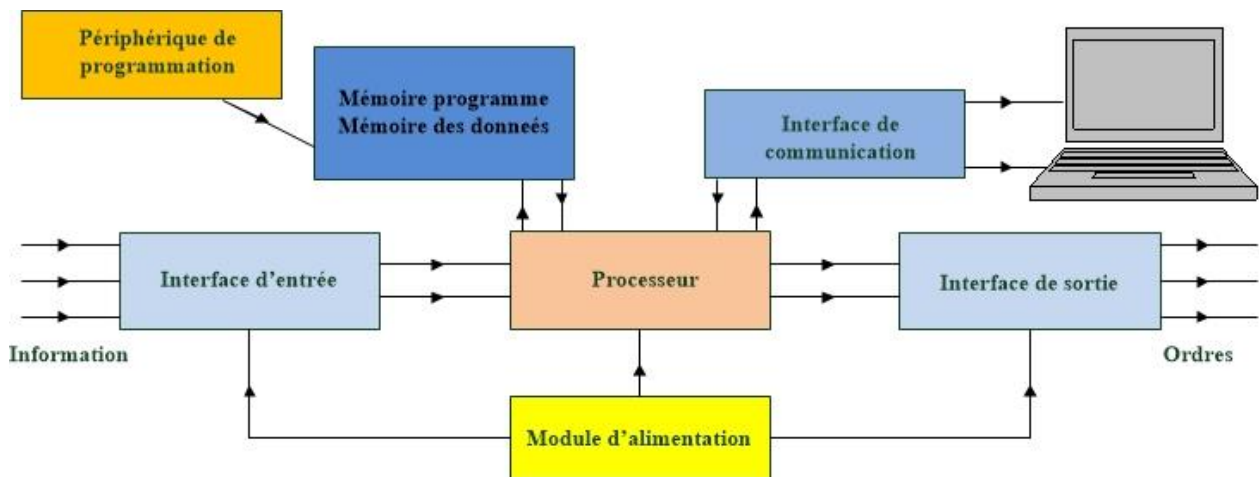


Figure II-5 : Structure interne d'un API.

II.5.1 L'unité centrale de traitement (CPU : Central Processing Unit)

C'est le cœur de la machine, Processeur appelé unité de traitement, il assure le contrôle de l'ensemble de la machine et effectue les traitements demandés par les instructions du programme. Il réalise les fonctions logiques, temporisation, comptage, calcul. Il comporte un certain nombre de registres (compteur ordinal, registre d'instructions, registre d'adresse, registres de données, accumulateurs, ...). Il est connecté aux autres éléments (mémoires, interfaces d'E/S, ...) par l'intermédiaire des bus. [10]



Figure II-6 : Exemple d'un CPU (CPU 313) [W15].

II.5.2 Mémoires :

Mémoire : La mémoire centrale est découpée en plusieurs zones :

- Zone mémoire programme ;
- Zone mémoire des données (états des E/S, valeurs des compteurs, temporisations, ...) ;
- Zone où sont stockées des résultats de calcul utilisées ultérieurement dans le programme ;
- Zone pour les variables internes.

Il existe différents types de mémoires :

Mémoires vives : RAM ce sont des mémoires volatiles : est utilisée pour stocker les données et les programmes lors du fonctionnement ; elles perdent l'information en cas de coupure de l'alimentation. Certaines d'elles sont équipées de batteries de sauvegarde (autonomie réduite). Elles sont accessibles en lecture et en écriture.

Mémoires mortes : les contenus sont figés. Ce sont des mémoires à lecture seule. Les informations sont conservées en permanence sans source externe.

ROM : mémoire programmé par le fabricant et ineffaçables ; elle représente un espace de stockage permanent pour le système d'exploitation et les données figées utilisées par le CPU (PROM, REEPROM ou EPROM, EEPROM).

II.5.3 Interfaces d'entrées /sorties

II.5.3.1 Interfaces d'entrées

Permettent de recevoir les informations du processus ou du pupitre et de mettre en forme (filtrage,) ce signal, tout en l'isolant électriquement. Il existe sur le marché des modèles de cartes d'entrée à 4,8,16, 32 ou 64 voies.

Les modules (cartes) d'entrées peuvent être :

- Logique (tout ou rien) adapté à des interrupteurs de fin de course, des capteurs de proximité...
- Analogique adaptées à des capteurs de température, de pression, débit, vitesse ou autres variables continues.



Figure II-7 : Exemple d'un module d'entrées (Sm321) [W16]

II.5.3.2 Interfaces de sorties

Permettent de commander les divers pré actionneurs et éléments de signalisation du processus. Elles traduisent les commandes du processeur aux signaux industriels tout en assurant l'isolement électrique.

Les modules (cartes) de sorties peuvent être :

- Logiques (tout ou rien) adaptées aux lampes, aux contacteurs, voyants, aux systèmes de verrouillage de porte, etc.
- Analogique adaptées aux moteurs (AC et DC), aux vannes continues, etc.



Figure II-8 : Exemple d'une interface de sortie [W17]

II.6 Module d'alimentation :

Le module d'alimentation (PS) transforme l'énergie externe (tension alternative) provenant du réseau afin de fournir aux différents modules de l'API les niveaux de tension nécessaires à leur bon fonctionnement. Il délivre, à partir du 220 V alternatif, des sources de tension nécessaires à l'automate tels que : +5V, 12V et 24V en continu.



Figure II-9 : Exemple d'un module d'alimentation (ABB) [W18]

II.7 Les bus

Le bus (interne) est un ensemble de pistes conductrices (pistes en cuivre) par lequel s'acheminent une information binaire (suite de 0 ou 1) sur chaque fil :

Le bus de données transporte les données utilisées dans les traitements effectués par le CPU.

Le bus d'adresses transporte les adresses des emplacements mémoire que le CPU utilise pour accéder aux données enregistrées dans ces emplacements.

Le bus de commandes transporte les signaux utilisés par le CPU pour le contrôle, tels que tops de synchronisation, sens des échanges, contrôle de validité des échanges, etc.

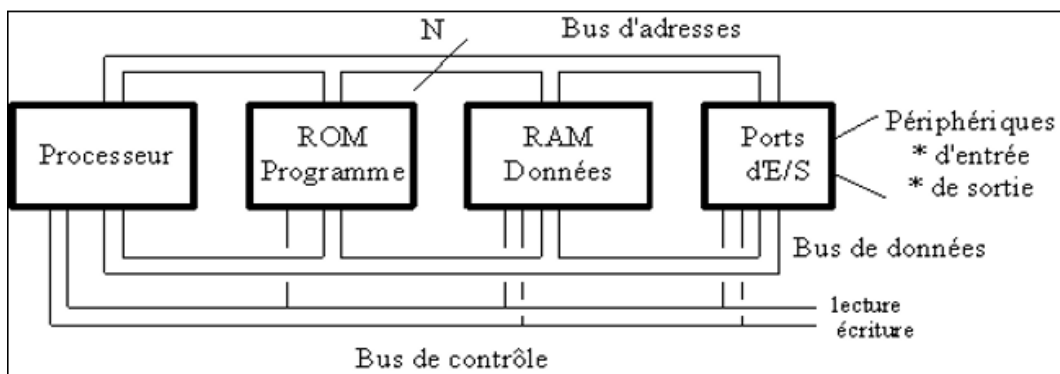


Figure II-10 : Les différents types de bus [8]

II.8 Autres composants spécifiques :

- Cartes de comptage rapide.
- Cartes /modules communication (RS485, Ethernet ...).
- Cartes de régulation PID.
- Cartes d'E/S intelligents.

- Les modules d'axes.
- Cartes Web, etc.



Figure II-11 : Exemple d'une carte PID [W19]

II.9 Caractéristiques techniques

Structure externe : compacte ou modulaire :

La conception monobloc (nano-automates ou micro-automates). Les capacités d'association et de communication des matériels récents, même parmi les gammes les plus basses, offrent de nombreuses possibilités de mise en réseau de ces constituants monoblocs.[10]



Figure II-12 : Un automate monobloc (logo siemens) [20]

La conception modulaire (automates multifonctions). Cette organisation réunit un module d'alimentation et une unité centrale dans un bac pouvant accueillir divers types de modules allant des modules d'E/S TOR aux plus complexes. Elle permet ainsi des configurations évolutives et personnalisées. [10]



Figure II-13 : Un automate modulaire (Allen Bradley) [W21]

- La tension d'alimentation : 220V, 240V...en courant alternatif.
- La taille de sa mémoire : 20koctet, 128koctet, ...
- Temps de scrutation ;
- Sauvegarde (EPROM, EEPROM, ...) ;
- Nombre d'entrées / sorties ;
- Modules complémentaires (analogique, communication,) ;
- Langages de programmation ;[11]

II.10 Cycle d'API

Cycle de l'API : Une caractéristique originale et unique des API est le fonctionnement cyclique de l'UC. Le programme est composé d'une suite d'instructions placés dans la mémoire (RAM, EPROM, EEPROM,). Il peut être par exemple, constitué des phases :

Traitement interne : l'automate effectue des opérations de contrôle et met à jour certains paramètres systèmes (détection des passages en RUN/STOP...);

Lecture des entrées : l'automate lit les entrées (de façon synchrone) et les recopie dans la mémoire image des entrées ;

Exécution du programme : l'automate exécute le programme instruction par instruction et écrit les sorties dans la mémoire image des sorties ;

Ecriture des sorties : l'automate bascule les différentes sorties (de façon synchrone) aux positions définies dans la mémoire image des sorties.

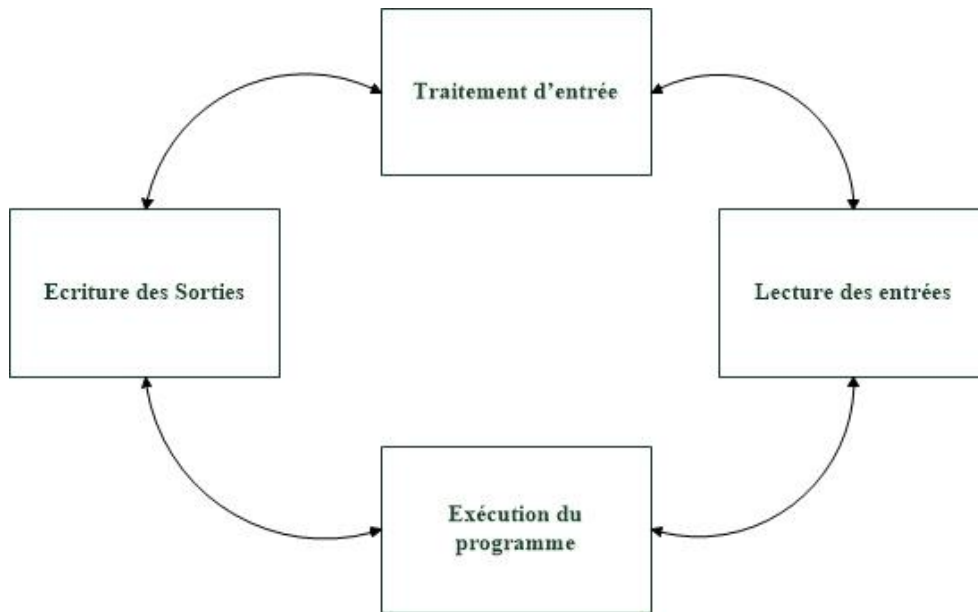


Figure II-14 : Cycle de traitement d'information d'un API.

Ce cycle se reproduit ainsi indéfiniment : à chaque cycle, tout le programme est exécuté. La durée d'un cycle est de l'ordre de 20 ms.

Suivant la conception de l'API ou de la programmation, on distingue 3 autres possibilités :

- Toutes les entrées sont acquises au début du cycle et les sorties sont commandées en fin de cycle, c'est-à-dire après que toutes les équations aient été résolues.
- Toutes les entrées sont acquises au début du cycle et les sorties sont activées après chaque résolution d'équation intégrée dans le programme. On parle d'un cycle synchrone vis à vis des entrées et asynchrone vis à vis des sorties ;
- Les équations sont traitées une par une en prenant uniquement la valeur des entrées concernées. Ce cycle peut présenter un inconvénient, en effet, une entrée peut être appelée plusieurs fois avec des valeurs différentes pour plusieurs équations au cours du même cycle. Risque de provocation d'aléas.
- Les entrées sont scrutées toutes les n ms, quelle que soit la durée du cycle ; les sorties sont activées à la demande. [10]

Remarques :

- L'organisation du cycle devient plus complexe lorsque l'on se trouve en présence d'instructions de sauts particulières.

- Si le cycle n'est pas effectué dans une période définie par l'automate, exemple d'un programme ou l'on engendre une boucle infinie involontaire, une alarme est déclenchée par l'intermédiaire du « chien de garde » qui est une temporisation exécutée au début de chaque cycle.
- Le temps effectué par l'UC pour exécuter un programme de 1 kmots (210 = 1024 mots), est appelé période d'un automate programmable. Ce temps varie d'un modèle à un autre et ne se différencie qu'à la milliseconde.

II.11 Critères de choix d'un automate

- Le nombre et la nature des E/S.
- La nature du traitement (temporisation, comptage, ...).
- Les moyens de dialogue et le langage de programmation.
- La communication avec les autres systèmes.
- La facilité de l'exploitation et de la maintenance.
- Les moyens de sauvegarde des données et des programmes.
- La fiabilité, robustesse, immunité aux parasites.
- La documentation, le service après-vente, durée de la garantie, la formation.
- Le type de processeur (la taille mémoire, la vitesse de traitement, les fonctions spéciales offerts...).

II.12 Conclusion :

Dans ce chapitre, on a donné un aperçu sur les API et leur fonction dans un milieu industrielle ; la nature des informations traitées ; ainsi que la structure interne et externe des API et à la fin de chapitre on propose les points essentiels pour choisir un API qui réponde à des besoins bien spécifiques

Chapitre III

Le S7-300 et Logiciel de Programmation (TIA PORTAL V12)

III.1 Introduction :

Ce chapitre sera en 1^{ère} partie consacrée essentiellement à la description de l'automates programmable SIEMENS S7-300 et à structures modulaire qu'on utilisera dans notre étude. Dans la 2^{ème} partie on présentera les différents aspects de la programmation dans le domaine de l'automatisme à savoir les langages, Grafset SFC (Graphe Fonctionnel de Commande des Étapes et Transitions), schéma à contact Ladder ou CONT ; schéma par block FDB et Texte structuré ou ST, Liste d'instructions ou IL.

III.2 L'automate S7-300 :

C'est un appareil offre des performances intéressent dans le domaine automatisme, souplesse d'utilisation et ne nécessite pas d'entretien spécifique (Figure III-1). Il peut supporter jusqu'au 512 Entrée/Sortie tout ou rien (TOR) et 64 E/S analogique, comme il peut être configuré avec un maximum de 32 modules de signaux pouvant être répartis sur un châssis de base et trois châssis d'extensions.



Figure III-1 : S7-300[W22].

III.2.1 Caractéristiques de l'API S7-300 :

La gamme S7-300 offre une grande variété de CPU tels que la CPU312, 314M, Chacune possède certaines caractéristiques différentes des autres. Par conséquent, le choix de la CPU, pour un problème d'automatisation donné, est conditionné par les caractéristiques offertes par la CPU choisie.

- Gamme complète de modules.
- Possibilité d'extension jusqu'à 32 modules.
- Bus de fond de panier intégré aux modules. Possibilité de mise en réseau avec : l'interface multipoint (MPI), PROFIBUS,
- Raccordement central de la PG avec accès à tous les modules.
- Liberté de montage aux différents emplacements.
- Configuration et paramétrage à l'aide de l'outil « configuration matérielle ».

III.2.2 Constitution de l'API S7-300 :

Dans l'automate programmable S7-300, le profilé support (châssis) contient 11 emplacements au maximum des modules est fixée dans l'ordre suivant

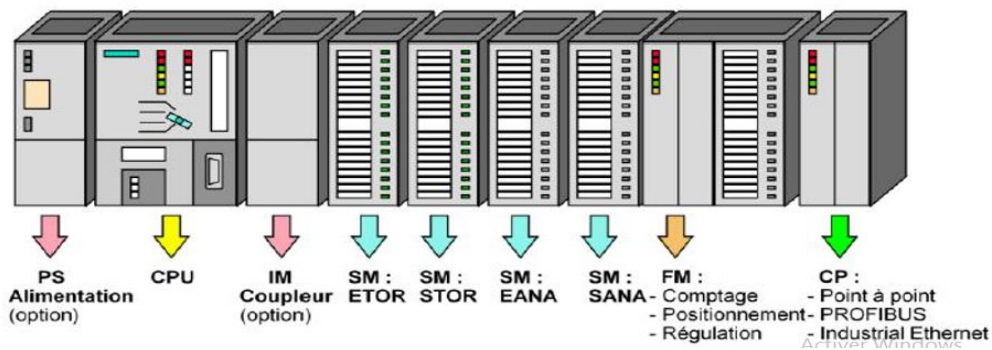


Figure III-2 : Les constituants du S7-300[16]

III.2.1 Module d'alimentation (PS) :

Il convertit la tension secteur (AC120/230 V) en une tension d'alimentation (24 Vcc) pour les modules de l'automate. Comme il montre la (FigureIII-2) ; Les tensions pour les capteurs, actionneurs qui dépassent le 24 V sont fournies par des blocs d'alimentation ou transformateurs supplémentaires.

III.2.2 Unité centrale (CPU) :

L'unité centrale a pour fonction la lecture des états des signaux d'entrées et la commande des sorties, l'exécution de programme d'utilisateur et permet aussi le réglage du comportement au démarrage et le diagnostic des défauts par les LEDS Elle est disponible en différentes versions dans certains API ; A utilisation standard : CPU 313, CPU 314, CPU 315, CPU 316 et d'autres CPU avec des fonctions intégrées : CPU 312 IFM, CPU 314 IFM.



Figure III-3 : La CPU 314[W16]

Les LEDS pour la signalisation d'état et de défauts :

SF = signalisation groupée de défauts, erreurs dans le programme ou défauts sur un module de diagnostic.

BATF = défaillance de la pile, pile déchargée ou absente.

DC5V = signalisation de la tension d'alimentation interne 5V.

FRCE = forçage, signalisation qu'au moins une entrée ou une sortie est forcée de manière permanente.

RUN = Clignotement à la mise en route de la CPU, allumage continu en mode RUN.

STOP = Allumage continu en mode STOP, clignotement lent lorsqu'un effacement Général est requis et clignotement rapide lorsqu'un effacement général est en cours.

- Un commutateur de mode de fonctionnement **MRES** = Effacement général (Module Reset).

STOP = Arrêt, le programme n'est pas exécuté.

RUN = Le programme est exécuté, accès en lecture seule avec une PG.

RUN-P = Le programme est exécuté, accès en écriture et en lecture avec une PG : la CPU traite le programme.

III.2.3 Connexion MPI :

Le port interface MPI se présente sous la forme d'une connexion à neuf ponts, protégée par un cache. Elle permet de réaliser des liaisons multipoints entre plusieurs appareils S7.

III.2.4 Logement pour la carte mémoire :

Une carte mémoire peut être à cet emplacement. Elle conserve le contenu du programme en cas de coupure de courant, même en l'absence de pile.

III.2.5 Logement de pile :

Un emplacement protégé par un cache est prévu pour une pile au lithium permettant de sauvegarder le contenu de la mémoire RAM en cas de coupure de courant.

III.2.6 Coupleurs (IM) :

Les coupleurs IM360/IM361 ou IM365 permettent de réaliser des configurations à plusieurs châssis, ces modules d'interface servent donc à interconnecter le châssis de base ZG au châssis d'extension EG dans une configuration à plusieurs rangées.

III.2.7 Modules de signaux (SM) :

ETOR = Modules d'entrées Tout Ou Rien permettent le raccordement de l'API aux capteurs logique.

STOR = Modules de sorties tout ou rien permettant le raccordement de l'API aux pré-actionneurs, chaque sortie est identifiée par une LED.

EANA = Modules d'entrées analogiques convertissent les signaux analogiques en valeurs numériques traitées par la CPU en utilisant des CAN.

SANA = Modules de sorties analogiques fournissent des signaux analogiques à partir des valeurs numériques transmises par la CPU en utilisant des CNA.

SM 374 = Est un module spécial qui offre à l'utilisateur la possibilité de tester son programme lors de la mise en service en cours de fonctionnement. Dans notre API S7- 300, ce module se trouve à la place d'un module d'entrée ou de sortie TOR.

III.2.8 Module de fonction (FM) :

Ces modules de fonction réduisent la charge de traitement de la CPU, Comme ils offrent aussi les fonctions spéciales telles que le comptage rapide, la régulation et le positionnement.

III.2.9 Processus de communication (CP) :

Les modules de communication permettent d'établir des liaisons homme-machine (point-point, PROFIBUS,) qui sont effectuées à l'aide des interfaces de communication.

III.3 Console de programmation (PG)

Pour programmer l'automate, l'automaticien peut utiliser : une console de programmation ayant pour avantage la portabilité.

Console d'exploitation : permet le paramétrage et les relevés d'informations (modification des valeurs et visualisation) sur site ; - Console de programmation, réglage et exploitation. Cette dernière effectue dans la phase de programmation l'écriture, la modification, l'effacement et le transfert d'un programme dans la mémoire de l'automate ou dans une mémoire REEPROM. Dans la phase de réglage et d'exploitation, elle permet d'exécuter le programme pas à pas, de le visualiser, de forcer ou de modifier des données telles que les entrées, les sorties, les bits internes, les registres de temporisation, les compteurs, ..., la sortie sur imprimante du programme en cas de présence d'un port de sortie.

La console comme il est montré sur la figure III-4, elle est généralement équipée d'un écran à cristaux liquides, peut afficher le résultat de l'auto-test comprenant l'état des modules d'E/S, l'état de la mémoire, de la batterie

Un PC avec lequel la programmation est plus conviviale, communiquant avec l'automate par le biais d'une liaison série RS232 ou RS485 ou d'un réseau de terrain.



Figure III-4 : Console de programmation PG [23]

III.4 Logiciel de programmation STEP7 (TIA Portal)

SIMATIC STEP7 est le logiciel de programmation le plus connu au monde et le plus utilisé dans le domaine de l'automatisation industrielle. Le STEP7 est le progiciel de base pour la configuration et la programmation de systèmes d'automatisation. Il fait partie de l'industrie logicielle SIMATIC. Le progiciel de base STEP7 existe en plusieurs versions.

Les tâches de bases qu'il offre à son utilisateur lors de la création d'une solution d'automatisation sont : [21]

- La création et gestion de projet.

- La configuration et le paramétrage du matériel et de la communication.
- La gestion des mnémoniques.
- La création des programmes.
- Le test de l'installation d'automatisation.

Il s'exécute sous les systèmes d'exploitation de Microsoft à partir de la version Windows 95. Par conséquent, il s'adapte à l'organisation graphique orientée objet qu'offrent ces systèmes d'exploitation [17] dans ce qui suit nous allons présenter un petit descriptif pour la prise en main du logiciel.

III.4.1 Description du logiciel TIA Portal

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

Les avantages du logiciel TIA portal

- Programmation intuitive et rapide : avec des éditeurs de programmation nouvellement développés.
- Efficacité accrue grâce aux innovations linguistiques de STEP 7 : programmation symbolique uniforme, Calculate Box, ajout de blocs durant le fonctionnement, et bien plus encore. Performance augmentée grâce à des fonctions intégrées : simulation avec PLCSIM, télémaintenance avec Téléservice et diagnostic système cohérent.
- Technologie flexible : Fonctionnalité motion control évolutive et efficace pour les automates S7-1500 et S7-1200.
- Sécurité accrue avec Security Integrated : Protection du savoir-faire, protection contre la copie, protection d'accès et protection contre la falsification.
- Environnement de configuration commun avec pupitres IHM et entraînements dans l'environnement d'ingénierie TIA Portal.

III.4.2 Vue du portal et vue du projet

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

- Vue du portal : elle est axée sur les tâches à exécuter et sa prise en main est très rapide.

- Vue du projet : elle comporte une arborescence avec les différents éléments du projet, les éditeurs requis s'ouvrent en fonction des tâches à réaliser. Données, paramètres et éditeurs peuvent être visualisés dans une seule et même vue [21]

III.4.2.1 Vue du portail :

Chaque portail permet de traiter une catégorie de tâche (action) la fenêtre affiche la liste la liste des actions peuvent être réalisées pour la tâche sélectionnée.



Figure III-5 : Vue du portales [21]

III.4.2.2 Vue du projet

L'élément « Projet » contient l'ensemble des éléments et des données nécessaires pour mettre en œuvre la solution d'automatisation souhaitée.

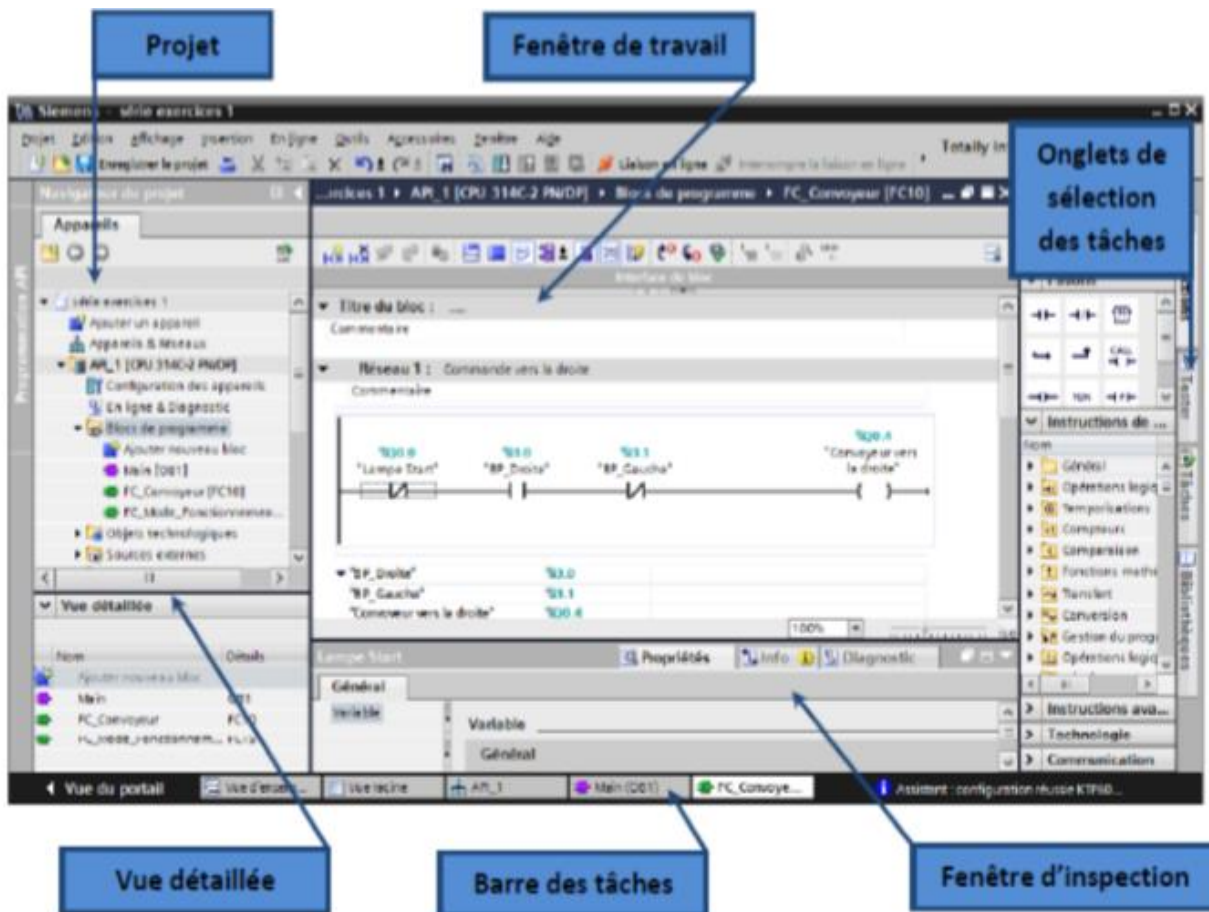


Figure III-6 : Vue du Projet [22]

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

La fenêtre d'inspection permet de visualiser des informations complémentaires sur un objet sélectionné ou sur les actions en cours d'exécution (propriété du matériel sélectionné, messages d'erreurs lors de la compilation des blocs de programme,).

Les onglets de sélection de tâches ont un contenu qui varie en fonction de l'objet sélectionné (configuration matérielle, bibliothèques des composants, bloc de programme, instructions de programmation).

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

III.4.3 Adressage des E/S

Pour connaître l'adressage des entrées et sorties présentes dans la configuration matérielle, il faut aller dans « Appareil et réseau » dans le navigateur du projet. Dans la fenêtre de travail, on doit s'assurer d'être dans l'onglet « Vue des appareils » et de sélectionner l'appareil voulu.

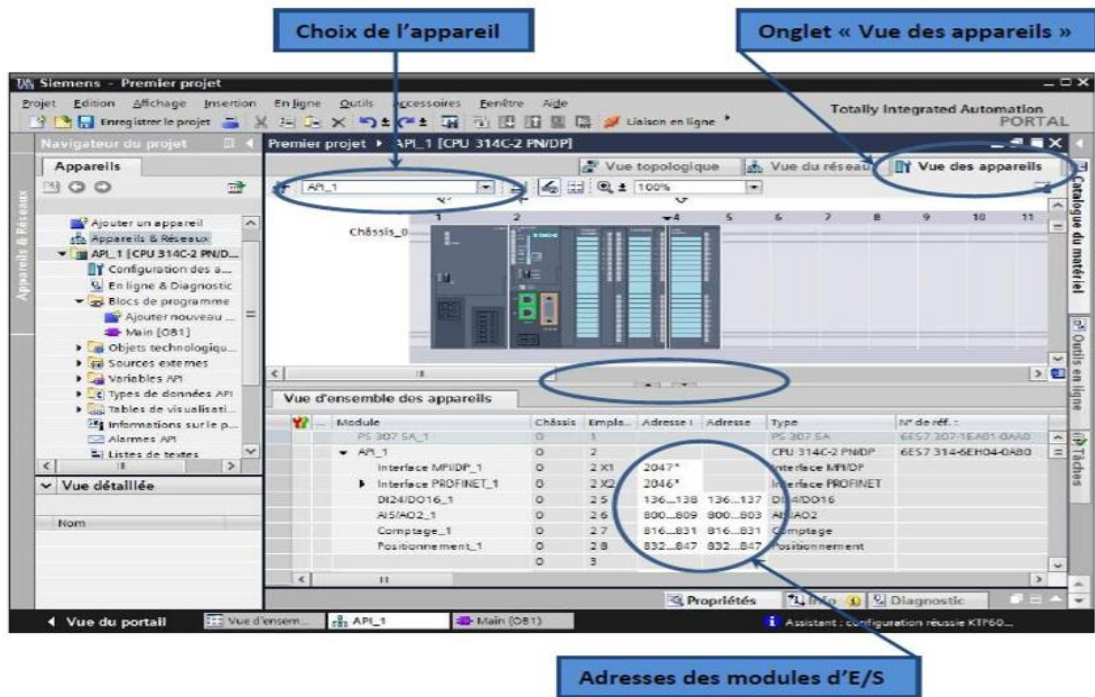


Figure III-7 : Adressage de la CPU [22]

On sélectionne la CPU puis à l'aide des deux petites flèches (voir figure III-7), on fait apparaître l'onglet « Vue d'ensembles des appareils ». Les adresses des entrées et sorties apparaissent. On peut les modifier en entrant une nouvelle valeur dans la case correspondante. [22]

III.4.4 Les variables API

Adresses symboliques 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.
- Adresse symbolique : correspond au nom que l'utilisateur a donné à la variable (ex : bouton marche). Le lien entre les adresses symbolique et absolue se fait dans la table des variables API. Lors de la programmation, on peut choisir d'afficher les adresses absolues, symboliques ou encore les deux simultanément.

III.4.4.1 Table des variables API

C'est dans la table des variables API 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 : On peut également insérer un commentaire qui nous renseigne sur cette variable. Le commentaire peut être visible dans chaque réseau utilisant cette variable.

III.4.4.1 Liaison avec l'automate

Il faut maintenant charger la configuration de l'automate dans celui-ci. Pour cela, il faut tout d'abord connecter l'automate au PC en utilisant l'interface SIMATIC S7 PC USB adapté. Ensuite, après avoir sélectionné la vue « En ligne et diagnostique », sélectionnez les options suivantes :

- Mode : MPI
- Interface PG /PC : pc Adapter

III.4.5 Adresse Ethernet de la CPU

Toujours dans les propriétés de la CPU, il est possible de définir son adresse Ethernet. Un double clic sur l'icône Ethernet de la station fait apparaître la fenêtre d'inspection permettant de définir ses propriétés.

Pour établir une liaison entre la CPU et la console de programmation, il faut affecter aux deux appareils des adresses appartenant au même sous réseau.

III.4.6 Programme utilisateur

La programmation structurée permet la rédaction claire et transparente de programmes. Elle permet la construction d'un programme complet à l'aide de modules qui peuvent être échangés et/ou modifiés à volonté. Pour permettre une programmation structurée confortable, il faut prévoir plusieurs types des modules : (OB) Bloc d'organisation, (FB) Bloc fonctionnel, (FC) Fonction, (SFB) Bloc fonctionnel système, (SFC) Fonction système, (DB) Bloc de données [18].

III.4.7 Structure de programme :

III.4.7.1 Programme linéaire :

Le langage de programmation en contact rappelle la logique des contacts à relais câblés que les automates programmables ont remplacé, la CPU traite successivement les différentes opérations.

III.4.7.2 Programme segmenté :

Le programme se subdivisé en blocs, chacun d'entre eux contenant uniquement le programme nécessaire pour réaliser la tâche qui lui est assignée. Un bloc peut à son tour se subdiviser en plusieurs réseaux. Lorsque l'on utilise des réseaux similaires, on peut créer des modèles. Le bloc d'organisation OB1 est le bloc principal dans l'exécution des programmes dans step7 il contient des opérations sur bits et fait l'appelle à d'autres blocs dans un ordre défini. [21]

III.4.8 Les Blocs

Le système d'automatisation utilise différents types de blocs dans lesquels peuvent être mémorisés le programme utilisateur et les données correspondantes. Selon les exigences du processus, le programme peut être structuré en différents blocs :

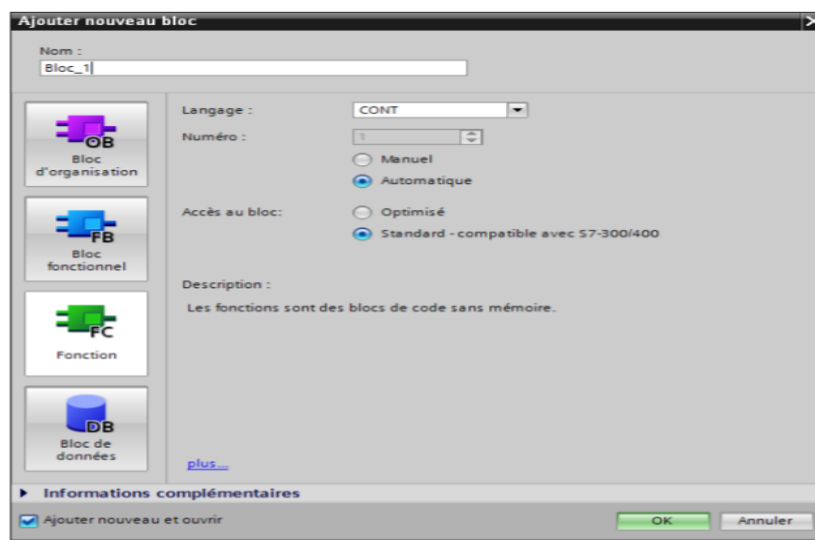


Figure III-8 : Les différents types de bloc [21]

III.4.8.1 Bloc d'organisation :

Les blocs d'organisation (OB) constituent l'interface entre le système OB d'exploitation et le programme utilisateur. L'ensemble du programme peut être concaténé dans un seul bloc

OB1 (programme linéaire) appelé de manière cyclique par le système d'exploitation, ou être structuré dans plusieurs blocs (programme structuré) [18].

III.4.8.2 Bloc Fonction :

Une fonction (FC) assure une fonctionnalité spécifique du programme. Les fonctions peuvent être paramétrables. Dans ce cas, des paramètres sont transmis à la fonction lorsqu'elle est appelée. Les fonctions conviennent pour la programmation de fonctionnalités récurrentes et complexes (e.g., effectuer des calculs).

Les fonctions système (SFC) sont des fonctions paramétrables, intégrées au système d'exploitation de la CPU, dont le numéro et la fonctionnalité sont définis de manière fixe. Pour de plus amples informations, se reporter à l'aide en ligne [19].

III.4.8.3 Bloc fonctionnel :

Du point de vue du programme, les blocs fonctionnels s'apparentent aux fonctions FB et SFB, mais ils disposent en plus de zones mémoires spécifiques, sous forme de blocs de données d'instance. Les blocs fonctionnels conviennent pour la programmation de fonctionnalités récurrentes encore plus complexes (e.g., assurer des tâches de régulation). [20]

III.4.8.4 Blocs fonctionnels système (SFB) :

Ce sont des blocs fonctionnels paramétrables, intégrés au système d'exploitation de la CPU, dont le numéro et la fonctionnalité sont définis de manière fixe. Pour de plus amples informations, se reporter à l'aide en ligne.

III.4.8.5 Blocs de données :

Les blocs de données (DB) sont des zones de données du programme DB utilisateur, mise à la disposition d'espace mémoire pour les variables de types données.

Il existe deux types blocs de données :

- **Bloc de données globales** : Contrairement aux blocs de code, les blocs de données ne contiennent pas d'instruction STEP7. Ils servent à l'enregistrement des données du programme utilisateur pouvant être utilisées par tous les autres blocs. Ils contiennent des données variables que le programme utilisateur utilise [20] ;

- **Bloc de données d'instance** : Un bloc de donnée d'instance est associé à chaque appel de bloc fonctionnel transmettant des paramètres. Ces blocs contiennent les paramètres

effectifs et les données statiques du FB. Les variables déclarées dans le FB déterminent la structure du bloc de donnée d'instance. L'instance est l'appel d'un bloc fonctionnel.

III.4.9 Mémentos

Des mémentos sont utilisés pour le fonctionnement interne de l'automate pour lesquelles l'émission d'un signal n'est pas nécessaire. Les mémentos sont des éléments électroniques bistables servent à mémoriser les états logiques '0' et '1'. Chaque automate programmable dispose d'une grande quantité de mémentos. On programme ces derniers comme des sorties.

III.4.10 Mnémoniques

Les mnémoniques sont les noms que l'on attribue aux variables globales de l'API. L'emploi des mnémoniques à la place des adresses absolues améliore considérablement la lisibilité et la clarté d'un programme et aide à isoler des défauts éventuels.

Les mnémoniques ainsi définies sont utilisables dans tout le programme utilisateur d'un module programmable [17].

III.4.11 Les différents types de variables

Dans l'environnement de STEP 7, on utilise plusieurs types de variables qu'on doit déclarer au préalable. Le tableau ci-dessous résume les types de variables utilisés.

Groupe	Types de données	Signification
Type de données binaires	BOOL BYTE WORD DWORD	Les données de ce type occupent 1 bit, 8bits, 16bits ou 32bits.
Type de données sur caractère	CHAR	Les données de ce type occupent Du jeu de caractère ASCII.
Type de données numérique	INT DINT REAL	Les données de ce type permettent de traiter des valeurs numériques.
Type de données temporelles	TIME DATE TIME OF-DAY S5TIME	Les données de ce type représentent les diverses valeurs de durée et de date dans le STEP 7.

Tableau III.1 : Différents types de variables contenues dans le STEP7 [18].

III.5 Langages de programmation :

Chaque automate possède son propre langage. Mais cependant les constructeurs proposent tous une interface logicielle répondant à la norme CEI 61131-3 (Commission Electrotechnique Internationale). Cette norme définit cinq langages de programmation utilisables, [11] qui sont :

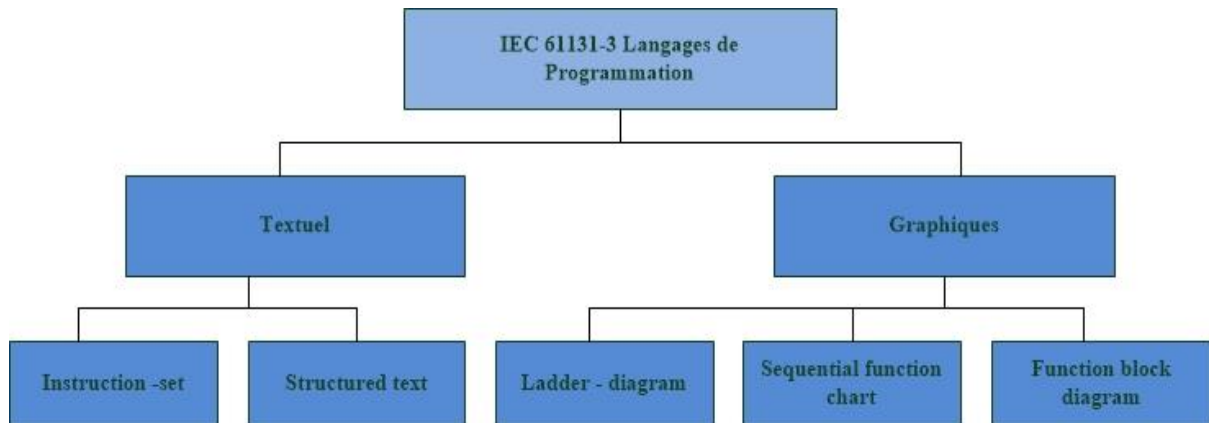


Figure III-9 : Classification des langages de programmation

III.5.1 GRAFCET ou SFC

Le langage de programmation graphique optionnel GRAPH vous permet de programmer des commandes séquentielles. Ceci implique la création d'une succession d'étapes, la définition des actions associées à chaque étape et celle des transitions indiquant les possibilités d'évolution entre deux étapes successives. Pour définir les actions associées aux étapes, vous utilisez un langage de programmation spécial (similaire à LIST), alors que pour déterminer les conditions de réceptivité des transitions, vous utilisez une représentation sous forme de schéma à contacts (langage de programmation CONT restreint). Il permet la représentation très claire de séquences même complexes, ce qui favorise une programmation et une recherche d'erreurs efficaces

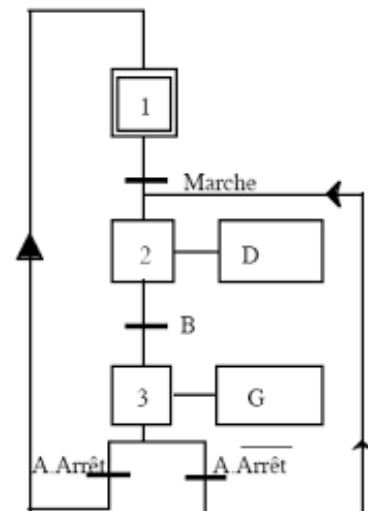


Figure III-10 : Exemple d'un langage SFC

III.5.2 Schéma par blocs ou FBD

Le programme est un langage de programmation graphique qui utilise des boîtes logiques. Le signe placé dans la boîte symbolise la fonction (par exemple & opération logique ET). Ce mode de représentation permet à des non-programmeurs, par exemple des spécialistes des procédés industriels, d'accéder à la programmation.

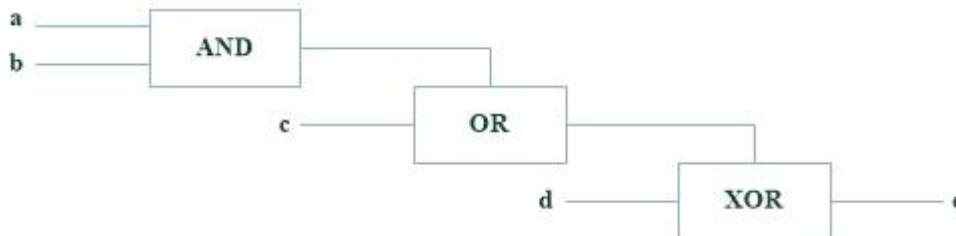


Figure III-11 : Exemple d'un langage FBD

III.5.3 Schéma à contact ou CONT :

Le schéma à contacts s'apparente aux schémas de circuits électriques, il fait appel à des symboles, tels que les contacts et bobines. Ce mode de représentation convient particulièrement à tous les familiers de technologie des contacteurs.

En milieu industriel, ce type de langage permet une adoption facile d'un automate programmable par des utilisateurs de tout niveau, en particulier par des techniciens d'entretien comme il facilite les opérations de maintenance.

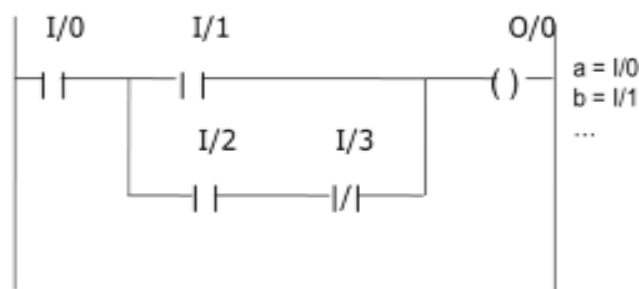


Figure III-12 : Exemple d'un langage CONT

Structure d'un programme LADDER

Dessiné entre deux barres de potentiel, un ensemble de lignes (réseaux). Un réseau est un ensemble d'éléments graphiques reliés entre eux par des connexions horizontales et verticales, ces éléments représentent

- Les entrées/sorties de l'automate (boutons-poussoirs, détecteurs, relais...).
- Des fonctions d'automatismes (temporisateurs, compteurs...).
- Des opérations arithmétiques et logiques et des opérations de transfert.
- Les variables internes de l'automate

Un programme en langage RELAIS est une suite de lignes d'instructions. Chaque ligne d'instructions est composée d'une ou plusieurs actions, et le plus souvent d'une ou plusieurs conditions ; et chaque ligne d'instructions est composée d'une ou plusieurs actions, et le plus souvent d'une ou plusieurs conditions.

III.5.4 Texte structuré ou ST

Ce langage est un langage textuel de haut niveau comme C, Pascal. Il permet la programmation de tout type d'algorithme plus ou moins complexe.

```

1 Sub Read_data_from_file()
2 Dim fso, f, ts, field, MyZf
3
4 ' Catch errors -- Fehler abfangen
5 On Error Resume Next
6
7 ' Create file object -- Objekt erstellen
8 Set fso = CreateObject("Scripting.FileSystemObject")
9 If Err.Number <> 0 Then
10     ShowSystemAlarm "Error #" & CStr(Err.Number) & " " & Err.Description
11     Err.Clear
12     Exit Sub
13 End If
14
15 Set f = fso.OpenTextFile(fileName, 1, 0, -2)
16 If Err.Number <> 0 Then
17     ShowSystemAlarm "Error #" & CStr(Err.Number) & " " & Err.Description
18     Err.Clear
19     Exit Sub
20 End If
21
22 ' Read values till end of file -- Werte bis zum Dateiende einlesen
23 Do While Not f.AtEndOfStream
24     MyZf = f.ReadLine
25 Loop
26 field = Split(MyZf, ";")
27
28 MaterialFromFile = field(0)
29 PressureFromFile = field(1)
30 TemperatureFromFile = field(2)
31
32 ' Tidy up -- Aufräumen
33 f.Close
34 Set f = Nothing
35 Set fso = Nothing
36 ShowSystemAlarm "Readout of the data was successful!"
37
38 End Sub

```

Figure III-13 : Exemple d'un langage ST [12]

III.5.5 Liste d'instructions ou IL

Langage textuel de même nature que l'assembleur (programmation des microcontrôleurs). Très peu utilisé par les automaticiens

La liste d'instructions se compose d'opérations STEP7. List permet de programmer librement (et parfois de manière complexe). Ce mode de représentation est plus particulièrement destiné aux programmeurs qui maîtrisent déjà d'autres langages de programmation. [12]

Network 1:				RLO	Value
Comment					
1	A	"Motor_1_Enabled"	1	1	
2	AN	"Motor_1_EmergencyStop"	0	1	
3	JC	n_OK	1		
4	=	"Motor_1_Start"	1	1	
5	AN	"Motor_1_SpeedOK"	0	1	
6	AN	"Motor_1_BreakesEnabled"	0	0	
7	=	"Motor_1_Stop"	0	0	
8	JU	End			
9	n_OK: SET				
10	AN	"Motor_1_BreakesEnabled"			
11	=	"Motor_1_Stop"			
12	End: NOP	0	0		

Figure III-14 : Exemple d'un langage IL [12]

III.6 Conclusion :

Dans ce chapitre, on a présenté l'automate S7-300 de SIEMENS ainsi que le logiciel qui lui est associé STEP7 TIA PORTAL. Puis on a montré la configuration matérielle ; et citer les différentes langages de programmation des API existants.

Chapitre IV

Programmation et Simulation de la Station de Lavage

IV.1 Introduction

De nombreux automatismes existent dans les systèmes et les objets techniques qui constituent notre environnement, il faut remarquer que les niveaux de complexité sont très divers.

L'automatisation d'une station de lavage permet de minimiser l'intervention humaine et préserver notre environnement par le contrôle de la consommation et éviter le gaspillage de l'eau.

Dans ce dernier chapitre, on va présenter le cahier des charges et citer les différents constituants de la station de lavage ; la deuxième partie sera consacré à la programmation en utilisant le logiciel TIA PORTAL qui s'effectuera par le langage LADDER, suit à cela nous présentons la simulation effectuée.

IV.2 Schéma de principe de la station de lavage

La figure IV-1 représente la station de lavage de voitures qu'on veut étudier. On peut voir les différents constituants montés sur le portique mobile.

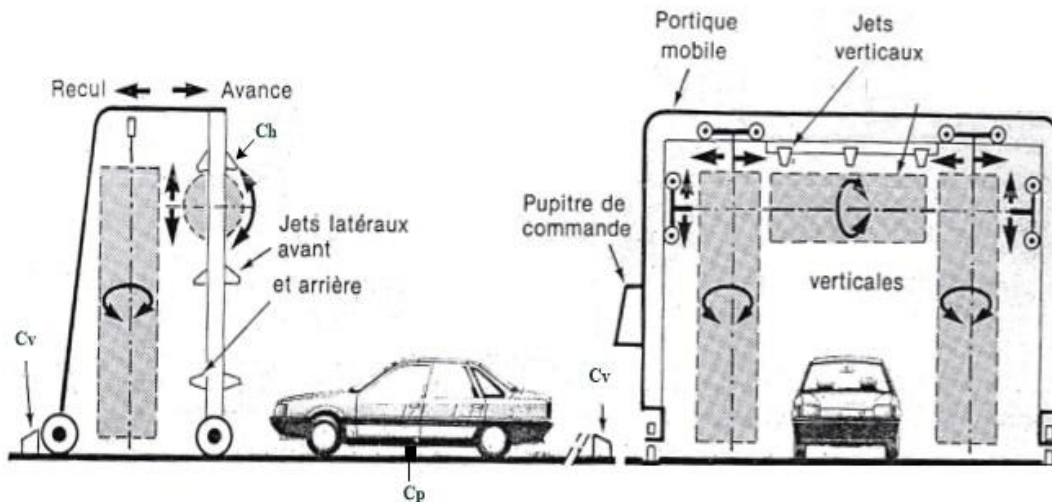


Figure IV-1 : Organisation d'une station automatisée de lavage de voiture [03].

Dans le schéma qui suit (fig IV-1) nous présentons l'actigramme appropriée de la station de lavage qui permet de définir la fonction globale du système. Cette figure montre la fonction globale de la station de lavage ainsi que la matière d'œuvre et les données de contrôle.

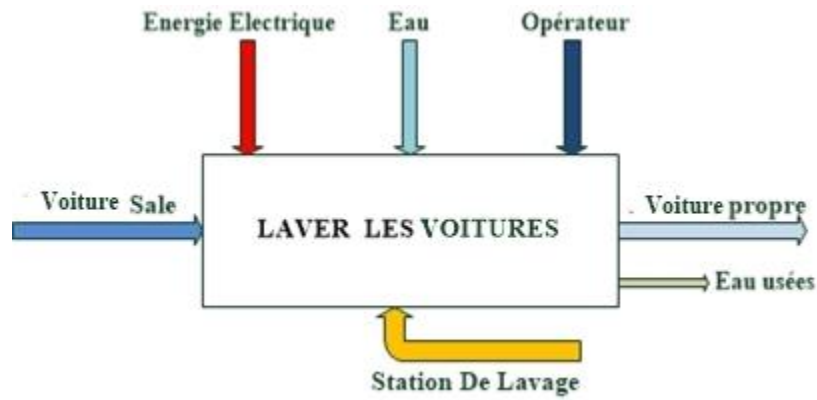


Figure IV-2 : Actigramme de la station de lavage.

Sur la Figure IV-3, on peut voir la décomposition de ce système automatisé en PC et PO, il met en évidence les ordres et le compte rendus de ces deux parties.

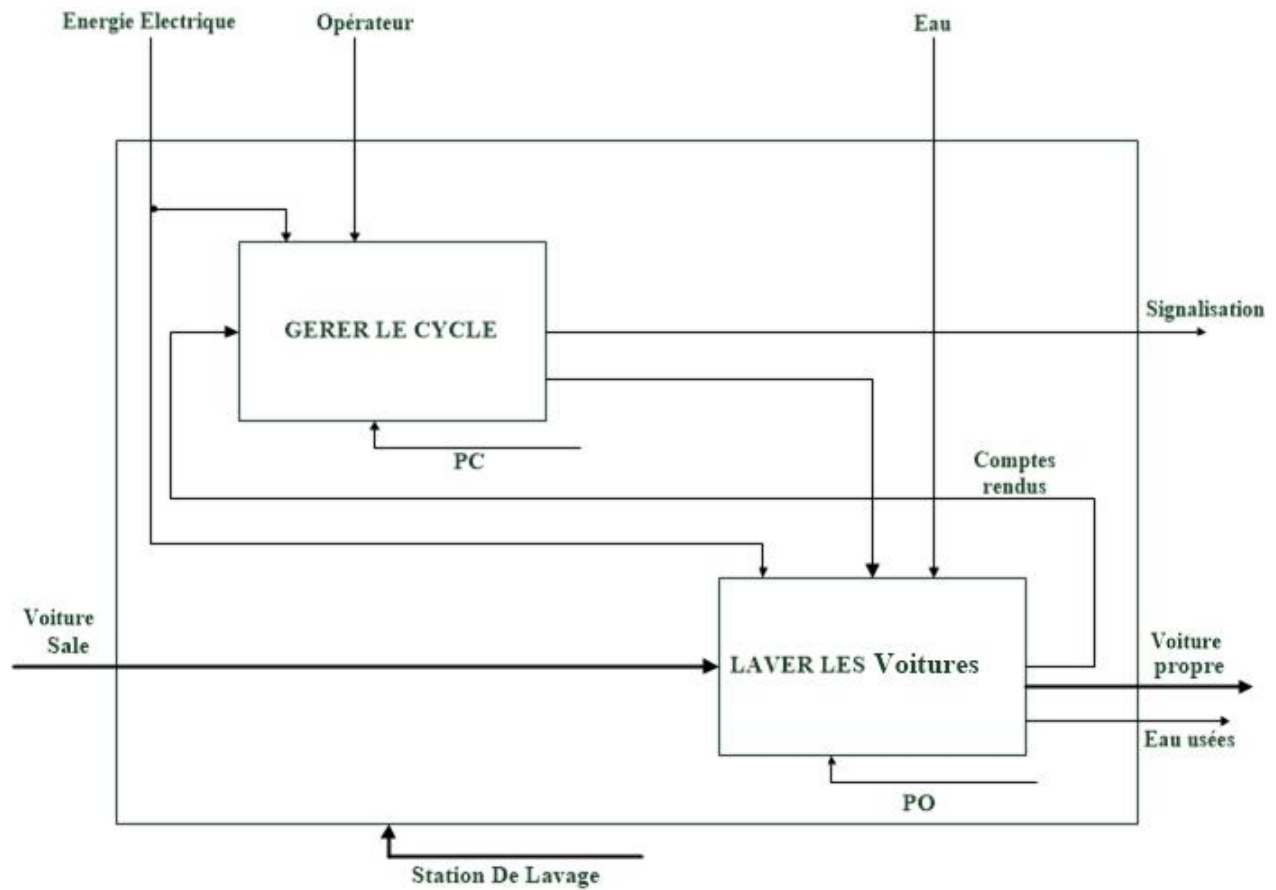


Figure IV-3 : Processus de la station de lavage [2].

IV.3 Présentation de la partie opérative :

Dans ce système, la partie opérative comporte essentiellement ce qui suit :

- 1- Un portique de lavage se déplace sur des rails entre deux butées avant et arrière afin d'assurer le nettoyage du véhicule présent dans la station ; il supporte deux rouleaux verticaux et un rouleau horizontal, entraîné par un moteur électrique triphasé de puissance 3kW à deux sens de rotation ;
- 2- Un moteur électrique triphasé de puissance 4kW associé à un mécanisme permettant la rotation de ces trois rouleaux ;
- 3- Un moteur électrique triphasé de puissance 3kW pour la montée et la descente du rouleau horizontal ;
- 4- Une pompe à eau gérée par un moteur triphasé de puissance 0.5 kW ; et deux électrovannes permettant d'arroser le véhicule à travers des buses d'injection de l'eau ou de l'eau savonné et le rinçage ;
- 5- Les trois moteurs sont protégés par des disjoncteurs magnétothermiques, et commandés par des contacteurs de puissance ;
- 6- Un dispositif de séchage placé sur la partie haute du portique (non présenté dans les figures IV-4 et IV-5) ;

Remarque : tous les moteurs utilisés ont les caractéristiques suivantes :

- Tension : 380V,
- Mode de branchement : 3Ph,
- Fréquence : 50Hz.

On peut voir sur la figure IV.4 le schéma électrique de ce système.

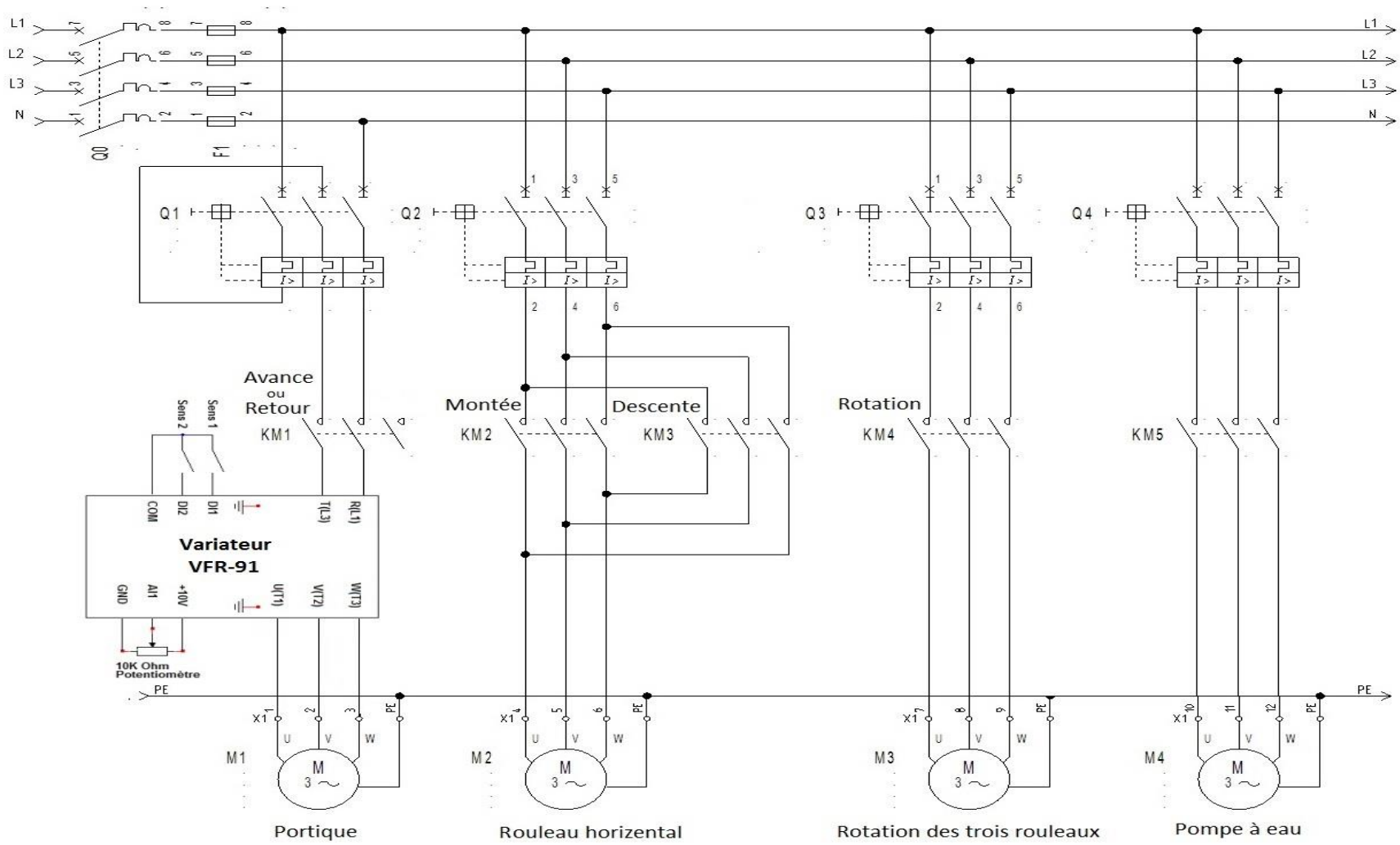


Figure IV-4 : Schéma de puissance de la station de lavage.

Description du tableau :

Dans les figures IV-4 et IV-5 sont représenté les différents actionneurs et préactionneurs de notre système. Le moteur du portique M1 est commandé par un contacteur KM1 et le convertisseur de fréquence VFR-91 va assurer le sens de rotation (le mouvement d'avancement et le recul du portique) ;

Le moteur du rouleau horizontal M2 est commandé par deux contacteur KM2 et KM3 afin de garantir le double sens de rotation (la montée et la descente) ;

Le moteur M3 est responsable de la rotation des trois rouleaux est commandé par le contacteur KM4 ;

Le moteur M4 fait fonctionner la pompe à eau à travers le pré actionneur KM5.

Comme il a été mentionné auparavant, les quatre moteurs sont protégés par des disjoncteurs magnétothermiques (Q1, Q2, Q3, Q4).

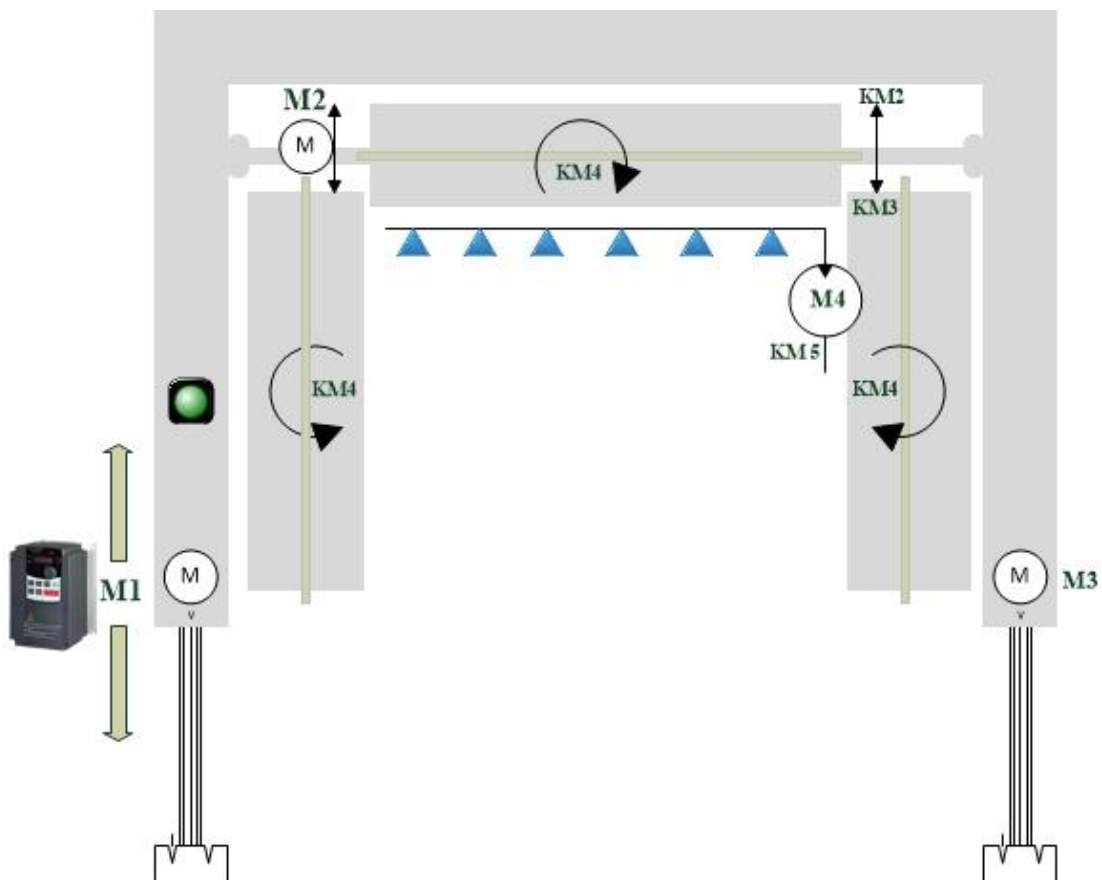


Figure IV-5 : Schéma de principe de la station de lavage (point de vue commande)

Actions	Pré actionneurs	Actionneurs
Descente rouleau horizontal	KM3	M1
Montée rouleau horizontal	KM2	M1
Rotation trois rouleaux	KM4	M2
Avance portique	KM1	M3
Reculé portique	KM1	M3
Prélavage	KEV1 et KM5	Vanne1 et pompe
Savonnage	KEV2 et KM5	Vanne2 et pompe
Rinçage	KEV1 et KM5	Vanne1 et pompe

Tableau IV-1 : Tableau récapitulatif des différents pré-actionneurs et actionneurs utilisés dans la station de lavage.

Lors du déplacement du portique pendant le lavage :

- On détecte les positions avant et arrière du portique à l'aide de capteurs type inductifs ou photoélectriques Cv et Cr.
- On détecte la présence de la voiture à l'aide de capteur type photoélectrique Cp.
- On détecte la position haute du rouleau horizontal à l'aide de capteur type photoélectrique Ch.

Capteurs	Désignations
Capteur présence de véhicule	Cp
Capteur avant portique	Cv
Capteur arrière portique	Cr
Capteur de position haute du rouleau horizontal	Ch

Tableau IV-2 : Capteurs de station de lavage.

IV.4 Présentation de la partie commande :

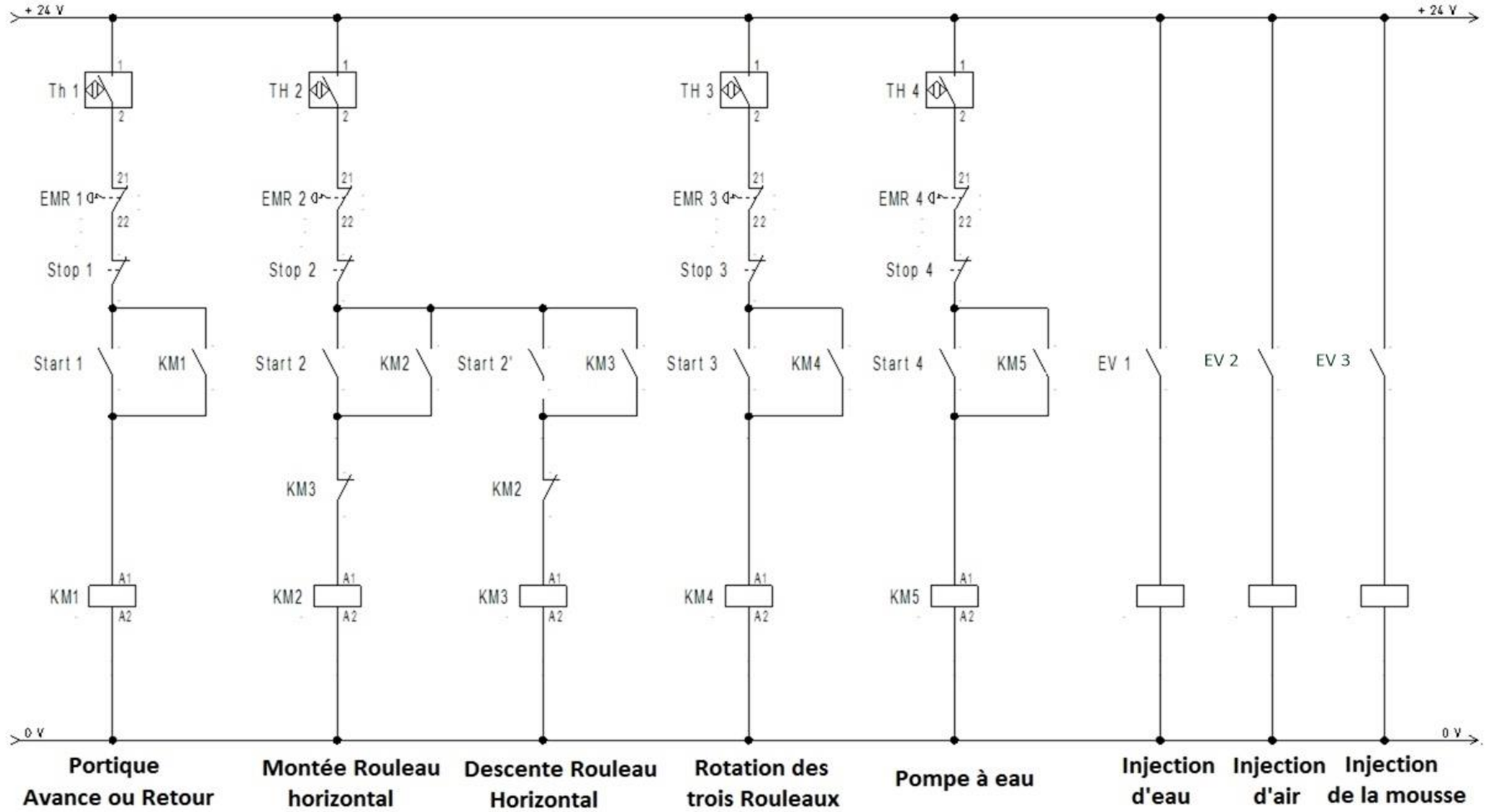


Figure IV-6 : Schéma de commande de la station de lavage.

Premièrement, pour la partie commande on a choisi l'alimentation qui va générer l'énergie pour l'API, puis comme deuxième étape on a choisi la CPU 315-2 PN/DP selon une estimation des besoins. Enfin, on a inséré les cartes d'interfaces E/S qui ont 32 entrées/sorties digitales chacune.

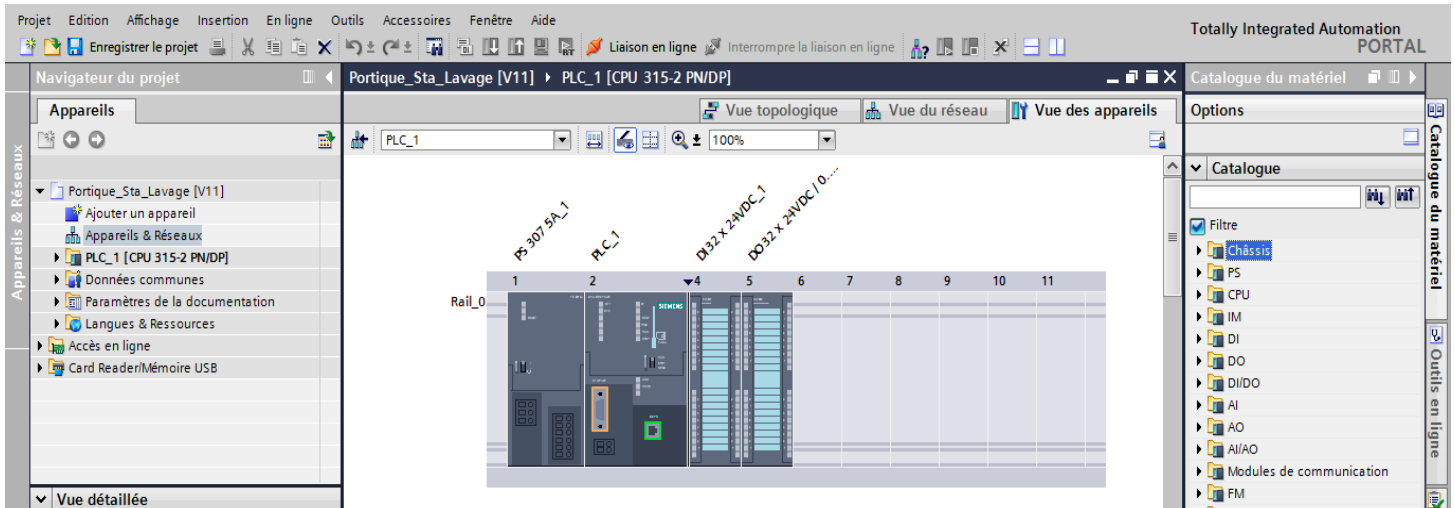


Figure IV-7 : Configuration matériels du système sur le logiciel TIA PORTAL.

IV.4.1 Les entrée/ sorties du programme

Les entrées :

- %I0.0 Bouton poussoir de départ cycle
- %I0.1 Bouton poussoir Arrêt urgence
- %I0.2 Position Avant Portique
- %I0.3 Position Arrière-Portique
- %I0.4 Présence de véhicule
- %I0.5 Position Haute de rouleau

Les Sorties :

- %Q4.0 Voyant Vert

- %Q4.1 Décente Rouleau Horizontal
- %Q4.2 Montée Rouleau Horizontal
- %Q4.3 Rotation des trois rouleaux
- %Q4.4 Avancer Portique
- %Q4.5 Retour Portique
- %Q4.6 Séchage véhicule
- %Q4.7 Prélavage
- %Q5.0 Savonnage
- %Q5.1 Rinçage

IV.4.2 La table des mnémoniques

Sur la figure IV.6, on peut voir une partie de la table mnémonique que nous avons renseigné et qui contient non seulement les entrées/sorties mais aussi les mémotos et les timers utilisés. Dans l'annexe 3 on trouve la mnémonique complète.

Nom	Type de données	Adresse	Réma...	Visibl...	Acces...	Comme...
7 -> KAV	Bool	%Q4.0		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Voyent_
8 -> KM1	Default tag table	Bool	%Q4.1	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Décente
9 -> KM2	Default tag table	Bool	%Q4.2	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Montée
10 -> KM3	Default tag table	Bool	%Q4.3	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Rotation
11 -> KM4	Default tag table	Bool	%Q4.4	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Avance...
12 -> KM5	Default tag table	Bool	%Q4.5	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Retour
13 -> KM6	Default tag table	Bool	%Q4.6	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Sécha...
14 -> KEV1&KMP	Default tag table	Bool	%Q4.7	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Prélav...
15 -> KEV2&KMP	Default tag table	Bool	%Q5.0	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Savonn...
16 -> KEV&KMP	Default tag table	Bool	%Q5.1	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Rinçage
17 -> Step_0	Default tag table	Bool	%M0.0	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Etape N
18 -> Step_1	Default tag table	Bool	%M0.1	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Etape N
19 -> Step_2	Default tag table	Bool	%M0.2	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Etape N
20 -> Step_3	Default tag table	Bool	%M0.3	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Etape N
21 -> Step_4	Default tag table	Bool	%M0.4	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Etape N
22 -> Step_5	Default tag table	Bool	%M0.5	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Etape N
23 -> Step_6	Default tag table	Bool	%M0.6	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Etape N
24 -> Step_7	Default tag table	Bool	%M0.7	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Etape N
25 -> Step_8	Default tag table	Bool	%M1.0	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Etape N
26 -> MER	Default tag table	Bool	%I0.6	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Mise_er
27 -> T1	Default tag table	Bool	%M2.0	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Fin_tem
28 -> T2	Default tag table	Bool	%M2.1	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Fin_tem
29 -> <Ajouter>				<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	

Figure IV-8 : Table de mnémonique du système dans le logiciel TIA PORTAL.

IV.5 Simulation du système de lavage

Les figures qui suivent, montrent les différentes étapes de la simulation de notre système. Cette opération nécessite le chargement du programme LADDER que nous avons écrit et élaboré. Ce chargement s'effectue dans la CPU virtuelle du PLCSIM qui est une partie du logiciel TIA PORTAL, afin de pouvoir tester les E/S du programme et suivre l'évolution et/ou le déroulement séquentiel des événements de la station ;

La figure IV-9 montre que la CPU et les modules d'entrées sorties sont bien connectés et que l'API est en ligne, ce qui nous permet de lancer la simulation à travers le PLCSIM.

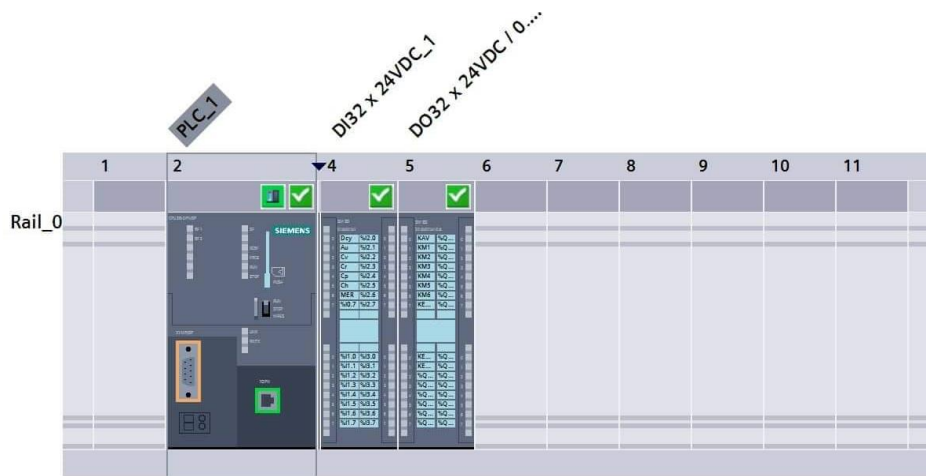
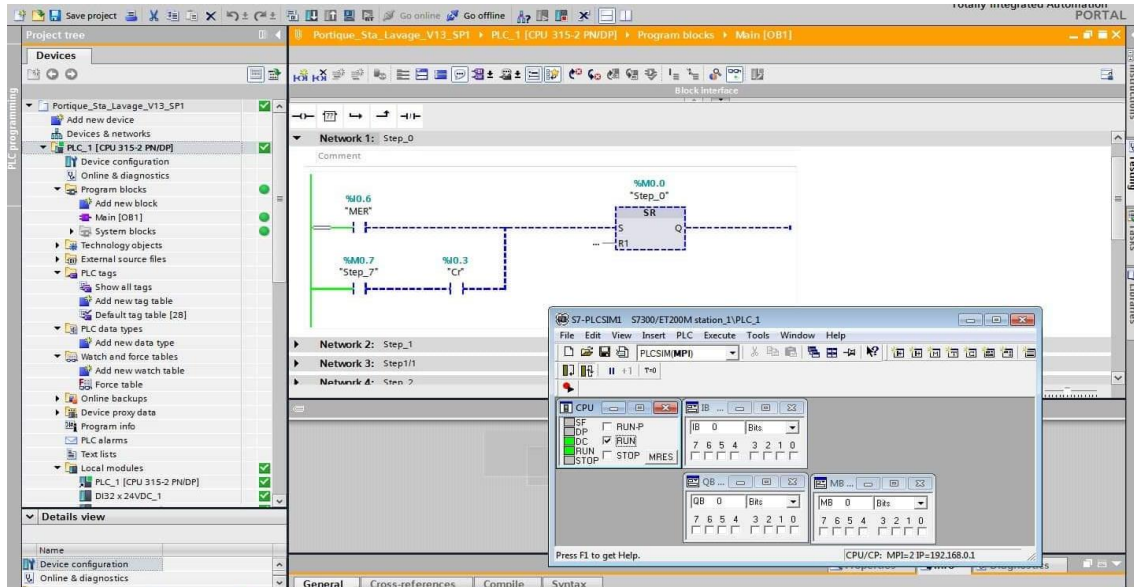
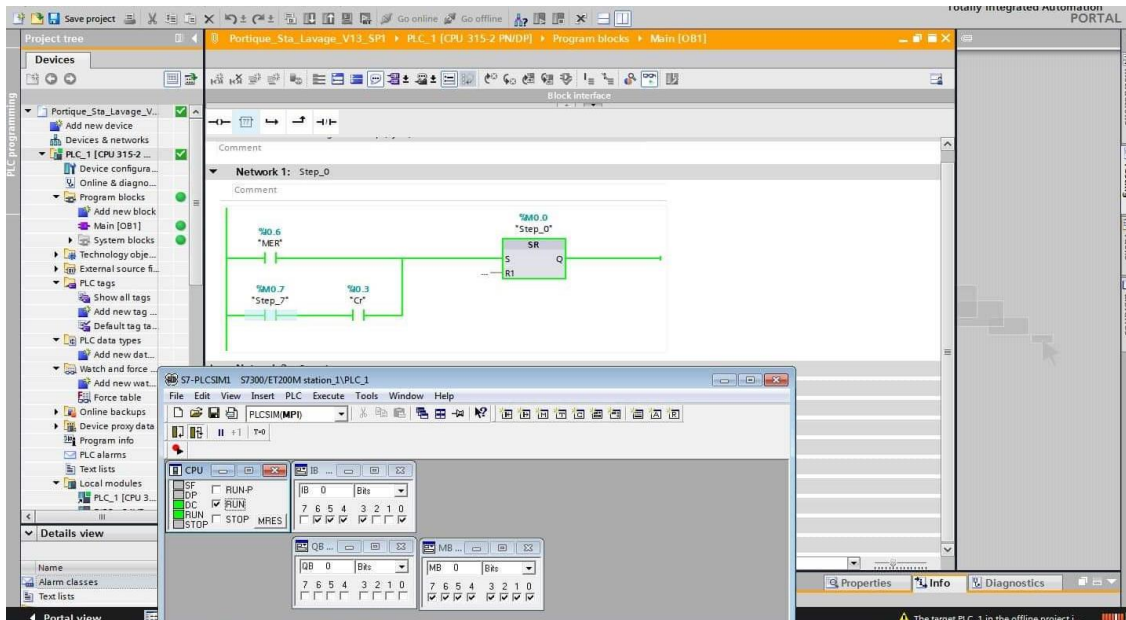


Figure IV. 9 : CPU et modules d'entrées/sorties en ligne.

Etape 01 : Cette étape représente l'état initial de la station de lavage pour lequel l'état du système doit être après chaque fin de cycle ; il faut que le portique soit en position arrière détecté par le capteur Cr ; le rouleau horizontal est en position haute détecté par le capteur Ch. et la véhicule doit être présente et détecté par le capteur Cp.



(a)

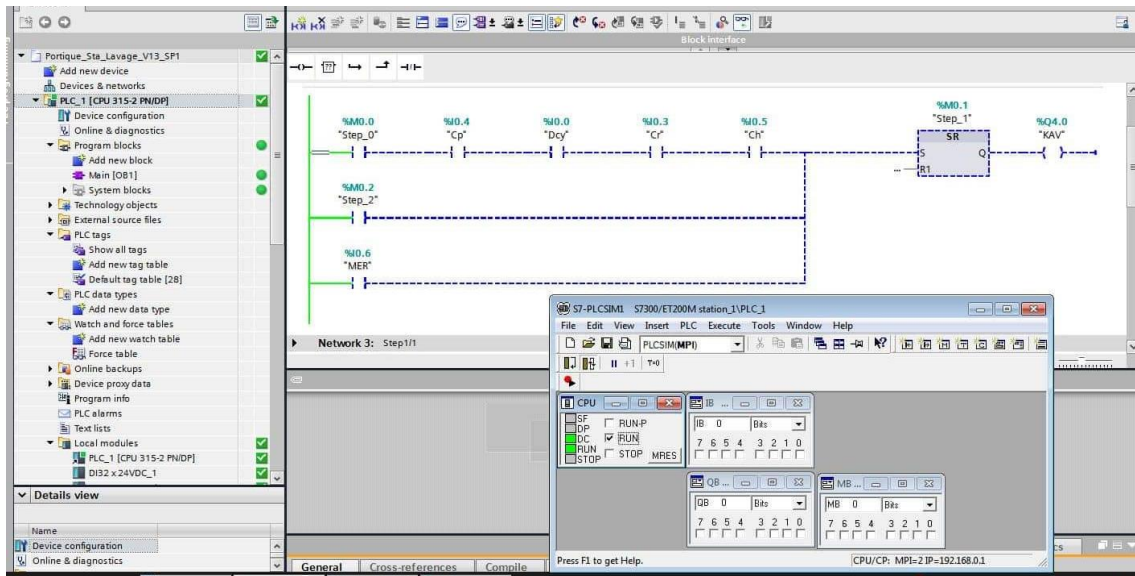


(b)

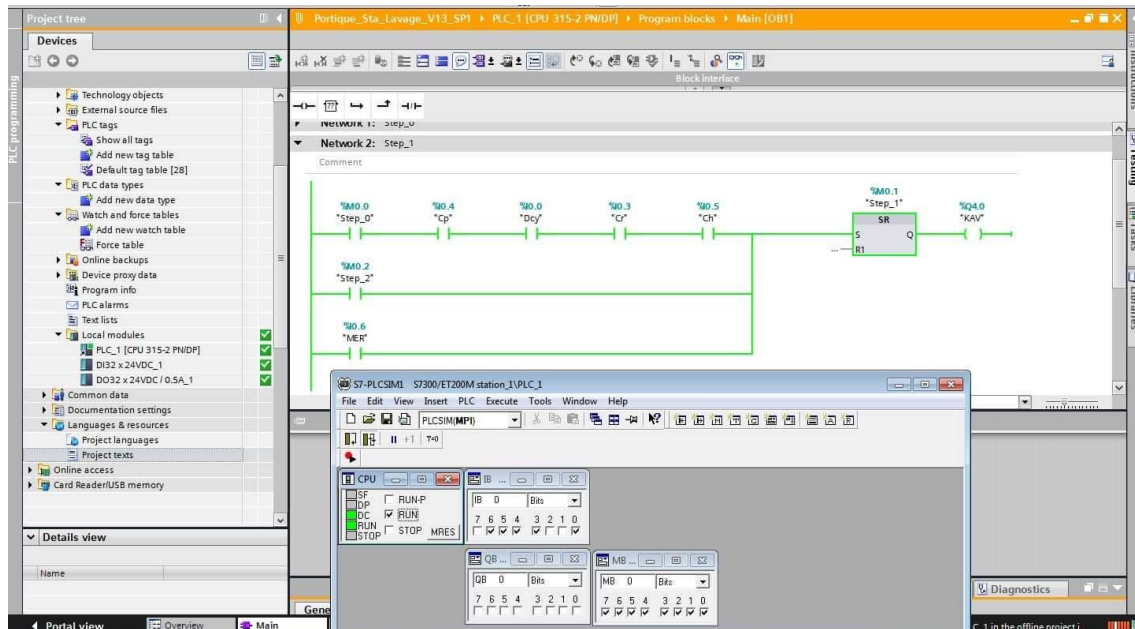
Figure IV.10 : Etat initial du système.

(a) Etat Off ; (b) Etat On.

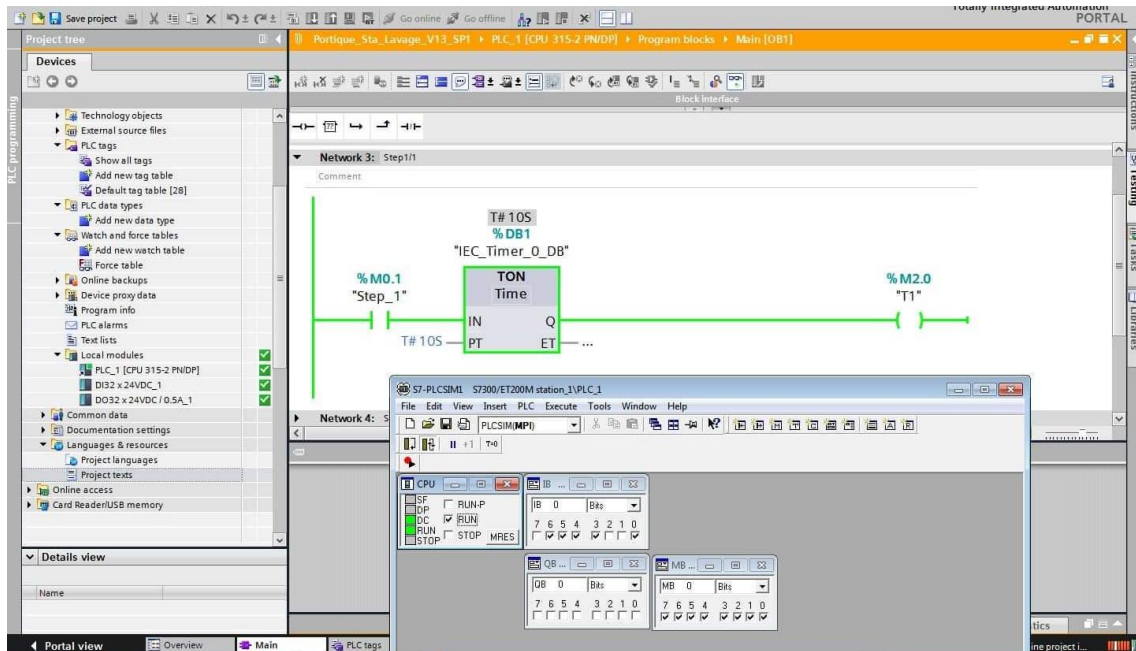
Etape 2 : Si les conditions initiales sont vérifiées, on appuie sur le bouton poussoir DCY (le départ de cycle) ; un voyant vert s'allume pendant 10 secondes indiquant que le cycle a démarré.



(a)



(b)

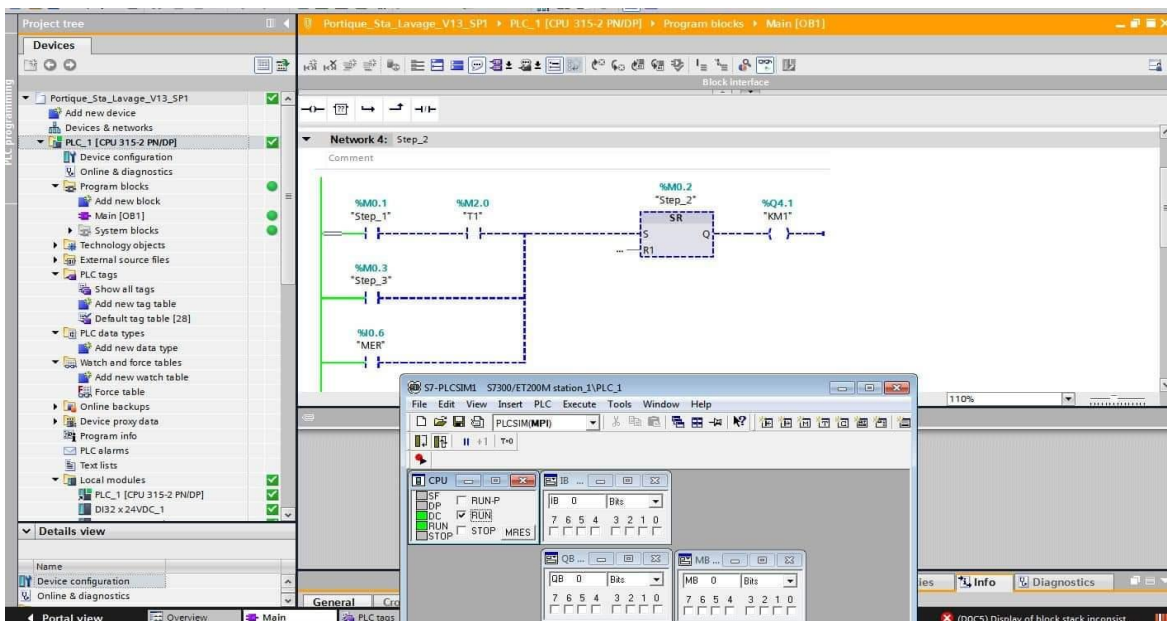


(c)

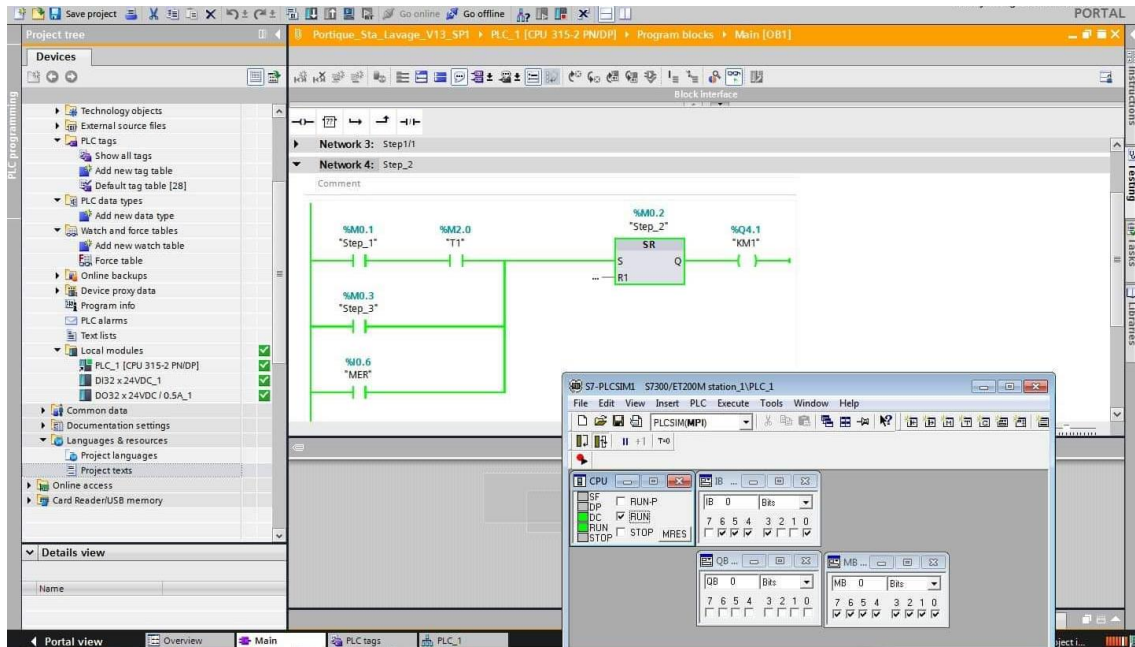
Figure IV-10 : Démarrage du cycle du système de lavage

(a) Etat Off ; (b) Etat On ; (c) Start-up du 1^{er} temporisateur

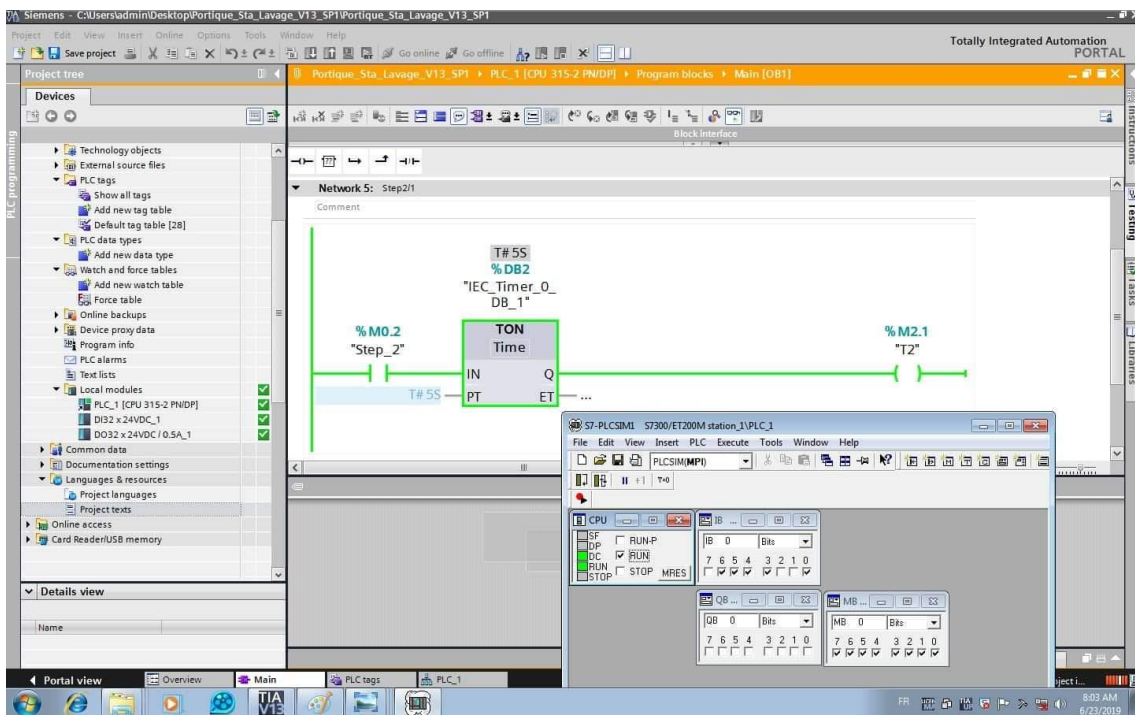
Etape 3 : Lorsque les 10 secondes se sont écoulé, le rouleau horizontal commence à descendre pendant une durée de 5 secondes.



(a)



(b)

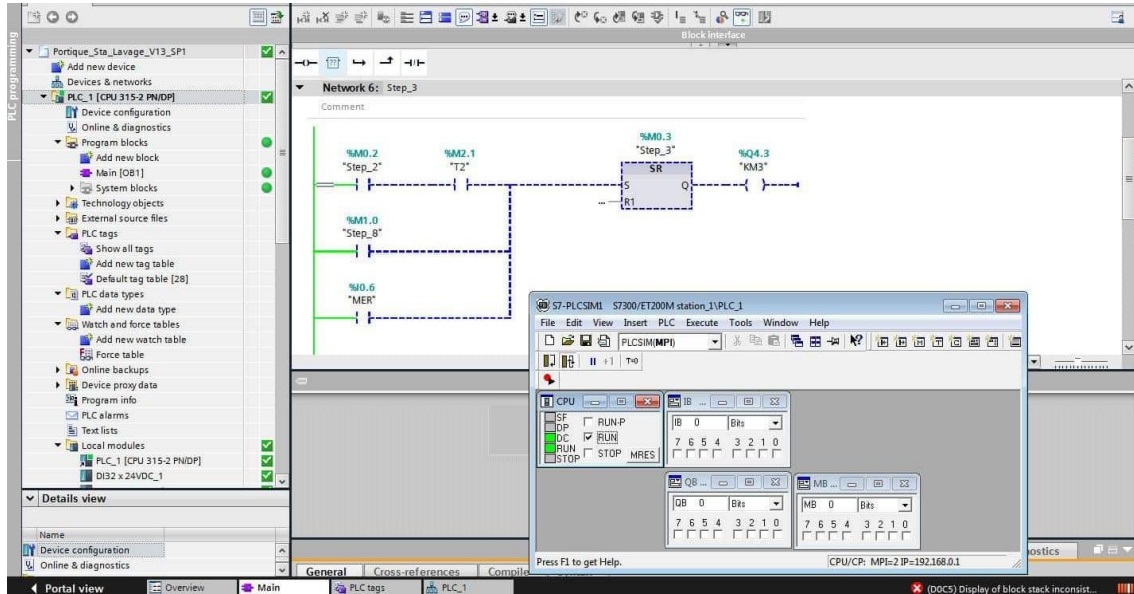


(c)

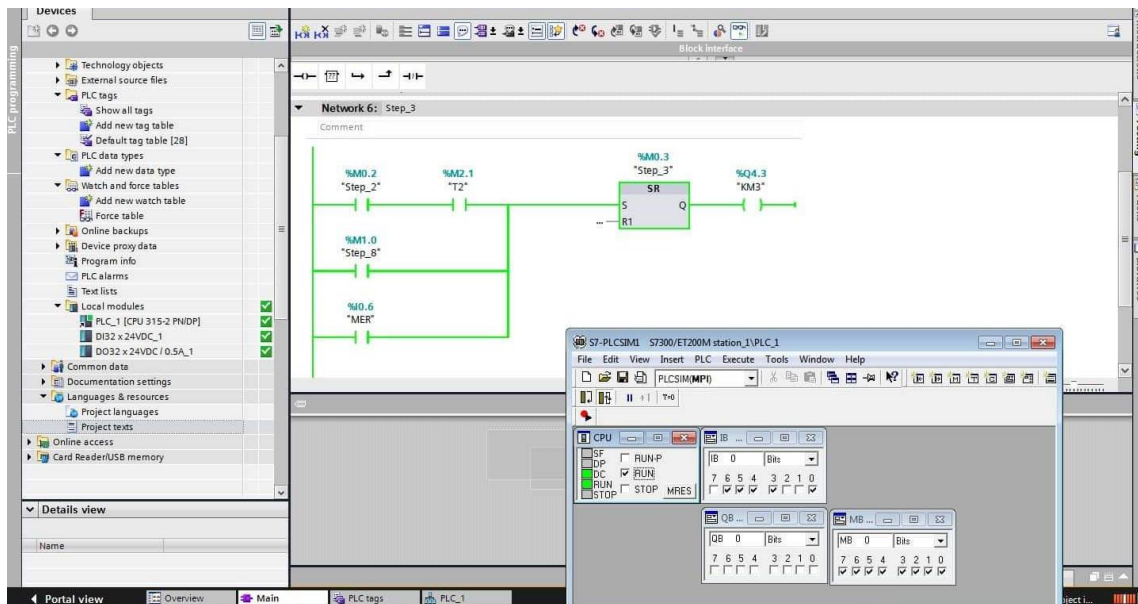
Figure IV.11. : Sortie de rouleau horizontal

État Off de la sortie KM1(a), Etat Off de la sortie KM1(b), (c) start-up du 2^{ème} temporisateur

Etape 4 : Au même moment et simultanément, le rouleau horizontal commencera à descendre ; la mise en rotation des trois rouleaux débutera aussi, et cette action sera activé jusqu'à la fin du cycle.



(a)

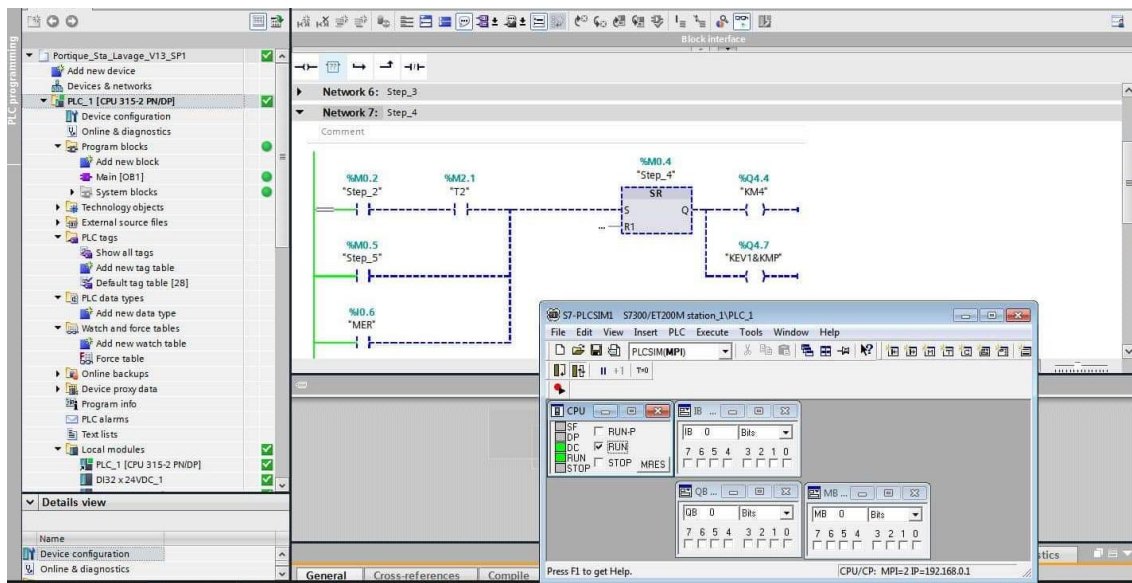


(b)

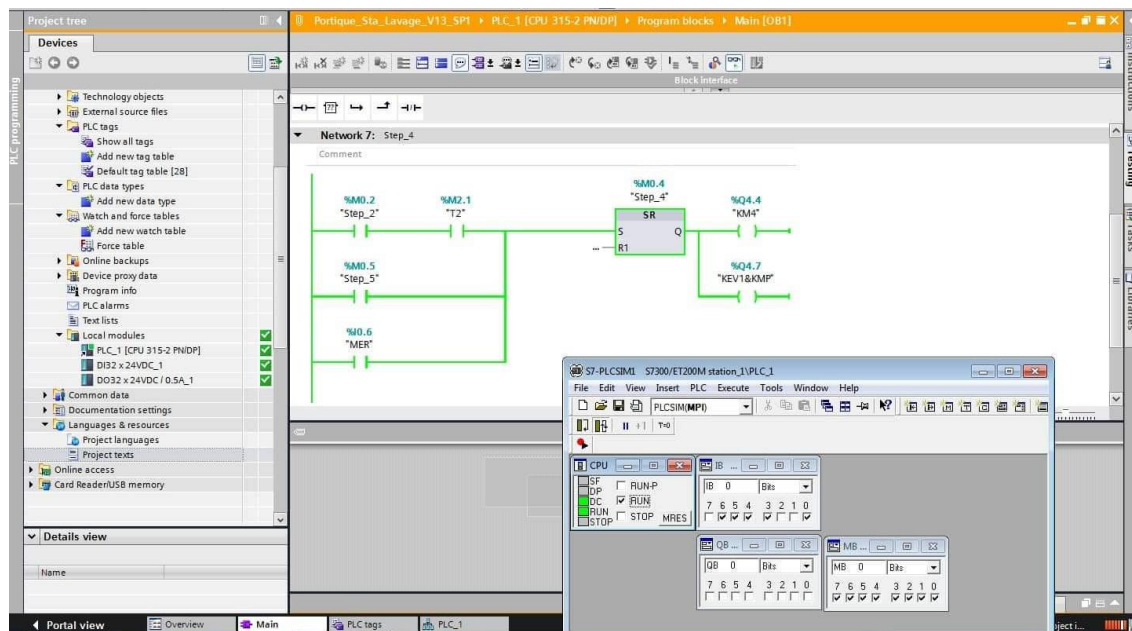
Figure IV.12 : Sortie des trois rouleaux

Etat Off de la sortie KM3 (a) , Etat On de la sortie KM3 (b)

Etape 5 : Le portique avance pour faire le pré-lavage en arrosant le véhicule avec de l'eau jusqu'à l'actionnement du capteur Cv.



(a)

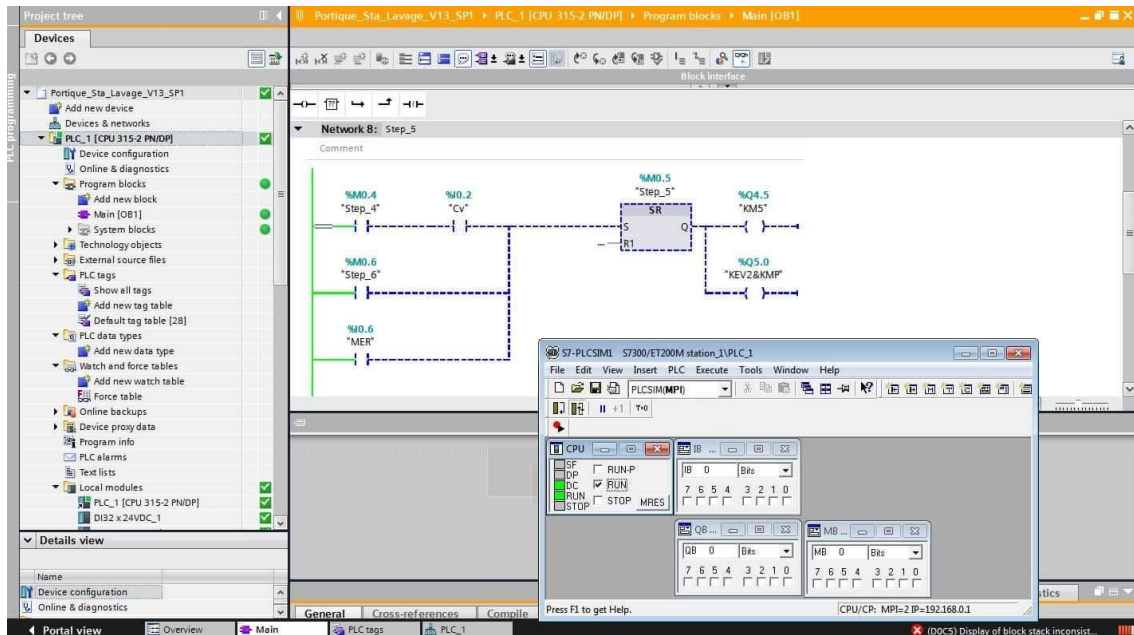


(b)

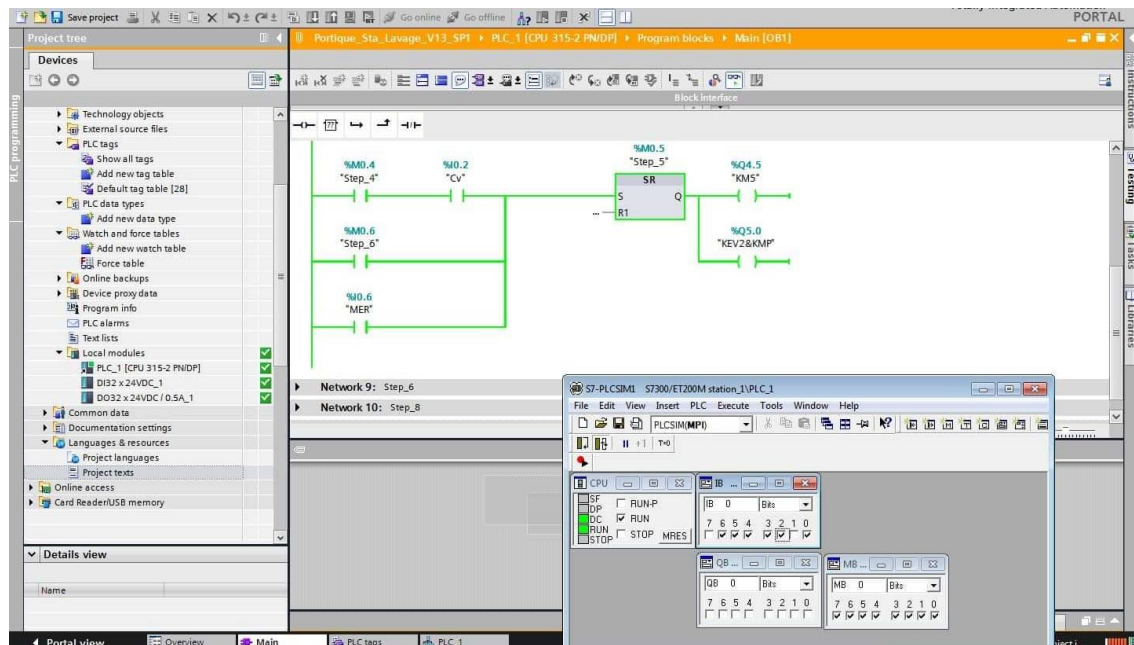
Figure IV.13 : Prélavage

Etat Off des sorties KM4 et KMP (a) ; état On des sorties KM4 et KMP (b)

Etape 6 : Le portique recule en arrosant le véhicule avec de l'eau savonnée jusqu'à l'actionnement du capteur Cr.



(a)

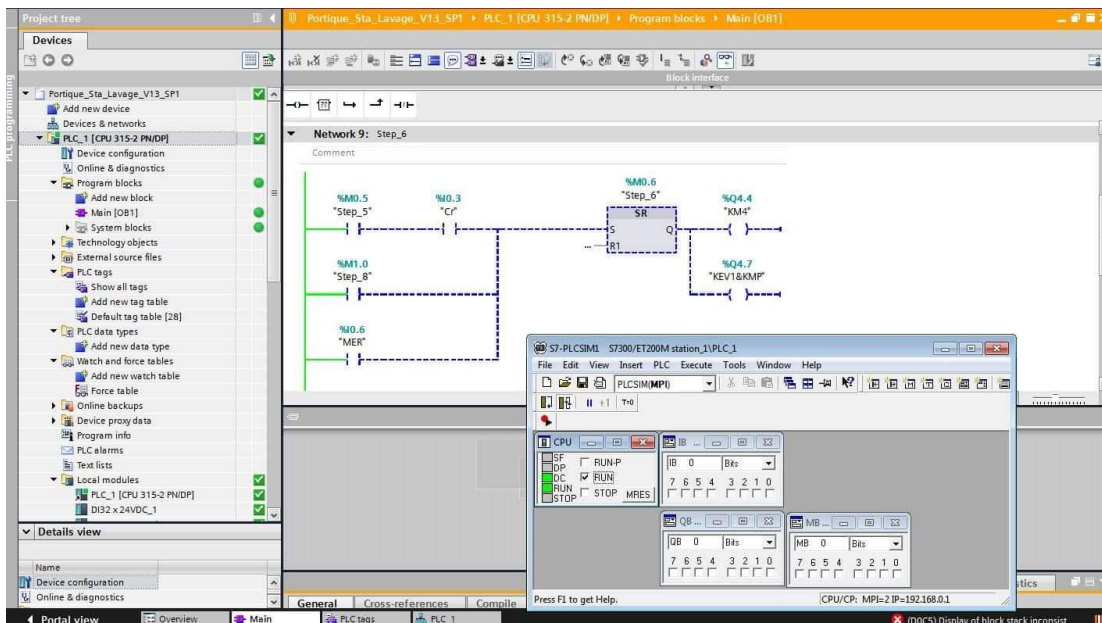


(b)

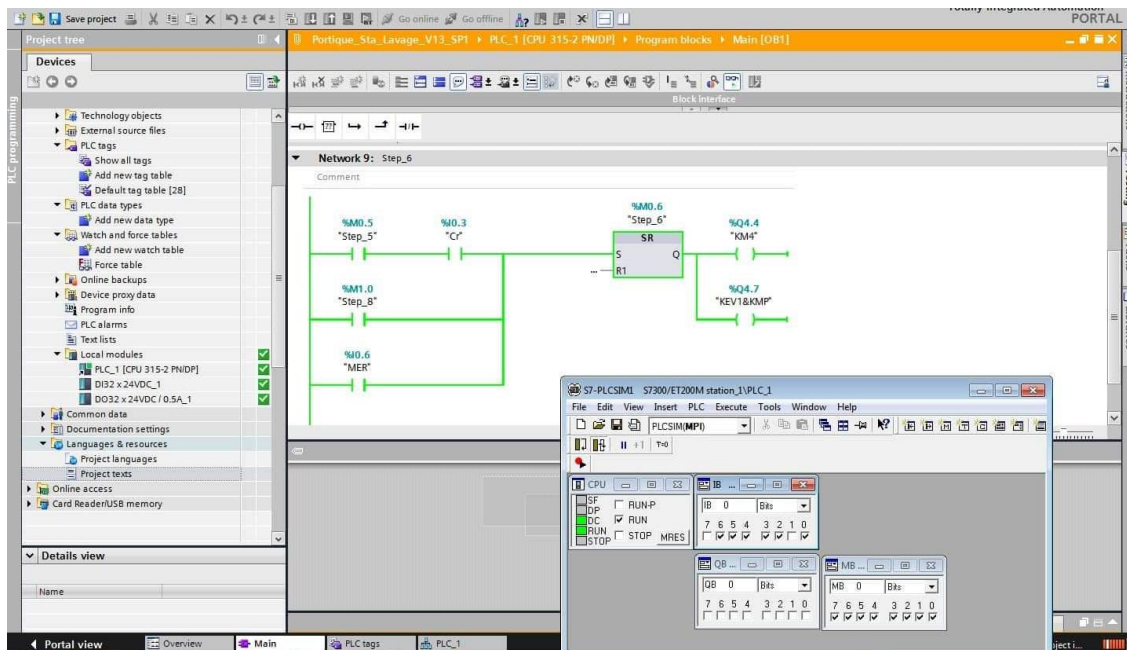
Figure IV.14 : Savonnage

Etat Off des sorties KM5 et KMP (a) ; Etat On des sorties KM5 et KMP (b)

Etape 7 : Le portique avance pour une deuxième fois pour effectuer l'opération de rinçage après l'actionnement du capteur Cv et les trois rouleaux sont arrêtés ;



(a)

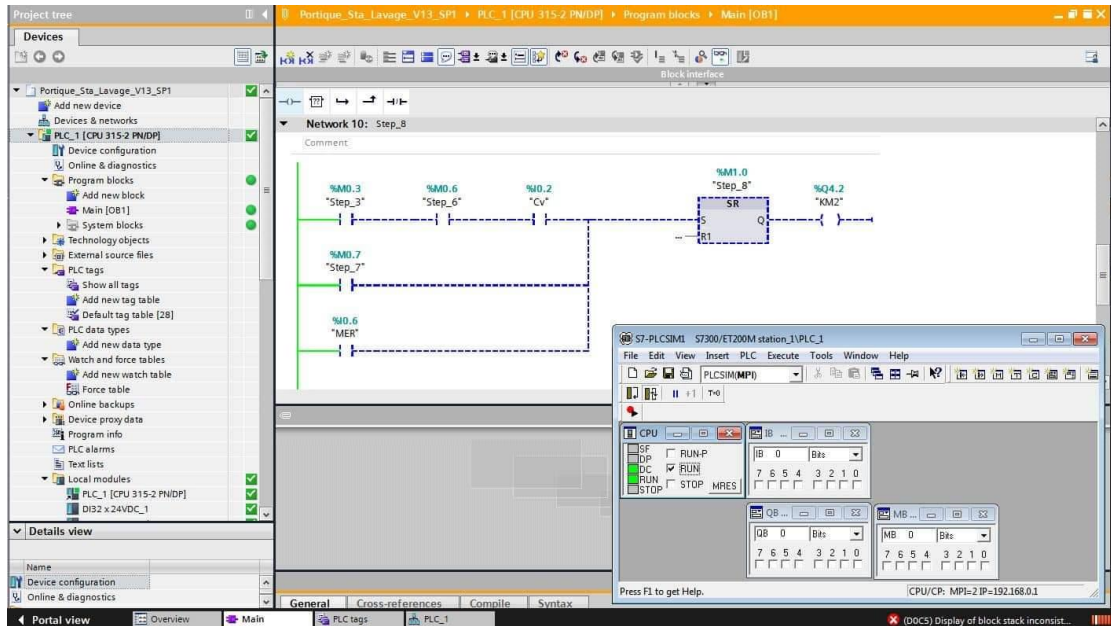


(b)

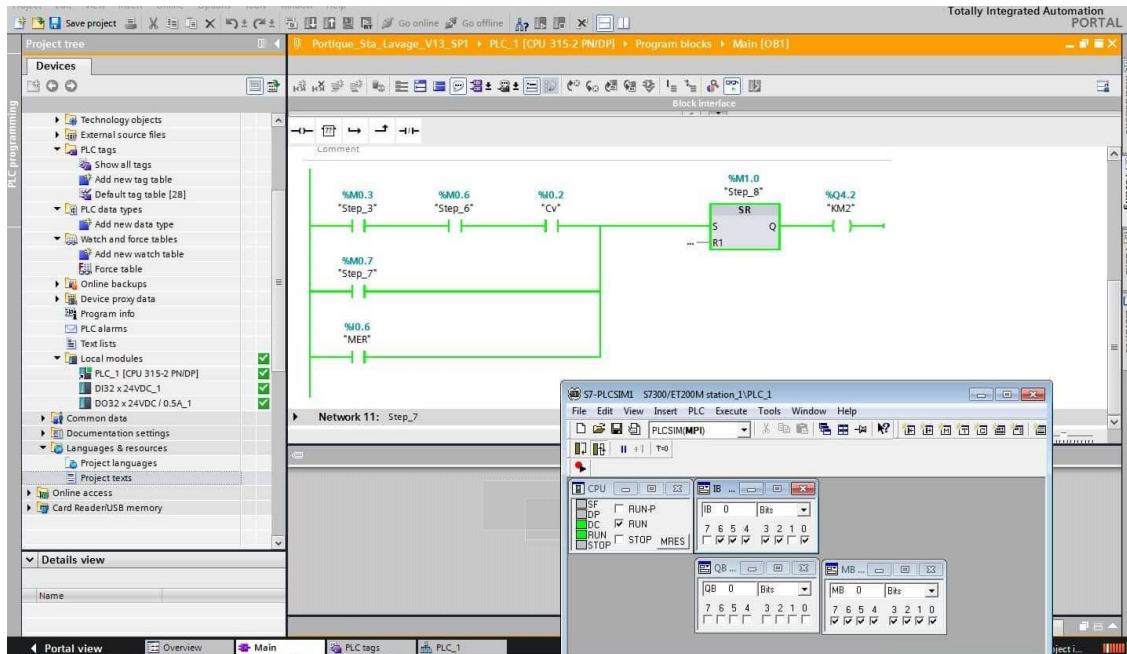
Figure IV.15 : Rinçage

Etat Off des sorties KM4 et KMP (a) ; état On des sorties KM4 et KMP (b)

Etape 8 : Le rouleau horizontal commencera à monter jusqu'à l'actionnement du capteur Ch



(a)

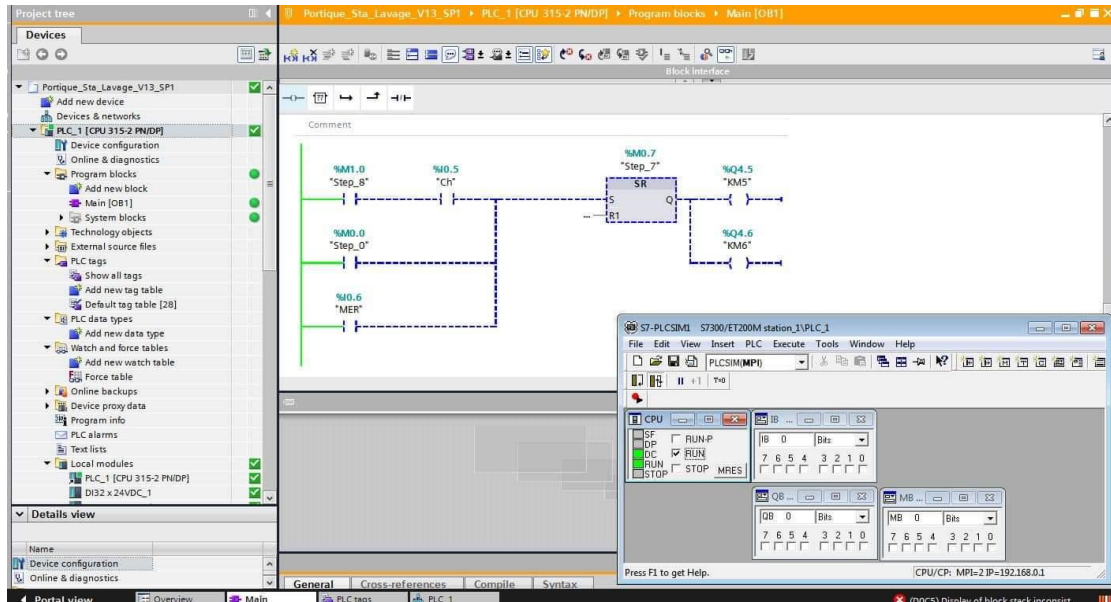


(b)

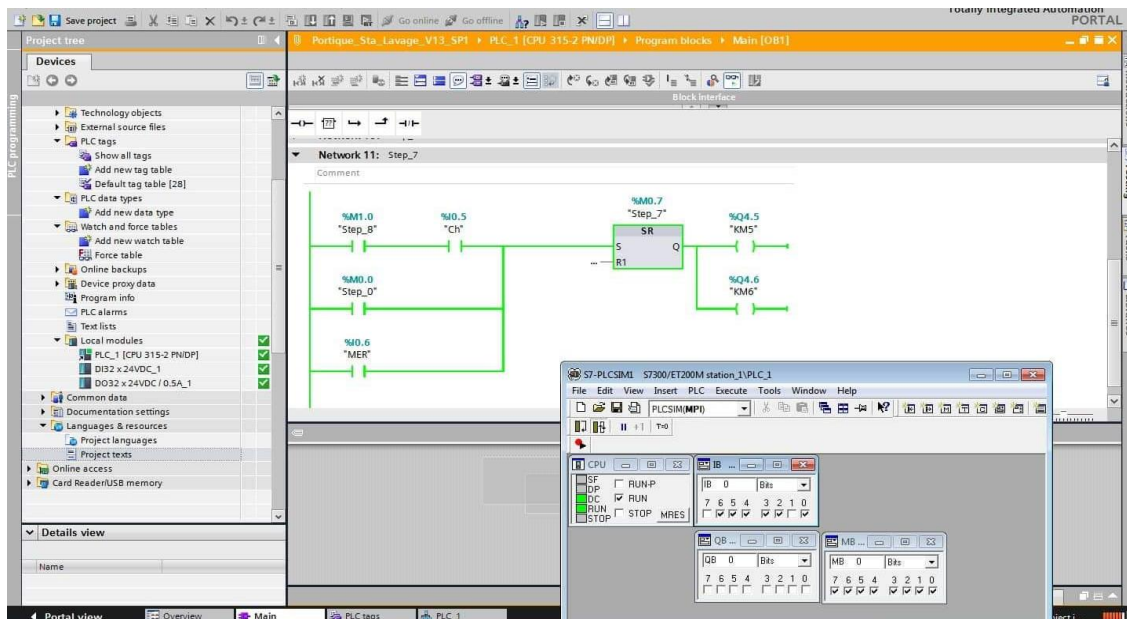
Figure IV.16 : Sortie Moteur M2

Etat Off de la sortie KM2 (a) ; Etat On de la sortie KM2 (b)

Etape 9 : Lorsque la 8^{ème} étape sera terminée, le portique va faire le retour pour une dernière fois. Au moment du recul du portique, le dispositif de séchage démarre jusqu'à l'actionnement du capteur Cr.



(a)



(b)

Figure IV.17 : Séchage et fin d'opération de lavage

Etat Off des sorties KM5 et KM6 (a) ; état On des sorties KM5 et KM6 (b)

Remarque : en annexe 3, on peut voir le GRAFCET de la station de lavage du point de vue système à titre indicatif.

IV.6 La conclusion :

Dans ce chapitre, nous avons commencé par présenter le schéma de principe, l'actigramme et le processus de fonctionnement de la station de lavage. Suite à cela, les deux parties essentielles qui composent un SAP qui sont la partie commande et la partie opérative ont été mise en évidence. Ensuite, nous avons montré les différentes étapes du programme élaboré qui permettent le fonctionnement du système.

En définitif, on peut dire que notre simulation c'est bien déroulé et que notre objectif a été atteint, ce qui permet pratiquement de minimiser l'effort physique, assurer la sécurité, gagner du temps et préserver notre environnement.

Conclusion Générale

Conclusion Générale

Ce travail de fin d'étude s'inscrit dans le cadre de l'automatisation d'une station de lavage automobile ; pour cela nous avons commencé par présenter des généralités sur les systèmes automatiques de production ; et monter les éléments essentiels qui nous permettent d'identifier la fonction globale pour chaque système de production ; au cours de ce travail, on a vu que les API sont devenues indispensables dans les chaînes de production industrielles à cause de leurs diversités d'utilisation comme la surveillance et le diagnostic direct qui nous aide à reconnaître les défauts et de la défaillance.

L'objectif de ce travail est de contribuer aux travaux d'automatisation des SAP afin d'expliquer le processus d'application des automates dans n'importe quel procédé industriel, de l'étude et la simulation (réalisation).

Nous avons passé en revue les automates programmables industriels, leurs caractéristiques, critères de choix, avantages, ainsi que les langages de programmation utilisables qui rendront un système automatisé plus simple et plus performant.

L'utilisation du logiciel STEP7 (TIA PORTAL) nous a permis d'implémenter quelques tâches d'automatisation sur l'automate S7-300 Siemens ; qui a la particularité d'intégrer des modules d'entrées/ sorties de type Tout Ou Rien (TOR), avec une programmation faite en langage Ladder.

Ce modeste travail, d'aspect pratique et riche d'informations, ne sera certainement utile dans le domaine professionnel puisque l'utilisation de ce logiciel (TIA PORTAL) de la Gamme Siemens et même pour les API de cette grande famille est actuellement le leader dans le domaine d'automatisation ; il nous permettra de réaliser différentes tâches sans prendre de risque dans un environnement difficile tout en préservant les performances idéales de chaque système automatisé.

En perspective et en relation à ce travail, la réalisation du système de lavage de voiture peut être concrétisée.

Bibliographie

- [1] Tahar Askri, « Introduction aux systèmes automatisés », 2005.
- [2] J. Perrin et F. Binet et C. Merlaud et J.P. Trichard « Automatique et informatique industrielle Bases théorique, méthodologique et technique »
- [3] BOUAMOUD Mohamed EL Amine et BRAHIMI Sofian Maamar ; « automatisation d'une station de lavage », mémoire de fin d'études de master, université de Saida, 2015.
- [4] Alain RIDEAAU et André BIANCIOTTO, « La technologie des systèmes automatisés », ISTIA, 17 octobre 2005.
- [5] <http://mickael78450.over-blog.com>
- [6] William Bolton, « Automates Programmables Industriels », DUNOD, Paris, 2015.
- [7] Michel G, « Architecture et application des automates programmables industriels », DUNOD, Paris, 1987
- [8] M. Bertrand, « Automates Programmable Industriel »
- [9] Schneider Electric « Automates Nano et plate-forme d'automatisme Micro »1999
- [10] Salim BEN SAOUD, LES AUTOMATES PROGRAMMABLES INDUSTRIELS (API)
- [11] P. Jargot. Langage de programmation pour api. Norme CIE 1131–3. Techniques de l'Ingénieur, S 8030, 23 pages, 1999
- [12] Duysinx P ; « Présentation des langages automates programmables ». Revue technique. SIEMENS PLC série S7, 2008.
- [13] LYCEE MONGE ; LA HAUVINIÈRE « Initiation-1-TIA-Portal-MS1.doc » 2009
- [14] JOHN alexandre Rabiot ; Article de journal LE FIGARO« Nettoyer sa voiture lavage manuel ou automatique » le figaro.fr Novembre 2011
- [15] <https://www.istobal.com> « L'impressionnante évolution du monde du lavage »

- [16] Document Siemens, « Automate programmable S7-300, caractéristiques électriques et Techniques des CPU SIMATIC » Edition Siemens, 2001
- [17] Document de formation pour une solution complète d'automatisation « Automatisierung- und Antriebstechnik, Siemens A&D Cooperates with Education, » Edition, 2005
- [18] J.-P. Thomesse, « Technique de l'Ingénieur, Ingénierie des systèmes homme Machine, » Edition, 2004
- [19] Document de formation pour une solution complète d'automatisation T I A. MODULE API
- [20] A. Simon, « Automates programmables industriels Niveau 1, » Edition l'Elan-Liège, 1991
- [21] Siemens AG, « SIMATIC STEP 7 dans le portail Totally Integrated Automation Portal », 2013.
- [22] Programmation des Automates S7-300, Introduction Au Logiciel TIA Portal

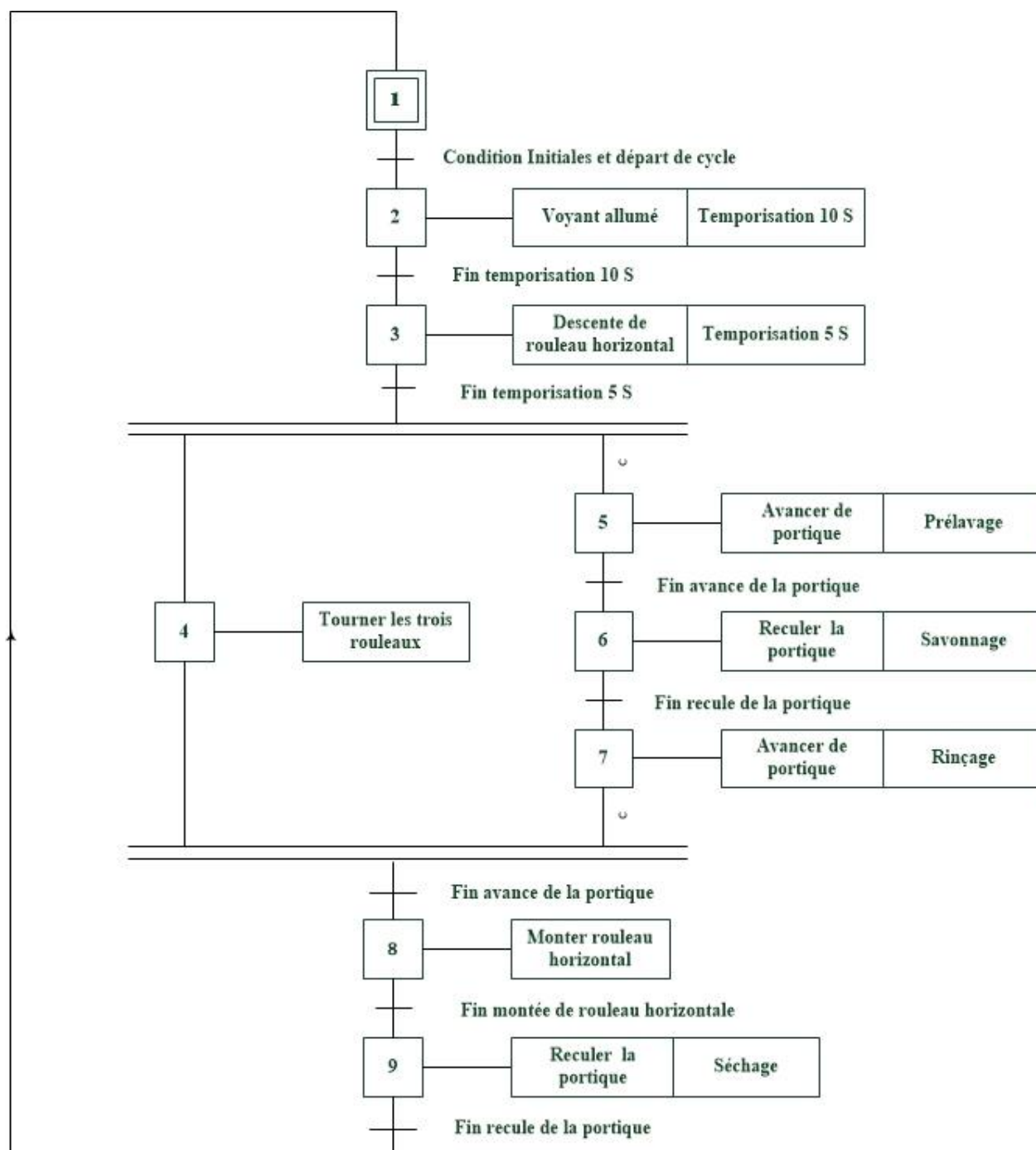
Webographie

- [W1] <https://www.indiamart.com/proddetail/roller-lever-limit-switch-15486362891.html>
- [W2] <https://www.ifm.com/de/en/product/VTV12A>
- [W3] <https://www.easypiscine.fr>
- [W4] <https://themarketresearchnews.com>
- [W5] <http://www.directindustry.fr/prod/raphael-valves-industries/product-119553-1290029.html>
- [W6] <https://sofamaroc.com/categorie-produit/automatisme-industriel-et-controle/>
- [W7] <https://fr.ubergizmo.com/2014/01/03/piller-distributeur-billets-malware-clef-usb.html>
- [W8] https://fr.123rf.com/photo_66555382_litre-bouteilles-en-plastique-sur-la-machine-d-emballage-%C3%A0-la-cha%C3%A9ne-de-coul%C3%A9e-equipement-d-emballag.html
- [W9] <http://triplewax-diamond.info/conseils-triplewax-nettoyage-exterieur>
- [W10] <https://www.moebelix.at/p/kaercher-hochdruckreiniger-k3-premium-full-control-0065920052>
- [W11] https://fr.123rf.com/photo_31179410_lavage-de-voiture-automatique-et-le-nettoyage-de-syst%C3%A8me-sans-contact-portail-de-la-rondelle-%C3%A0-la-statio.html
- [W12] <https://www.hellopro.fr/cablage-armoire-electrique-2013086-4305499-produit.html>
- [W13] <http://m.ahghxny.com/pr.jsp?moduleId=301&&>
- [W14] <https://hilimi.weebly.com/blog/archives/06-2018>
- [W15] <https://erdedelas.gq/siemens-cpu-315-2dp-manuellement>
- [W16] <https://m.made-in-china.com/product/Compatible-Siemens-6es7-322-1bf01-0AA0-PLC-8do-for-Industrial-Automation-Control-694499325.html>
- [W17] <http://www.tpautomation.de/Automatisierungssysteme/SIMATIC-S7-1500/Analogbaugruppen/6ES7531-7NF00-0AB0-S7-1500-SM-531-HF-8-Kanal-Analogeingabe-U/I::32855.html?language=de>
- [W18] <https://es.farnell.com/abb/1svr427035r1000/psu-ra-l-din-120w-12v-10a/dp/2103856>
- [W19] https://www.mabeo-direct.com/A-331401-siemens-passerelles-communications-profibus_asi

- [W20] <https://www.automation24.pl/zestaw-logo-8-12-24-rce-6ed1052-1md00-0ba8-logo-power-24v-1-3a-6ep1331-1sh03>
- [W21] <https://cellcode.us/quotes/controllers-plc-allen-bradley.html>
- [W22] <https://topmyanmarengineering.com/2017/06/22/simatic-s7-300-series/>
- [W23] https://cache.industry.siemens.com/dl/files/394/21742394/att_88161/v1/GS_FieldPGf.pdf

ANNEXE III

On trouve dans cet annexe le GRAFCET de la station de lavage point vu système :



Résumé

Résumé : Dans ce travail, nous avons étudié les systèmes automatiques de production ; la structure et les différents types des SAP existants. Chaque système automatisé a un cerveau qui se trouve dans la partie commande ; ce cerveau va gérer les différentes tâches afin de produire la matière d'œuvre plus le valeur ajouté ; dans notre cas nous avons utilisé l'automate programmable industriel S7-300 de la gamme SIEMENS pour piloter la partie opérative. L'utilisation du logiciel TIA PORTAL nous a permis de faire la programmation de la station de lavage objet de notre étude en langage Ladder puis on a simulé le fonctionnement à l'aide du PLCSIM. Les résultats obtenus ont été satisfaisant.

Mots clés : Station de lavage, logiciel TIA PORTAL, API S7-300, LADDER.

Abstract : In this work, we studied automatic production systems; the structure and the different types of existing SAPs. Each automated system is in the control part; this brain will manage different things in order to produce the exploitation material as well as the added value; We used the S7-300 industrial programmable logic controller from the SIEMENS range to control the operative part. The use of the TIA PORTAL software allows us to program the washing station of the object of our study in Ladder language on a simulated operation using PLCSIM. The results obtained were satisfactory.

keywords : Station de lavage, logiciel TIA PORTAL, API S7-300, LADDER.