

**REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE**  
**MINISTERE DE L'ENSEIGNEMENT SUPERIEUR ET DE LA RECHERCHE**  
**SCIENTIFIQUE**  
**UNIVERSITE MOHAMED BOUDIAF - M'SILA**

FACULTE DE TECHNOLOGIE  
DEPARTEMENT DE GENIEELECTRIQUE  
N° :.....



DOMAINE : SCIENCE ET TECHNOLOGIE  
FILIERE : ELECTROTECHNIQUE  
OPTION : ELECTROMECHANIQUE

**Mémoire présenté pour l'obtention**  
**Du diplôme de Master Académique**

**Présenté par :** Nadji Fouad  
Zemih Said

**Dirigé par :** Zouggar El Oualid

**Intitulé**

**Etude pour l'automatisation d'une presse  
rotative à carreaux granito à l'aide  
d'un automate S7-300**

**Soutenu devant le jury composé de :**

Nom et prénom Enseignant :

Mr	ABDOU Abdelhak	Université de M'sila	Président
Mr	ZOUGGAR EL Oualid	Université de M'sila	Rapporteur
Mr	MOHAMED ZINELAABIDINE GHELLAB	Université de M'sila	Examineur

**Année universitaire : 2020/2021**

بِسْمِ اللَّهِ الرَّحْمَنِ الرَّحِيمِ

## إهداء

إلى روح أبي الطاهرة رحمة الله عليه

إلى من زينت حياتي بضياء البدر، وشموع الفرح، إلى من منحتني

القوة والعزيمة، إلى الغالية على قلبي. **أمي**

إلى من أكرمني الله بها عندما كتبها من نصيبي **زوجتي العزيزة**

إلى العينين التاني استمد منهما السعادة، بناتي **نرجس و سحر**

إلى بهجة القلب **ابني الغالي زيد**

# ***Remerciements***

*Cette dernière période fut enrichissante et pleines d'activité grâce au bon "DIEU" tout puissant, qui nous a donné volonté, patience et santé. Nous avons eu la chance d'évoluer parmi des personnes qui nous ont toujours assuré de leur soutien, nous tenons à remercier très sincèrement*

*La première personne est notre promoteur et dirigeant de ce mémoire, **Mr. Zouggar El Oualid** pour nous avoir offert l'opportunité d'effectuer ce travail.*

*Nous tenons à remercier monsieur le directeur du département de génie électrique **Mr.Chouchou abdelmadjid** et tous les enseignants du département, particulièrement mesieur : **bouzidi riad**.*

*Enfin, nous tenons à exprimer notre gratitude envers tout le personnel de **l'USINE SORADAR***

*NADJI FOUAD et ZEMIH SAID*

# Sommaire

<b>INTRODUCTION GÉNÉRALE</b> .....	1
<b>CHAPITRE I : Automatisation et automate programmable</b>	
Introduction.....	3
I-1 L'automatisation .....	3
I-2 Les objectifs de l'automatisation.....	4
I-3 Conséquence de l'automatisation.....	4
I-3-1 Conséquence positive .....	4
I-3-2 conséquence négative .....	4
I-4 Structure d'un système automatisé.....	5
I-5 Différents type de commande.....	7
I-6 Technologies des automatismes .....	7
I-6-1 La logique câblée.....	8
I-6-2 Logique programmée .....	8
I-7 Les Automates Programmables Industriels.....	8
I-7-1 Historique.....	8
I-7-2 Définition générale.....	9
I-8 Architecture des automates.....	10
I-8-1 Aspect extérieur des API .....	10
I-8-2 Structure interne des API .....	11
I-9 Principe de fonctionnement d'un A.P.....	14
I-10 Programmation de l'API.....	15
I-11 Critères de choix d'un type d'automate et le langage.....	16
I-11-1 Gamme d'un automate .....	16
I-11-2 Critères technologiques.....	17
I-11-3 Les critères de choix d'un langage de programmation.....	17
Conclusion.....	17
<b>CHAPITRE II : Etude technique de la presse rotative</b>	
Introduction.....	18

II-1 Fabrication industrielle de carrelage granito.....	18
II-2 Présentation de la presse rotative .....	19
II-3 les processus de production.....	19
II-4 Identification des parties composantes de la presse.....	20
II-5 Le doseur .....	20
II-5-1 Centrale oléodynamique.....	20
II-5-2 Equipement pneumatique.....	21
II-5-3 Agitateur.....	21
II-5-4 Sonde électrique de contrôle de niveau.....	21
II-6 La moule vibrante .....	22
II-7 La table rotative.....	23
II-8_Le trémie .....	23
II-9 Centrage de la table et pressage oléodynamique .....	24
II-10 Chariot extracteur et démouleur .....	25
II-10-1 Mouvement de la pelle .....	25
II-11 Caractéristiques de la machine.....	26
II-12 Partie électrique .....	27
II-12-1 Alimentation .....	27
II-12-2 Armoire électrique.....	27
II-12-3 Panneau de commande.....	28
II-12-4 Fin de course.....	29
II-13 Partie pneumatique.....	30
II-14 Partie hydraulique.....	31
II-15 Problématique.....	32
Conclusion.....	33

### **CHAPITRE III : Automatisation de la presse avec l'API S7-300**

Introduction .....	34
III-1 Présentation de la gamme SIMATIC de SIEMENS.....	34
III-2 Les différentes variantes dans la gamme SIMATIC.....	35
III-2-1 SIMATIC S7.....	35
III-2-2 SIMATIC C7.....	35
III-2-3 SIMATIC M7.....	35

III-3 Choix d'automate.....	35
III-4 Présentation de l'automate S7-300.....	36
III-4-1 Module d'alimentation (PS) .....	36
III-4-2 Unité centrale (CPU).....	37
III-4-4 Module de signaux (SM).....	39
III-4-5 Capacité d'extension du S7-300.....	40
III-5 Matériel utilisé.....	41
III-6 Console de programmation.....	41
III-7 Logiciel de programmation « step7 ».....	42
III-8 La conception d'un programme avec Step7.....	42
III-8-1 Création du projet avec STEP7.....	43
III-8-2 Configuration matérielle.....	44
III-8-3 Adressage absolu des variables.....	46
III-8-4 Adressage mnémotechnique( symboliques ).....	48
III-8-5 Table des mnémotechniques globale.....	49
III-8-6 L'élaboration du programme.....	55
III-8-6-1 Système d'exploitation.....	55
III-8-6-2 Programme utilisateur.....	55
III-8-6-3 Organisation du programme utilisateur.....	56
III-8-6-4 Création de blocs de code et blocs de données.....	58
III-8-6-5 Traitement cyclique du programme.....	61
III-8-6-6 Langages de programmation.....	62
III-8-6-7 L'instruction.....	63
III-8-6-8 L'éditeur de programme (CONT/LIST/LOG).....	64
III-8-6-9 Quelques exemples du programme.....	65
III-8-7 Simulation de programme avec S7-PLCSINI.....	68
Conclusion.....	69

## **CHAPITRE IV : Supervision de la presse**

Introduction .....	70
IV-1 Interface Homme Machine.....	70
IV-2 La supervision .....	70
IV-3 Avantages de la supervision .....	71

IV-4 Structure d'un système de supervision.....	71
IV-4-1 Le module de visualisation (affichage) .....	71
IV-4-2 Le module d'archivage.....	71
IV-4-3 Le module de traitement.....	71
IV-4-4 Module de communication.....	71
IV-5 Choix de l'Interface Homme Machine.....	72
IV-6 Supervision sous Wincc flexible.....	72
IV-6-1 Description de Wincc.....	72
IV-6-2 Choix de pupitre de commande .....	73
IV-6-3 Intégration de WinCC flexible dans STEP7 .....	74
IV-6-3-1 Avantages de l'intégration au STEP7 .....	74
IV-6-4 Application la supervision sous WinCC Flexible 2008.....	74
IV-6-4-1 Insertion d'une station HMI dans SIMATIC manager.....	74
IV-6-4-2 Choix de la station (pupitre) .....	75
IV-6-4-3 Configuration du réseau.....	76
IV-6-4-4 Création des vues.....	77
Conclusion .....	79
<b>CONCLUSION GÉNÉRALE .....</b>	<b>80</b>

## **BIBLIOGRAPHIE**

# Table des figures

FIG ( I-1) : Structure d'un système automatisé .....	6
FIG ( I-2) : Les différents types de solutions d'automatisme.....	7
FIG ( I-3) : Présentation un API.....	9
FIG ( I- 4) : Les automates compactes.....	10
FIG ( I-5) : Les automate programmables modulaires.....	11
FIG ( I-6) : structure interne d'un API.....	11
FIG ( I-7) : Le cycle d'un API.....	15
FIG ( II -1) : présentation carrelage bicouche.....	18
FIG ( II -2 ) : La presse rotative a carreaux.....	19
FIG ( II -3) : Les parties composantes de la presse.....	20
FIG ( II - 4) : Une photo prise de doseur.....	21
FIG ( II -5) : Une photo prise de moule vibrante.....	22
FIG ( II -6) :système croix de malte et freinage table.....	23
FIG ( II -7) : Une photo prise de trémie.....	24
FIG ( II -8) : Système pressage des carreaux et centrage la table.....	25
FIG ( II -9) : chariot et démouleur.....	26
FIG ( II -10) : Armoire électrique de la presse.....	27
FIG (II -11) : La cabine pneumatique.....	30
FIG (II -12) : centrale hydraulique de pressage.....	31
FIG ( III-1) : Présentation de la gamme SIMATIC .....	34
FIG ( III-2) : Lautomate programable s7-300.....	36
FIG ( III-3) : L'unité centrale CPU313.....	37
FIG ( III-4) : Module de signaux (SM).....	39
FIG ( III-5) : Capacité d'extension du S7-300.....	40
FIG ( III-6) : Console de programmation.....	41
FIG ( III-7) : Organigramme pour la création de projets sous Step7.....	42
FIG ( III-8) : Création du projet avec STEP7.....	43
FIG ( III-9) : Les étapes de la configuration matériel.....	44
FIG ( III-10 ) : La fenêtre HW Config.....	45
FIG ( III-11 ) : Adressage des modules S7-300 .....	46
FIG ( III-12 ) : Adressage des éléments de mémoire utilisateur.....	47

FIG ( III-13 ) : La table des mnémoniques.....	54
FIG (III-14) : blocs de programme dans step7.....	57
FIG (III-15) :Les parties de bloc de code.....	58
FIG (III-16) : création un bloc de code.....	59
FIG (III-17) :Création un bloc de données.....	60
FIG (III-18) : structure arborescente de programme step7.....	60
FIG (III-19) : Traitement cyclique du programme. ....	61
FIG (III-20) : Représentation des langages de programmation step7.....	62
FIG (III-21) : Les types des opérations.....	63
FIG (III-22) : Structure de l'éditeur de programme CONT/LIST/LOG.....	63
FIG (III-23) : Structure de la section instructions d'un bloc CONT.....	64
FIG (IV.1) : Structure d'un système de supervision.....	72
FIG (IV-2): Le pupitre opérateur MP270 8 touch.....	73
FIG (IV-3): Insertion d'un nouveau Station.....	75
FIG (IV-4) : Choix de pupitre.....	75
FIG (IV -5) : Espace de travail.....	76
FIG (IV -6) : la configuration des réseaux.....	77
FIG (IV-7): vue d'accueil.....	78
FIG (IV-8): vue de temporisation.....	78
FIG (IV -9): vue d'état des entrées /sorties.....	79

# Introduction générale

## INTRODUCTION GENERALE

L'être humain, par sa nature, a toujours recherché le moyen d'économiser ses efforts. Il n'a jamais cessé de mettre son intelligence et son imagination au service de ce but et ceci afin de créer un partenaire qui fera le travail à sa place. L'arrivée récente des systèmes automatisés a permis d'éliminer un bon nombre des travaux pénibles et de réaliser des tâches répétitives et fastidieuses.

L'évolution de la technologie ainsi que le développement de l'électronique et de l'informatique ont donné naissance à l'automatique. Celle-ci a évolué vers l'étude et la maîtrise des systèmes de plus en plus complexes permettant l'exécution et le contrôle des tâches techniques d'une manière plus rapide et plus efficace et cela en s'appuyant sur des machines et des robots qui fonctionnent avec une intervention humaine très réduite, les automates programmables industriels (API) est l'une des plus importantes de ces techniques de contrôle.

Un automate programmable est un appareil dédié au contrôle d'une machine ou d'un processus industriel, constitué de composants électroniques, comportant une mémoire programmable par un utilisateur non informaticien, à l'aide d'un langage adapté. En d'autres termes, un automate programmable est un ordinateur logique, ou ordinateur, au jeu d'instructions volontairement réduit, destiné à la conduite et la surveillance en temps réel de processus industriels [4].

Les entreprises industrielles sont soumises à l'obligation de suivre le développement technologique et la rénovation permanente des unités de productions, comme le passage de la logique câblée (généré plusieurs difficultés à cause de leur encombrement et leur manque de souplesse) à la logique programmée qui présente de meilleurs avantages par rapport à la logique câblée.

Le travail qui nous a été demandé c'est l'étude d'automatisation d'une presse rotative à carreaux granito, pour remplacer le système de commande (logique câblée) par une technologie programmée via un automate programmable industriel S7 300 de SIEMENS.

Nous avons donc organisé notre travail de la manière suivante :

- ❖ **Chapitre I** qui vise à la description théorique sur les systèmes automatisés. Nous évoquerons ensuite, les automates programmables avec un aperçu sur leurs architectures principe de fonctionnement et langages de programmations.

- ❖ **Chapitre II** sera consacré à la description et l'étude technique de la presse rotative a carreaux granito.
- ❖ **Chapitre III** sera dédié à l'automatisation de la presse à l'aide d'un API marque SIEMENS de type S7-300 par le logiciel de programmation STEP7.
- ❖ **Chapitre IV** sera réservé à la création d'interface homme-machine (HMI) pour la presse rotative à l'aide du logiciel WinCC flexible de SIEMENS .
- ❖ Nous terminons notre travail par une conclusion g é n é r a l e et des perspectives.

# Chapitre I

# Automatisation et automate programmable

## **Introduction**

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 et une meilleure régularité des résultats.

Nous évoquerons dans ce chapitre, en premier lieu des notions générales sur les systèmes automatisés, leur structure et leurs technologies, puis nous passerons à la description détaillée de l'architecture des automates programmables industriels (API), les différents langages de programmation et les critères de choix essentiels d'un automate.

### **I-1 L'automatisation:**

Tout système de production nécessite la présence d'opérateurs humains d'exploitation, de réglage et de maintenance. Automatiser un système de production consiste donc à réduire l'intervention humaine tout au long du processus de fabrication et à optimiser l'utilisation des matières et des énergies par l'emploi de nouvelles technologies.

En d'autre terme, l'automatisation de la production consiste à transférer tout une ou partie des tâches de coordination, auparavant exécutées par des opérateurs humains, dans un ensemble d'objets techniques appelé Partie Commande (PC).[1]

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.

Cette automatisation permet donc globalement de rendre plus compétitif le produit, l'entreprise et les hommes. Nous parlons alors de Systèmes Automatisés de Production (SAP).

## I-2 Les objectifs de l'automatisation:

L'automatisation permet d'apporter des éléments supplémentaires à la valeur ajoutée par le système. Ces éléments sont exprimables en termes objectifs par :

- Accroître la productivité du système c'est-à-dire augmenter la quantité de produits élaborés pendant une durée donnée, cet accroissement de productivité exprime un gain de valeur ajoutée sous forme :

- > d'une meilleure rentabilité.
- > d'une meilleure compétitivité.....etc.

- Améliorer la flexibilité de production .
- Améliorer la qualité du produit grâce à une meilleure répétabilité de la valeur ajoutée .
- S'adapter à des contextes particuliers .
- Adaptation à des environnements hostiles pour l'homme.
- Adaptation à des tâches physiques ou intellectuelles pénibles pour l'homme.
- Augmenter la sécurité. ....etc.

D'autres objectifs. à caractères sociaux. financiers... peuvent s'ajouter à ceux-ci.[12]

## I-3 Conséquence de l'automatisation:

### **I-3-1 Conséquence positive :**

- Améliorer les conditions de travail (effectuer des tâches pénibles, dangereuses et répétitives).
- Améliorer la sécurité.
- Améliorer la précision.
- Réduire les coûts de fabrications (produit plus compétitif).
- Augmenter la productivité (réduire le temps de travail nécessaire à la production. donc augmenter les cadences de travail).
- Flexibilité (une machine peut s'adapter à plusieurs productions).
- Confidentialité (une machine ne peut pas parler).
- Un Système automatisé. peut travailler 24h sur 24h.

### **I-3-2 Conséquence négative :**

- Incidence sur l'emploi (licenciement chômage : la mise en place d'une machine se substituant à 10 salariés n'aboutit pas à la création de 10 emplois).

- Investissement pour l'achat de machines.
- Coût de maintenance.
- Consommation d'énergie.
- Formation d'un personnel plus qualifié (technicien de maintenance. de contrôle...).

#### **I-4 Structure d'un système automatisé:**

Un Système Automatisé de Production est globalement composé d'une Partie Opérative (PO) et d'une Partie Commande (PC) Figure (I-1).

**La Partie Opérative** Elle est formée de l'ensemble des divers organes physiques qui interagissent sur le produit pour lui conférer une valeur ajoutée : les pré-actionneurs, les actionneurs et les capteurs.

- Les pré-actionneurs servent de relais de puissance entre la commande et les actionneurs.
- Les actionneurs transforment l'énergie reçue en une énergie exploitable par les effecteurs qui agissent et transforment le produit.
- Les capteurs recueillent les informations : état ou position du produit, alarmes, etc. traduisant un changement d'état du procédé. Ces changements induisent le calcul de la commande du procédé de production par son Système de Contrôle/Commande ou Partie Commande.

**La Partie Commande** La Partie Commande (PC) ou Système de Contrôle/Commande est l'ensemble des moyens et d'informations destiné :

- à coordonner la succession des actions sur la Partie Opérative .
- à surveiller son bon fonctionnement .
- à gérer les dialogues avec les intervenants.
- à gérer les communications avec d'autres systèmes.
- à assurer le traitement des données et des résultats relatifs au procédé, aux matières d'œuvre, aux temps de production, à la consommation énergétique... (gestion technique).

Un procédé industriel passe d'un état initial à un état final via une succession d'états intermédiaires qui marquent son évolution. Cette évolution est surveillée grâce aux capteurs qui renvoient des comptes rendus de l'état du système. La Partie Commande traite ces informations et élabore les ordres renvoyés vers les actionneurs ou préactionneur.[2]

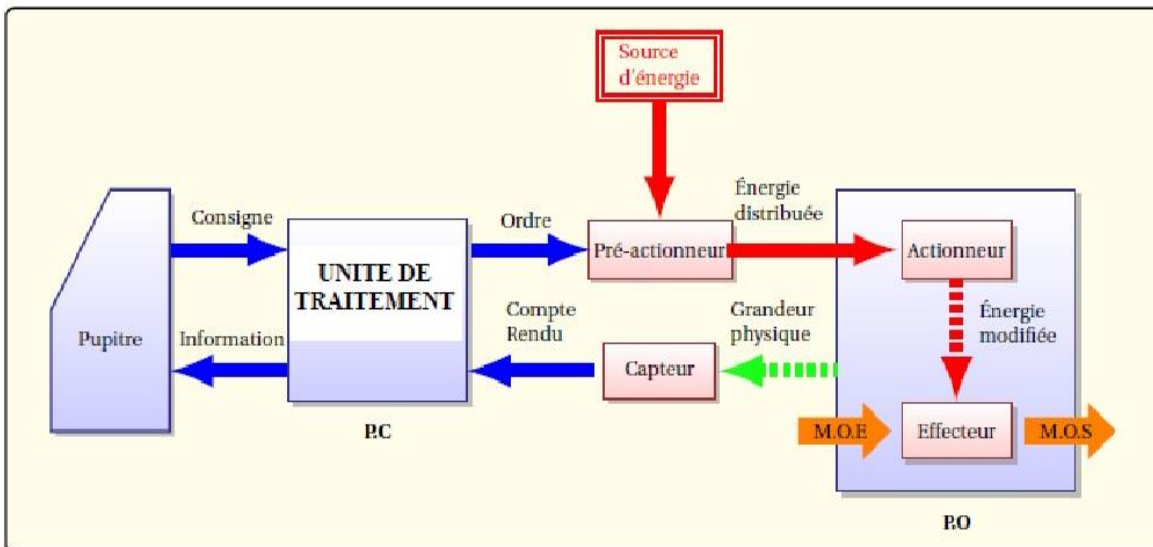


FIG ( I-1) : Structure d'un système automatisé.

Le tableau ( I-1) suivant montre les différents éléments d'un système automatisé. [3]

Diagramme fonctionnel	Exemples	Commentaires
<p><b>EFFECTEUR</b></p> <p>↓ W2</p> <p>Matière d'oeuvre → <b>AGIR Sur la M.O.</b> → Matière d'oeuvre</p>	<p>Fraise, Foret, Mors d'étau, pince de robot...</p>	<p>Les effecteurs sont multiples et variés et sont souvent conçus spécialement pour s'adapter à l'opération qu'ils ont à réaliser sur la Matière d'oeuvre.</p> <p>Ils reçoivent leur énergie des actionneurs.</p>
<p><b>ACTIONNEUR</b></p> <p>Énergie d'entrée (W1) → <b>CONVERTIR L'énergie</b> → Énergie utile (W2)</p>		<p>Convertissent l'énergie qu'ils reçoivent des pré-actionneurs en une autre énergie utilisée par les effecteurs. Ils peuvent être Pneumatiques, Hydrauliques ou Electriques</p>
<p><b>PRE-ACTIONNEUR</b></p> <p>↓ Pilotage</p> <p>Énergie du réseau (W1) → <b>DISTRIBUER L'énergie</b> → Énergie distribuée à l'actionneur</p>	<p>Variateur    Contacteur    Distributeurs</p>	<p>Distribuent l'énergie aux actionneurs à partir des ordres émis par la PC.</p>
<p><b>CAPTEUR</b></p> <p>Information source → <b>DETECTER MESURER Une grandeur</b> → Information image</p>		<p>Renseignent la PC sur l'état de la PO, Ils peuvent détecter des positions, des pressions, des températures, des débits...</p> <p>Peuvent être électriques ou pneumatiques.</p> <p>Signaux du type TOR, Analogique ou Numérique.</p>
<p><b>TRAITEMENT</b></p> <p>Signal d'entrée (capteurs, consignes...) → <b>TRAITER L'information</b> → Signal de sortie (Pilotage des préactionneurs, signalisation...)</p>	<p>Automate    Séquenceur pneumatique    Cellules logiques</p>	<p>Dans les systèmes modernes, l'API assure de plus en plus cette fonction.</p> <p>Certains systèmes purement pneumatiques peuvent être contrôlés par des séquenceurs ou des fonctions logiques.</p>
<p><b>DIALOGUE</b></p> <p>Consignes de l'opérateur, Infos de la PC → <b>FAIRE COMMUNIQUER Homme/machine</b> → Consignes Vers la PC, Infos vers opérateur</p>		<p>L'unité de dialogue permet à l'opérateur d'envoyer des consignes à l'unité de traitement et de recevoir de celle-ci des informations sur le déroulement du processus.</p>

Tableau (I-1) : Différents éléments d'un système automatisé.

### I-5 Différents type de commande:

Les systèmes automatisés sont constitués d'un ensemble d'opérations élémentaires effectuées sans intervention humaine excepté l'ordre de marche, ces systèmes se trouvent en deux types : combinatoires et séquentiels.

Les systèmes automatisés **combinatoires** sont des systèmes dont les sorties dépendent des entrées seulement. Chaque combinaison des variables d'entrée fait correspondre un seul état de sortie. l'outil mathématique utilisé pour décrire les systèmes combinatoires est l'algèbre de Boole qui est basée sur la construction de la table de vérité et l'établissement des équations logiques des actionneurs par l'utilisation des règles de simplification comme le tableau de Karnaugh.

Par opposition, les sorties à l'instant  $t$  des systèmes **séquentiels** sont déterminées par les entrées à cet instant et les états antérieurs de ce système. ce qui implique qu'une même combinaison d'entrée ne générera pas toujours les mêmes sorties. Ces systèmes sont les plus répandus dans le domaine industriel, plusieurs approches sont utilisées pour faire la synthèse des systèmes en vue de concevoir la partie commande. Parmi elles, on peut citer : la méthode de Huffman, le GRAFCET et les réseaux de Pétri.[3]

### I-6 Technologies des automatismes :

Nous disposons de nombreux outils technologiques pour réaliser l'organe de commande de notre système que l'on regroupe habituellement en deux catégories fondamentales : les solutions câblées et les solutions programmées (FIG ( I-2 )).

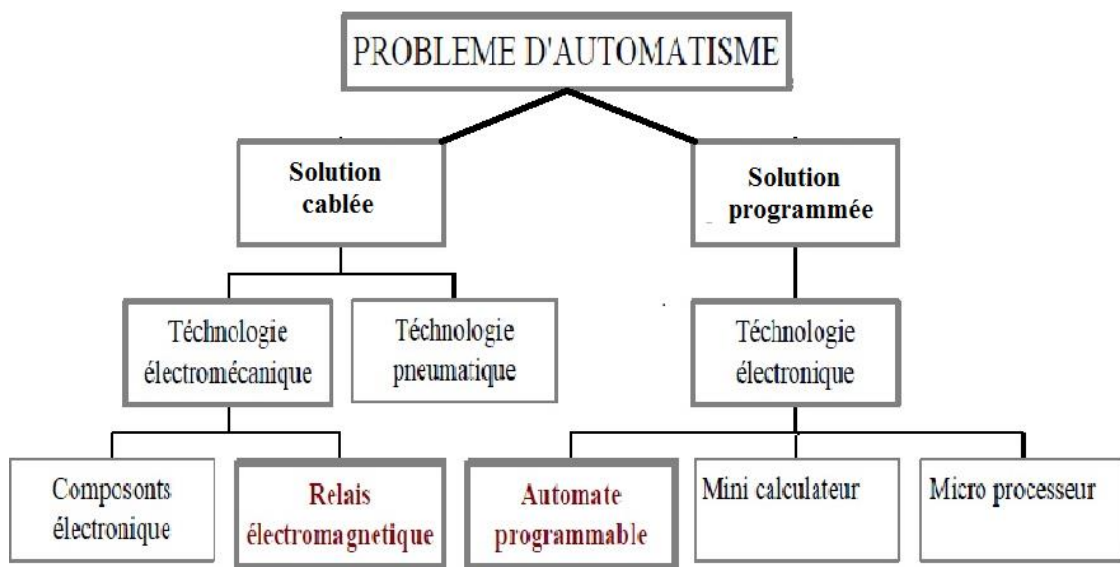


FIG ( I-2 ) : Les différents types des solutions d'automatisme .[1]

**I-6-1 La logique câblée:**

L'automatisme est obtenu en reliant entre eux les différents constituants de base ou fonctions logiques par câblage. La logique câblée correspond à un traitement parallèle de l'information. Plusieurs constituants peuvent être sollicités simultanément.

Elle est étudiée et réalisée une fois pour toutes sur un schéma donné : Les fonctions sont réalisées par voie matérielle. Elle exige un grand nombre de composants et rend les montages encombrants et chers.

Enfin, elle n'offre guère de souplesse : la durée des études pour réaliser un montage donné (et donc pour le modifier le cas échéant) est longue.

**I-6-2 Logique programmée:**

La logique câblée est choisie en cas de réalisation unitaire, comportant de nombreuses entrées sorties, nécessitant des modifications de temps en temps (par exemple partie de ligne de production automatique). La programmation est réalisée directement en différents langages à l'aide d'une console de programmation (actuellement on utilise un PC, ce qui permet de saisir et simuler l'automatisme au calme, avant le test sur site).

L'élément principal de cette technique s'appelle l'Automate Programmable Industriel ou l'API.

**Avantages :**

- Souplesse et adaptabilité de l'installation (Remplacement des fonctions combinatoires et séquentielles par un programme).
- Facilité de modification de la loi de contrôle: Il suffit de modifier le programme.
- Solution plus compacte : Faible liaison entre le volume matériel et la complexité du problème (effet simplement sur les entrées : sorties et taille mémoire).

**I-7 Les Automates Programmables Industriels:****I-7-1 Historique:**

Le premier est apparu en 1969 aux Etats Unis sur une demande de Ford pour remplacer les grosses armoires à relais des chaînes de la construction automobile. La réponse vint d' ALLEN BRADLEY. La NASA fait en suite un appel d'offres sur le même thème auquel MODICON répondit. Sa date de création coïncide avec le début de l'ère du microprocesseur et avec la généralisation de la logique câblée modulaire.

Le premier automate français fût le PB6 de MERLIN GERIN en 1971.[2]

**I-7-2 Définition générale:**

Un automate programmable industriel (API) est une machine électronique spécialisée dans la conduite et la surveillance en temps réel de processus industriels et tertiaires.

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, sa programmation à partir des langages spécialement développés pour le traitement des fonctions d'automatismes facilitent son exploitation et sa mise en œuvre.

Selon la norme française EN 61131-1. un automate programmable 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 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'automate programmable 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.[3]

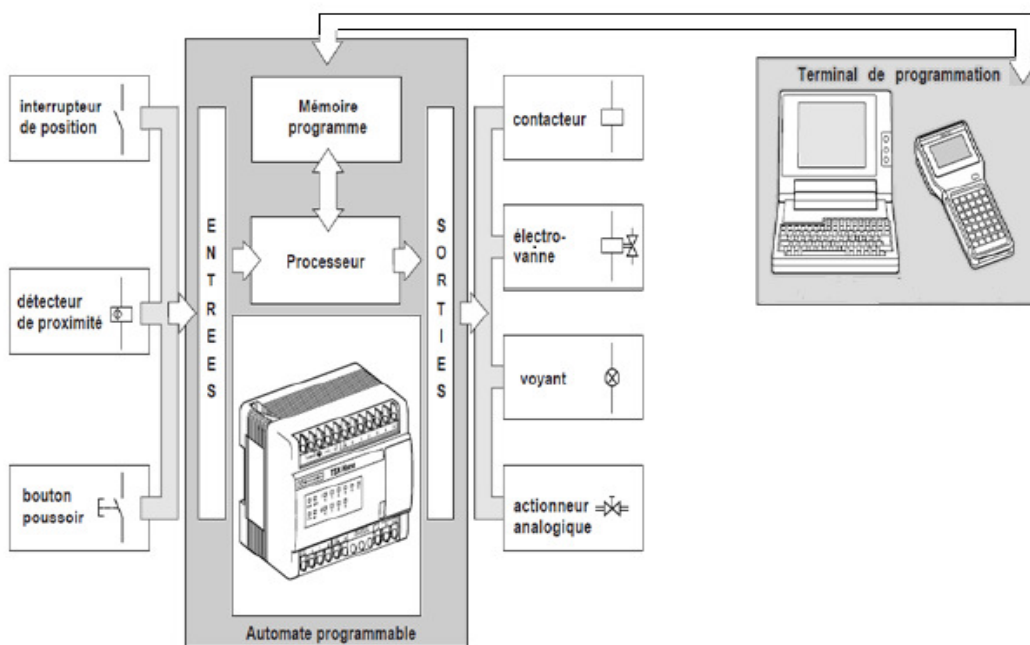


FIG ( I-3 ) : Présentation un API .[7]

**I-8 Architecture des automates:**

**I-8-1 Aspect extérieur des API :**

Les automates programmables industriels peuvent être de type compact ou modulaire.

**A- Type compact (centralise) :**

Il intègre le processeur, l'alimentation, les entrées et les sorties dans un seul boîtier (rack) Selon les modèles et les fabricants, il pourra réaliser certaines fonctions supplémentaires (comptage ,E/S analogiques ...) et recevoir des extensions en nombre limité, Ces automates de fonctionnaient simple. sont généralement destinés à la commande de petits automatismes.

SIEMENS LOGO	CROUZET MILLENIUM	SCHNEIDER ZELIO	SCHNEIDER TWIDO	MOELLER PS4
				

FIG ( I- 4 ) : Les automates compactes.

**B- Type modulaire :**

L'automate programmable ce présente comme un ensemble de blocs fonctionnels, généralement chaque bloc est physiquement réalisé par un module spécifique (coffret. Rack. baie ou cartes), ces différents modules s'articulant autour d'un canal de communication ( le bus interne).

L'automate programmable est du type modulaire contenant un boîtier ( rack), un module d'alimentation, un processeur, des modules d'E S, des modules de communication et de comptage, Cette organisation modulaire permet une grande souplesse de configuration pour les besoins de l'utilisateur, Ainsi qu'un diagnostic et une maintenance facilités et elle destinée pour les automatismes complexes où puissance, capacité de traitement et flexibilité sont nécessaires.[11]



FIG ( I- 5 ) : Les automate programmables modulaires

**I-8-2 Structure interne des API :**

La structure interne d'un automate programmable industriel (API) est assez voisine de celle d'un système informatique simple. Cette structure comporte quatre parties principales: Une unité centrale de traitement (UCT) , Une mémoire , des Interfaces et des modules d'entrées-sorties , Une alimentation et Un bus interne (liaisons parallèles) est utilisé pour échanger les informations entre les différents éléments de l'automate (entrées. sorties. mémoires).

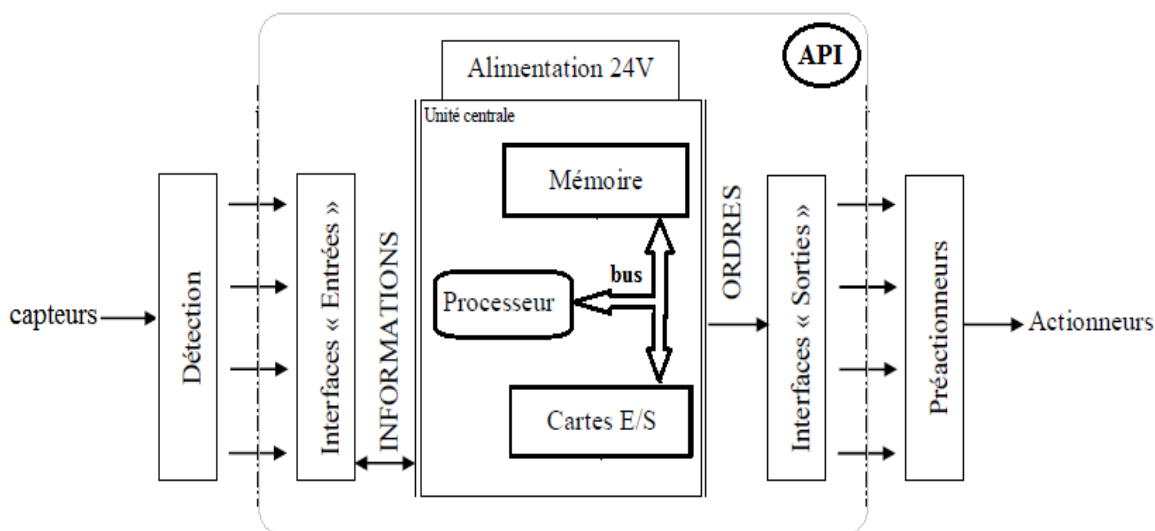


FIG ( I- 6 ) : structure interne d'un API .[10]

**A- Processeur :**

Son rôle consiste d'une part à organiser les différentes relations entre la zone mémoire et les interfaces d'entrées et de sorties et d'autre part à exécuter les instructions du programme, Les instructions sont effectuées les unes après les autres. séquencées par une horloge, il réalise également d'autres fonctions :

- Gestion des entrées sorties :
- Surveillance et diagnostic de l'automate par une série de tests lancés à la mise sous tension ou cycliquement en cours de fonctionnement .
- Dialogue avec le terminal de programmation aussi bien pour l'écriture et la mise au point du programme qu'en cours d'exploitation pour des réglages ou des vérifications de données

**B- Mémoire :**

Elle est conçue pour :

- recevoir les informations issues des capteurs d'entrées.
- recevoir les informations générées par le processeur et destinées à la commande des sorties.
- recevoir et conserver le programme du processus.

Il existe dans les automates trois types de mémoires qui remplissent des fonctions différentes ( classement par leur utilisation) :

- **Mémoire de programme** : Cette mémoire est utilisée pour stocker le programme (la liste ordonnée des instructions à traiter par le processeur). Elle est en général de type EPROM (effacement par rayons ultraviolets),EEPROM (effaçable électriquement) ou FLASH EPROM (effacement par bloc) .
- **Mémoire système** : Cette mémoire présente dans le cas d'automates à microprocesseurs. est utilisée pour stocker le système d'exploitation et elle est programmée en usine par le constructeur, elle peut donc sans problème être réalisée en technologie PROM (c'est-à-dire programmable une seule fois. sans possibilité d'effacement) voire ROM (mémoire morte accessible uniquement en lecture).
- **Mémoire de données**: Elle est utilisable en lecture-écriture des données pendant le fonctionnement. C'est une mémoire de type RAM (mémoire vive dans laquelle on peut lire. écrire et effacer) utilisant une technologie spéciale (CMOS) à très faible consommation électrique du moins. à l'état de repos et elle nécessite une batterie de sauvegarde.

**C- Interfaces et cartes d'Entrées / Sorties :**

Les entrées reçoivent des informations en provenance des éléments de détection (capteurs) et du pupitre opérateur (BP). Les sorties transmettent des informations aux pré-actionneurs (relais, Electrovanes ...) et aux éléments de signalisation (voyants) du pupitre. Le nombre de ces entrées et sorties varie suivant le type d'automate. Les cartes d'E/S, ont une modularité de 8, 16 ou 32 voies, Les tensions disponibles sont normalisées (24, 48, 110 ou 230V continu ou alternatif ...).

L'interface réalise trois fonctions principales :

- Le découplage mécanique (borniers à vis par exemple) entre le câblage processus et le câblage interne de l'automate.
- Le découplage électrique (isolation galvanique) : Le problème est de se protéger contre les tensions de mode commun existant non seulement entre les signaux d'entrée et l'automate mais aussi entre les signaux d'entrée eux-mêmes.
- L'adaptation des niveaux de tensions (Par exemple, atténuer les entrées haut niveau hors standards, amplifier les entrées bas niveau, effectuer la transformation courant 'tension)
- La conversion analogique numérique.
- Filtrage des signaux parasites : élimination des parasites industriels de fréquence supérieure à celles du signal utile.
- La synchronisation des transferts conformément aux procédures d'échange du BUS de l'automate.

**D- Modules complémentaires (spéciaux) :**

Les automates compacts permettent de commander des sorties en T.O.R et gèrent parfois des fonctions de comptage et de traitement analogique, les automates modulaires permettent de réaliser de nombreuses autres fonctions grâce à des modules intelligents que l'on dispose sur un ou plusieurs racks, ces modules ont l'avantage de ne pas surcharger le travail de la CPU car ils disposent bien souvent de leur propre processeur, nous pouvons citer les suivants :

- Cartes de comptage rapide.
- Cartes d'entrées / sorties analogiques.
- Cartes de communication (RS485, Ethernet ...).
- Cartes de régulation PID.

**E- Alimentation électrique :**

Tous les automates actuels sont équipés d'une alimentation 240 V, 50 et 60 Hz. 24 V DC. Les entrées sont en 24 V DC et une mise à la terre doit également être prévue.

**F- Les bus :**

Ils sont composés d'un certain nombre de conducteurs (pistes en cuivre) et destinés pour le transport des informations binaires (suite de 0 ou 1) c'est-à-dire (OV ou 5V) sur chaque fil, l'unité centrale de l'API dispose de trois bus :

- **Bus de donnée** : Il permet de véhiculer des données du microprocesseur vers un composant ou d'un composant vers le microprocesseur. Il est donc bidirectionnel.
- **Bus d'adresse** : La mémoire est composée de nombreuses cases mémoires. Chaque case est repérée par une adresse. Le bus adresses est unidirectionnel (du microprocesseur vers les autres composants).
- **Bus de commande**: Il assure la commande des boîtiers mémoires et entrées-sorties par le microprocesseur.

**I-9 Principe de fonctionnement d'un A.P.I :**

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

On définit alors la notion de cycle et de temps de cycle (entre 1ms et 30ms environ), il existe plusieurs types de cycle mais le plus répandu est le celui représenté sur la figure. ce cycle comprend 5 phases :

- **Phase 1** : Lecture ou Acquisition des entrées: Prise en compte des informations des modules d'entrées et écriture de leur valeur dans RAM (zone DONNEE).
- **Phase 2** : Exécution de programme ou traitement des données : Lecture du programme (située dans la RAM programme) par l'unité de traitement. lecture des variables (RAM données), traitement et écriture des variables (internes, sorties ...) dans la RAM données.
- **Phase 3** : Traitement de toute demande de communication.

- **Phase 4** : Exécution du test d'auto--diagnostic (Gestion du système Autocontrôle de l'automate)
- **Phase5** : Ecriture des sorties : Lecture des variables de sorties dans la RAM données et transfert vers le module de sorties.[2]

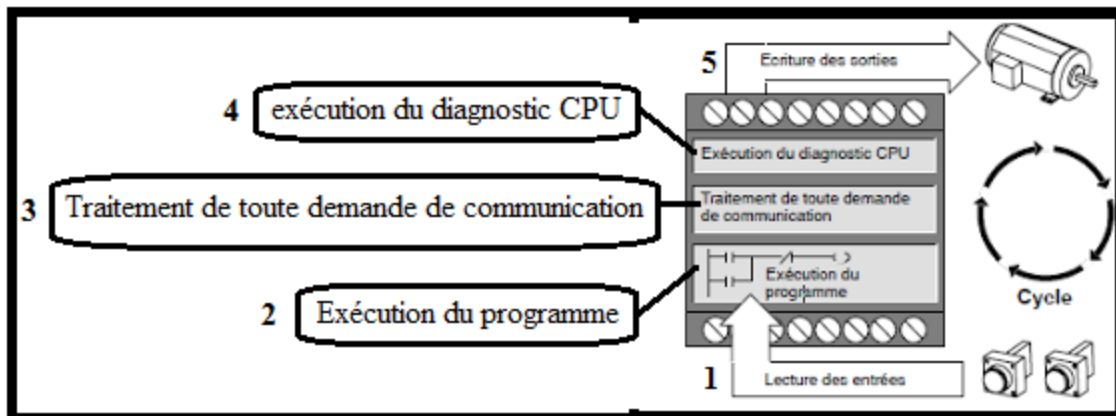


FIG (I- 7) : Le cycle d'un API

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

### I-10 Programmation de l'API:

La norme IEC 1131-3 définit entre autres choses, cinq langages qui peuvent être utilisés pour la programmation d'applications d'automatisme. Les cinq langages sont :

- **SFC** (« séquentiel fonction char ») issu du langage GRAFCET, ce langage, de haut niveau, permet la programmation aisée de tous les procédés séquentiels .
- **FBD** (« function block diagram », ou schéma par blocs) ce langage permet de programmer graphiquement à l'aide de blocs, représentant des variables, des opérateurs ou des fonctions, il permet de manipuler tous les types de variables .
- **LD** (« ladder diagram », ou schéma à relais) ce langage graphique est essentiellement dédié à la programmation d'équations booléennes (true /false) .
- **ST** (« structured text » ou texte structuré) ce langage est un langage textuel de haut niveau. Il permet la programmation de tout type d'algorithme plus ou moins complexe .

- IL (« instruction list », ou liste d'instructions) ce langage textuel de bas niveau est un langage à une instruction par ligne. Il peut être comparé au langage assembleur.[4]

**I-11 Critères de choix d'un type d'automate et le langage:**

Les critères de choix essentiels d'un automate programmable industriel sont :

**I-11-1 Gamme d'un automate :**

pour tenter de rapprocher les automates disposant de caractéristiques similaires, On définit la gamme d'un automate suivant deux notions : quantitative et qualitative.

Classification quantitative (suivant le nombre d'E/S), on a alors :

- Bas de gamme  $20 \leq E+S \leq 100$
- Milieu de gamme  $100 \leq E+S \leq 500$
- Haut de gamme  $500 \leq E+S$

20 étant la limite de rentabilité d'un A PI.

Classification qualitative (relatif aux instructions de calcul ou de traitement numérique) :

- Classe 1 pas de traitement numérique
- Classe 2 traitement numérique simple
- Classe 3 traitement numérique évolué

On définit ainsi deux coefficients :

Iv ( relative à la gamme) :  $Iv = Ub+Yb + \alpha (Ua+ Ya) + T + C$

Avec Ub : nombre d entrées binaires.

Yb : nombre de sorties binaires.

Ua : nombre d'entrées numériques.

Ya : nombre de sorties numériques.

$\alpha$  : taille du mot de données numériques ( en bits).

T : nombre de temporisateurs.

C : nombre de compteur.

Ic ( relative à la classe) :  $Ic = \sum$  des coûts,

Avec :

Nature du traitement	+ ou -	Comparaison sur mot	Comptage temporisation	* ou /	Transcodage	Moyenne	$\sqrt{\quad}$	Régulation
Coût	1	1	1	2	3	3	4	5

Tableau (I-2) : Cout relative à la classe.

D'où on a :	Bas de gamme	$Iv \leq 300$
	Milieu de gamme	$300 \leq Iv \leq 1000$
	Haut de gamme	$Iv \geq 1000$
	Classe 1	$Ic \leq 1$
	Classe 2	$1 \leq Ic \leq 8$
	Classe 3	$Ic \geq 8$

La sélection d'un AP L apte à satisfaire un besoin, dépend de la taille de l'application envisagée (indice  $Iv$ ) et la complexité des traitements numériques (indice  $Ic$ ).

### I-11-2 Critères technologiques :

En plus des critères de gamme et du prix, s'ajoute le critère technologique relatif à :

- La technologie de l'unité centrale . la vitesse de traitement et les fonctions spéciales offertes par le processeur permettront le choix dans la gamme souvent très étendue.
- Le langage de programmation : Un automate utilisant des langages de programmation de type GRAFCET est préférable pour assurer les mises au pont et dépannages dans les meilleures conditions.
- La nature et la taille de la mémoire
- Caractéristiques des Entrées-Sorties
- Fonctions ou modules spéciaux : certaines cartes (commande d'axe, pesage ..) permettront de "soulager" le processeur et devront offrir les caractéristiques souhaitées (résolution,..).
- Fonctions de communication : l'automate doit pouvoir communiquer avec les autres systèmes de commande (API, supervision ...) et offrir des possibilités de communication avec des standards normalisés (Profibus ....).

### I-11-3 Les critères de choix d'un langage de programmation :

- Les éléments techniques : la maîtrise des codes et l'adaptation aux besoins opérationnels.
- Les éléments extra-techniques : l'automaticien veille sur la qualification des équipements et la validation des procédés.

### Conclusion :

Dans ce chapitre on a présenté une étude théorique sur les systèmes automatisés. on a illustré les avantages des automates programmables par rapport aux systèmes câblées.

Vu la souplesse, la rapidité, facilité du diagnostic lors des pannes, l'utilisation d'un outil programmé à base des (API) dans le monde industriel est devenu indispensable.

Dans le chapitre suivant, nous présenterons la presse rotative.

# Chapitre II

## Etude technique de la presse rotative

## Introduction

Ce chapitre présente une étude technique détaillée de la presse rotative et son principe de fonctionnement, nous allons décrire aussi les différentes parties de cette machine

### II-1 Fabrication industrielle de carrelage granito:

Le carreau traditionnel bicouche soi-disant “terrazzo” est formé par une première et une deuxième couche, préparées séparément, qui deviennent après un corps unique au moyen de la pression exercée sur eux par une presse hydraulique.

La première couche est la partie visible du carrelage: elle formée par grenaille de marbre, poudre de marbre ou bien granit et ciment blanc ou gris.

La deuxième couche à la fonction d'absorption de l'eau du mélange composant la première couche et fonctionne comme support à la première couche, elle est composée par du sable, pierre broyée et ciment gris.

La production de carreaux bicouche s'effectue au moyen d'une presse rotative à six/sept stations ou se passent de façon séquentielle toutes les phases à partir de la préparation jusqu'à la formation et démoulage des carreaux.

Après l'étuvage que peut se passer soit de façon naturelle (environ 2 jours dans un lieu couvert ) soit accélérée au moyen de chambres d'étuvage fermées par utilisation de vapeur d'eau (en ce cas le temps peut être réduit 4 a 1 jour), les carreaux sont prêts pour être grésisés et polis au moyen de meules abrasives pour obtenir une surface plane grésisée et polies.

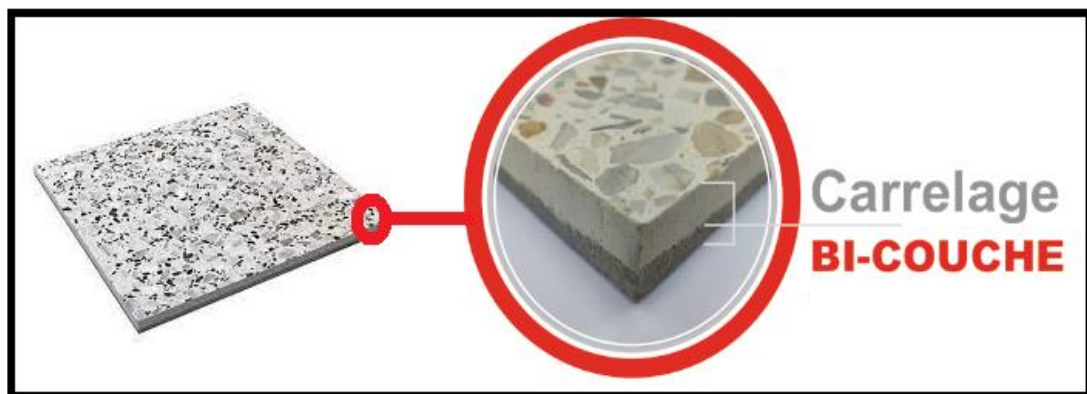


FIG ( II - 1 ) : présentation carrelage bicouche

## II-2 Présentation de la presse rotative:

Notre machine est une presse ROTATIVE MARQUE **O.C.E.M. S.r.l. Officina Costruzioni SERIE OPA 650/300T**.

La Presse Rotative est la machine principale dans le processus de production de carreaux granito, puisque c'est là où on fait le compactage des matériaux de la première et deuxième couche, qui finalement formera le carreau.[6]



FIG ( II - 2 ) : La presse rotative a carreaux.

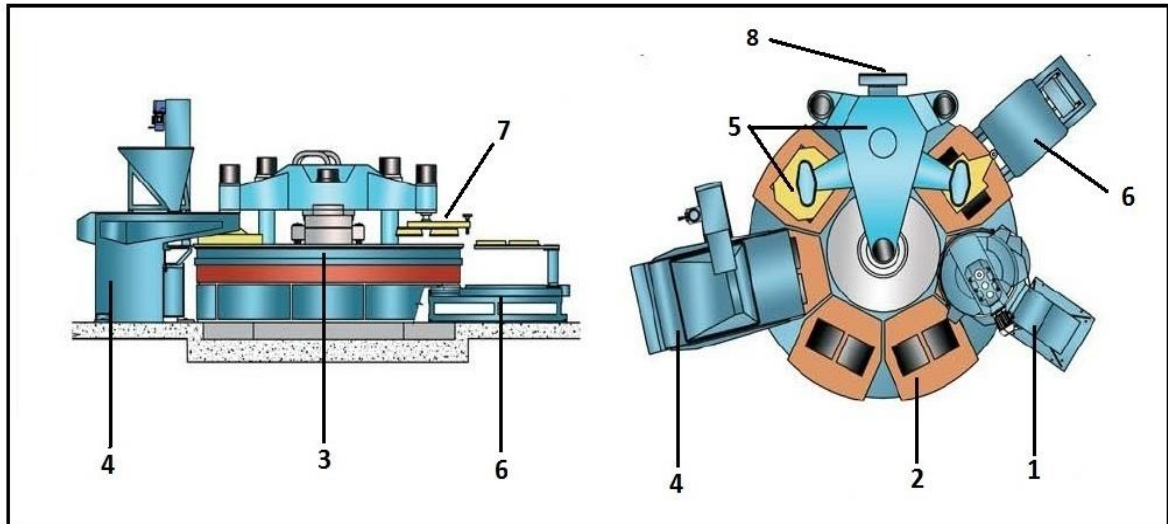
## II-3 les processus de production:

le fonctionnement de la presse rotative en six étapes :

- **étape 1** : Déchargement du mélange de la première couche, du doseur au moule.
- **étape 2** : Etendu du mélange dans le moule moyennant de vibreur.
- **étape 3** : Rotation de la table avec un système de centrage et de freinage de la table Moyennant de vérin pneumatique.
- **étape 4** : Chargement de la deuxième couche par un trémie avec racleur simple.

- **étape 5** : Option d'un système de pré-pressage et Pressage final du carreau pour son correcte compactage.
- **étape 6** : Démoulage et extraction du carreau moyennant le chariot extracteur.[6]

**II-4 Identification des parties composantes de la presse:**



- |             |                       |                   |
|-------------|-----------------------|-------------------|
| 1- DOSEUR   | 2- MOULE VIBRANTE     | 3- TABLE ROTATIVE |
| 4- TREMIE   | 5- PRESSE HYDRAULIQUE | 6- CHARIOT        |
| 7-DEMOULEUR | 8-FREIN DE TABLE      |                   |

FIG ( II - 3 ) : Les parties composantes de la presse

**II-5 Le doseur :**

La fonction du doseur est de charger de manière automatique de la première couche de carreaux dans les cavités des moules de la presse.

**II-5-1 Centrale oléodynamique:**

La centrale oléodynamique est dotée d'une pompe double, d'un distributeur commande électriquement, d'une vanne de séquence, d'une vanne de réglage de la pression et d'un moteur électrique pour l'alimentation du cylindre de coupage, qui provoque la rotation alternée de la plaque de coupage de la position de chargement a la position de déchargement.[6]

### II-5-2 Equipement pneumatique:

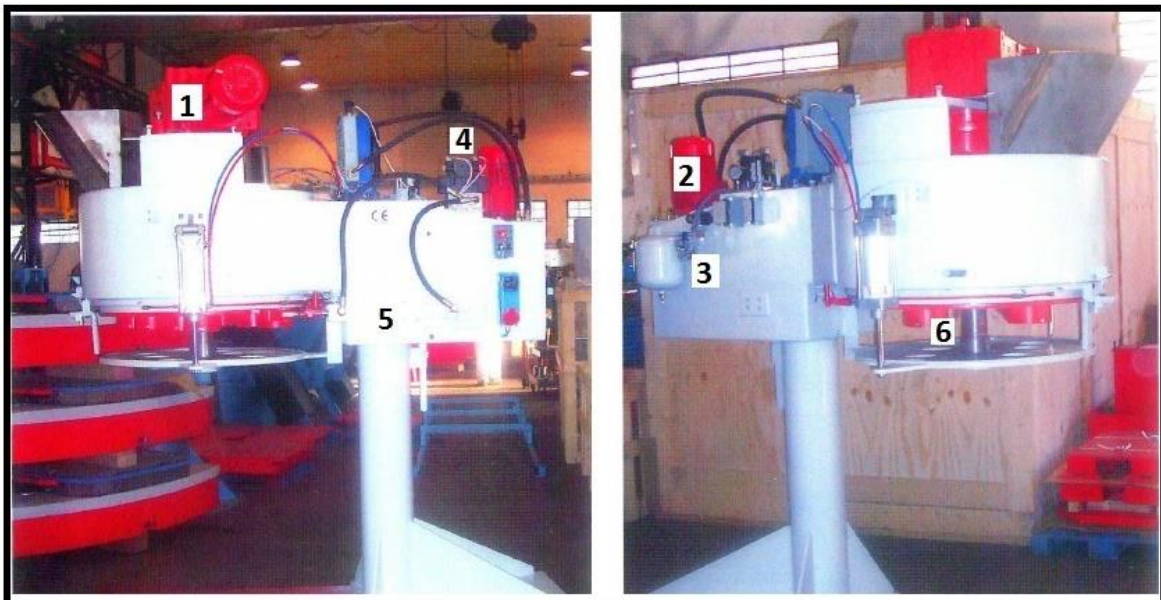
Le déchargement du mélange des douilles de dosage est effectué au moyen de pulvérisateurs d'air comprimé (un par douille) alimentés par un poumon d'air au moyen d'une vanne électrique.

### II-5-3 Agitateur:

A l'intérieur du réservoir, à une hauteur réglable à partir du fond, tournent 6 palettes montées sur un dispositif planétaire commandé par un motoréducteur à engrenages a deux vitesses de rotation, Les 4 palettes planétaires et les 2 palettes périphériques servent a homogénéiser le mélange contenu dans le réservoir et a achever le chargement des douilles de dosage lorsqu'elles se trouvent dans la position de chargement.

### II-5-4 Sonde électrique de contrôle de niveau:

Le doseur est dotée d'une sonde qui contrôle le niveau du mélange contenu dans le réservoir. La sonde peut commander un tapis qui alimente le doseur lorsque le niveau du mélange baisse et arrête le tapis lorsque le mélange atteint le niveau maximum préétabli. Grace a la sonde, le mélange est toujours homogène et régulier.[6]



1- motoréducteur de l'agitateur  
2- moteur centrale hydraulique

3-électrovane de pulvérisateur d'air  
4- distributeur

5- cylindre de coupage  
6 - douille de dosage

FIG ( II - 4 ) : Une photo prise de doseur

## II-6 La moule vibrante :

### **A- Fermeture des moules pendant la vibration et le pressage:**

La fermeture et l'ouverture des moules est réalisée au moyen d'un système pneumatique ( vérins pneumatiques.), ce qui présente les avantages suivants :

- elle est réglable en fonction de la dimension des moules et de l'état des fonds de moule.
- elle est rapide en raison du fait qu'elle peut être exécutée pendant la rotation de la table.
- elle est simple.

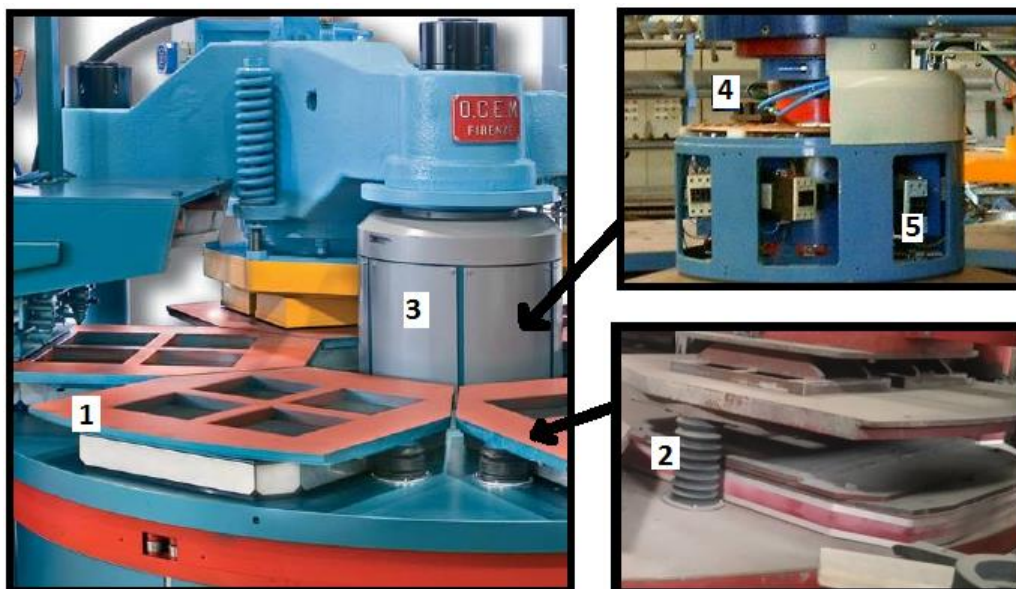
### **B- Vibration de la première couche:**

Afin de permettre une répartition parfaite du mélange à l'intérieur du moule, les moules et les stations vibrantes doivent être rigoureusement nivelés et parfaitement réglés.

Une répartition correcte de la première couche permet également :

- une répartition uniforme des granulats.
- une réduction des bulles d'air dans le mélange.
- une épaisseur uniforme de la première couche.
- le dépôt de presque tous les granulats sur la surface.

Les vibreurs sont alimentés par les pistes porte-courant montées dans le carter central de la table, par l'intermédiaire des contacteurs alimentés par les charbons électriques.



1- Moule vibrante                      2- vérin pneumatique                      3- Le carter central  
4- Les charbon électrique                      5- contacteur

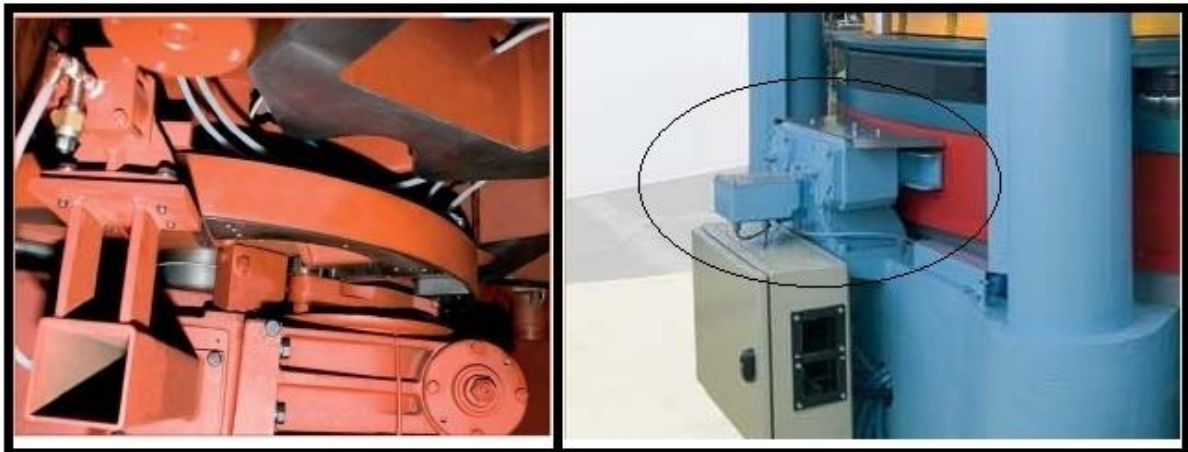
FIG ( II - 5 ) : Une photo prise de moule vibrante

Généralement, l'épaisseur de la première couche:

- n'est pas inférieure au tiers de la hauteur du moule.
- n'est pas supérieure à la moitié de la hauteur du moule.

### **II-7 La table rotative :**

la rotation de la table effectuée avec un mécanisme à croix de Malte, le mécanisme est commandé par un motoréducteur, lorsque la table se trouve dans la station, le réducteur de croix de Malte tourne jusqu'à ce qu'il finisse sa course de centrage, le freinage de la table rotative est commandé par un électrovanne pneumatique.



1- croix de malte

2- vérin freinage table

FIG ( II - 6 ) :système croix de malte et freinage table

### **II-8 Le trémie :**

Le mouvement avant-arrière du tiroir de la trémie est effectué au moyen d'un réducteur à vis sans fin ( FIG(II-7) ), La course du tiroir peut être simple ou double, La vitesse du mouvement du tiroir de la trémie est réglée au moyen d'un inverter ( variateur de fréquence) qui permet de varier le nombre de tours du moteur et donc sa vitesse..

Un certain nombre de paramètres (P sur l'inverter) est fixé par OCEM pendant les essais de mise en place de l'unité, mais ils peuvent être modifiés dans certaines limites imposés par OCEM.



1- vibreur fix trémie 2-tiroir trémie 3- réducteur à vis sans fin

FIG ( II - 7 ) : Une photo prise de trémie

### **II-9 Centrage de la table et pressage oléodynamique :**

La pompe de pressage fonctionne d'une manière continue. La pression du circuit hydraulique peut être réglée entre 0 et 200 bar au moyen des vannes de réglage, en fonction du produit à réaliser, de l'épaisseur du carreau, de la granulométrie de la matière première et de la surface totale du carreau dans le moule.

L'opération de deuxième pressage est exécutée comme suite :

Lorsque la table s'arrête dans la station, la tige de centrage de la table commande l'électrovanne pneumatique qui ouvre le distributeur hydraulique de la pompe.

A la fin du temps de pressage prédéterminé, le distributeur se ferme et l'huile du cylindre de pressage retourne dans la centrale hydraulique à travers d'un filtre à huile et système de refroidissement eau -huile .

Quatre ressorts de retour soulèvent le piston de pressage (vérin simple effet) et les tampons de pressage.



1- centrale hydraulique      3- piston de pressage      5- pompe hydraulique  
 2- vérin centrage table      4- distributeur hydraulique      6 - moteur électrique

FIG ( II - 8 ) : Système pressage des carreaux et centrage la table

### II-10 Chariot extracteur et démouleur :

Le mouvement du chariot extracteur est synchronisé avec le démouleur. Pendant la course en avant du chariot, le bras porte-carreaux du chariot se lève pneumatiquement de manière à se rapprocher des carreaux contenus dans le moule.

A la fin de la course en avant du char

iot, le démouleur descend, commandé par l'électrovanne, la descente du démouleur terminée, la fin de course commande la montée du démouleur, ce dernier remonte dans sa position de repos.[6]

#### **II-10-1 Mouvement de la pelle :**

La pelle monte lorsque le chariot entre, aux fins d'approcher les plateaux porte-carreaux du moule, de manière à éviter toute chute des carreaux pendant le démoulage, La pelle monte lorsque l'électrovanne est excitée, l'électrovanne reste excitée pendant la course en avant du chariot, a la fin de course désexcite d'électrovanne , mais la pelle reste en haut. L'électrovanne abaisse la pelle lorsque le démouleur à effectué sa course vers le bas .

La vitesse de montée/descente de la pelle est réglée au moyen des régulateurs de débit d'air placés sur le bâti du chariot.[6]

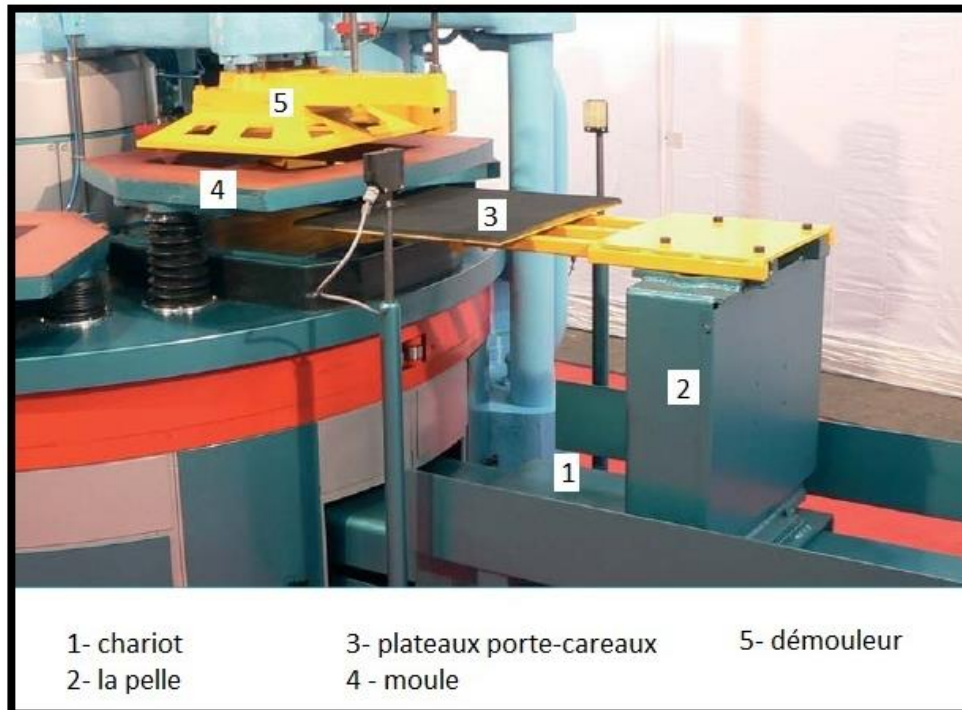


FIG ( II - 9 ) : chariot et démouleur

**II-11 Caractéristiques de la machine :**

**Dimensions et poids:**

Corps de la presse    Longueur : 3.200 mm.  
                                  Largeur : 2.450 mm.  
                                  Hauteur : 2.350 mm.  
                                  Poids : 10.800 kg

Poids total de la presse (moules exclus) : 15.000 kg.

Doseur                      Poids : 1.500 kg.  
Série de moules        Poids : 1.500 kg.

**Puissance installée:**

Presse                      Centrale hydraulique :                      15,1 kw.  
                                  Rotation de la table :                      1,1 kw.

	Vibreux des moules (chacun)	0,8 kw
	Trémie de la deuxième couche	1,1 kw
	Chariot extracteur	1,5 kw
Doseur	Agitateur :	2,2 kw
	Central hydraulique :	4 kw

## II-12 Partie électrique :

Dans cette étude, nous avons discuté de l'étude électrique de la machine, du lieu de distribution d'énergie et du panneau de commande qui fait fonctionner la presse.

La machine de et discuter des principaux électrique composants qui sont expliqués dans les points : l'alimentation , l'armoire électrique, Panneau de commande , fin de course.

### **II -12-1 Alimentation:**

A partir de ta TGBT (Tableau Générale basse tension), on alimente l'armoire électrique de la presse ,l'armoire électrique est connectée a la boite générale de dérivation de la presse au moyen de câble, le corps de la presse est connecté au moyen de câble a la trémie ,au chariot ,a la centrale hydraulique et au doseur.

### **II-12-2 Armoire électrique:**

Elle contient tous les équipements électriques nécessaires aux fonctionnements et la protection de la presse tels que, les sectionneurs, disjoncteurs, contacteurs, relais...etc.



FIG ( II - 10 ) : Armoire électrique de la presse

Les composants de l'armoire électrique sont organisés par symboles pour chaque composant.

La commande électrique actuelle de la presse rotative est une technologie commande à relais et contacteurs ( logique câblée ).

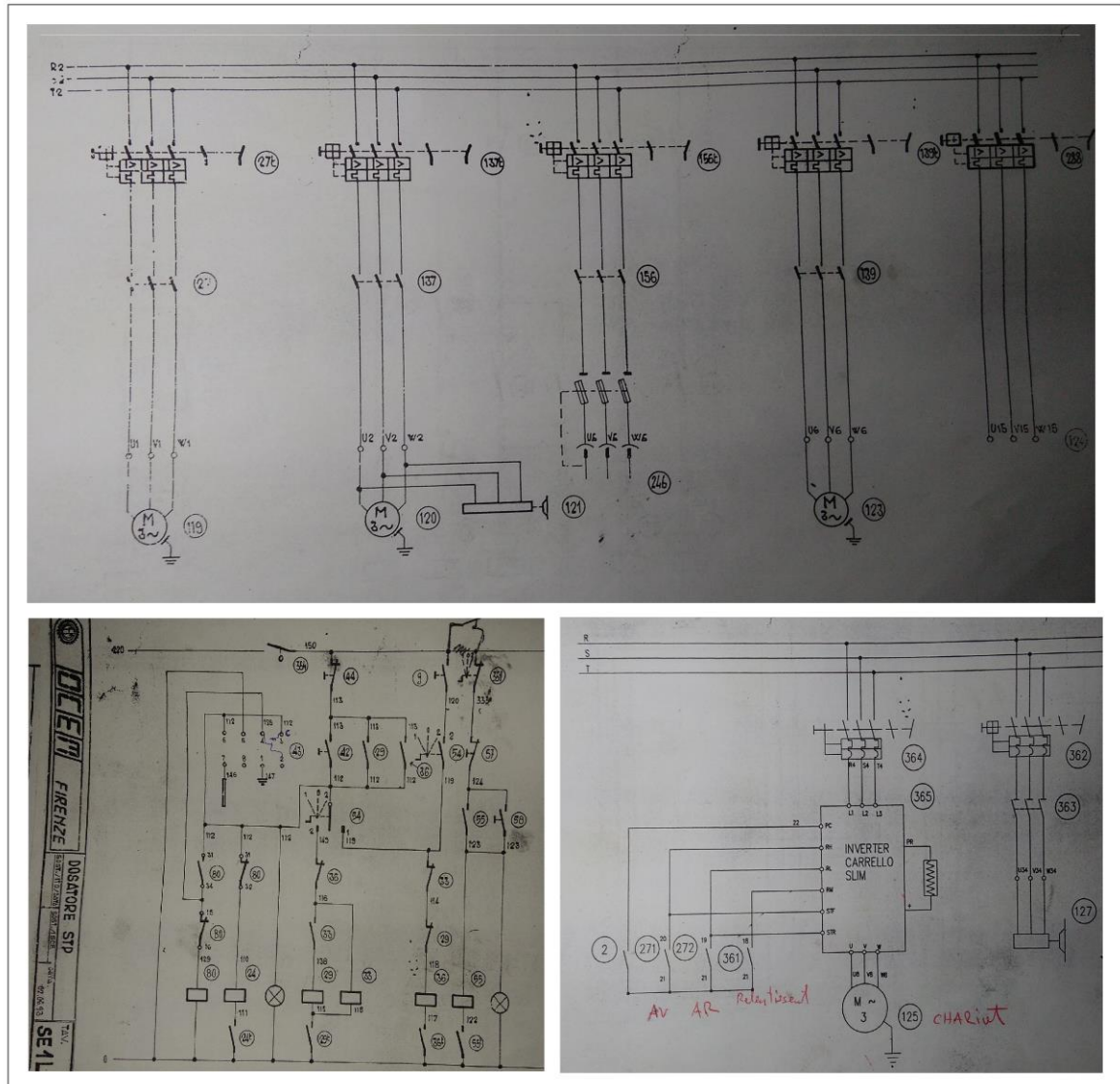


FIG ( II - 11 ) : Quelques photos du schéma électrique de la presse.

**II-12-3 Panneau de commande :**

Il contient les clés de fonctionnement de la machine ( selecteurs et boutons pousiore ), il se compose d'une clé et d'un symbole pour chaque opérateur ainsi que pour l'armoire électrique et ses composants.

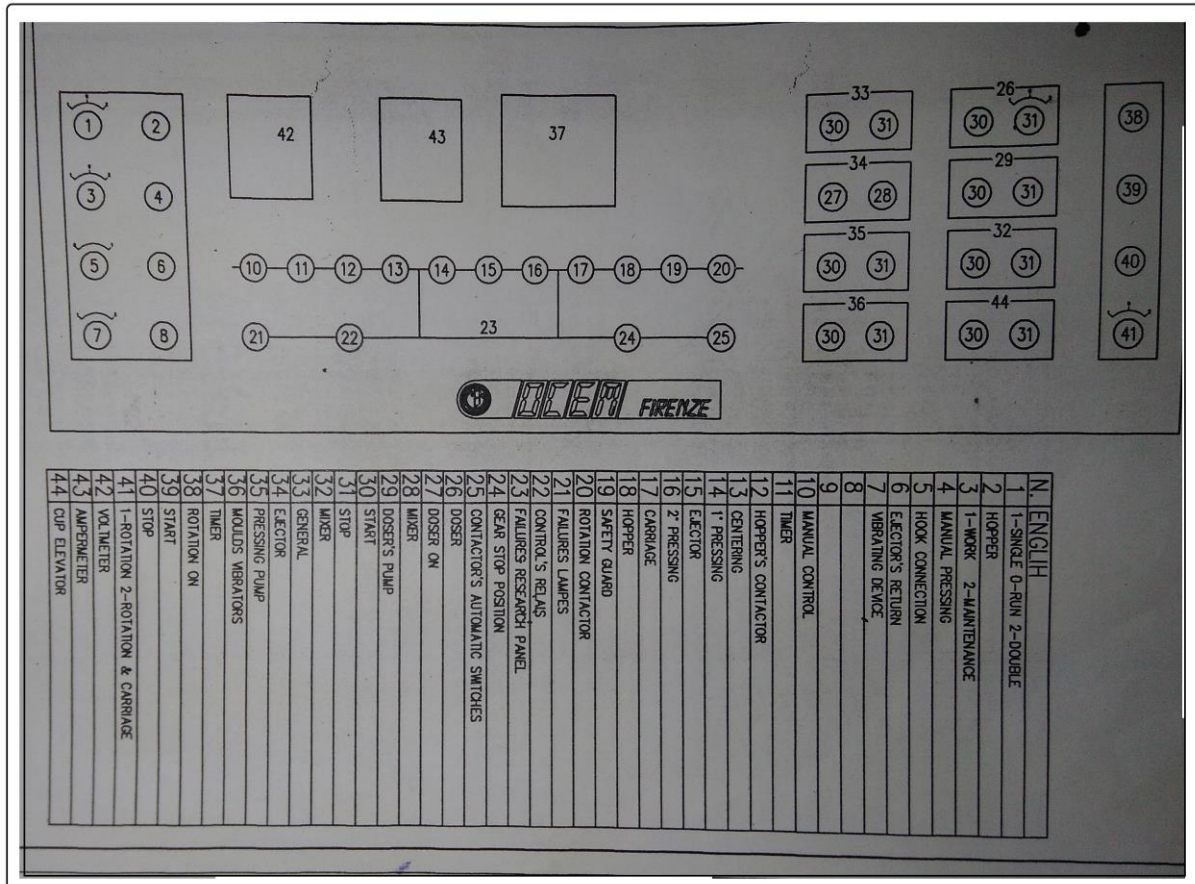


FIG ( II - 12 ) : Schéma du panneau commande de la presse.

**II-12-4 Fin de course :**

les fins de course utilisée dans la presse sont :

- Les interrupteurs de position.
- Les capteurs inductifs.
- Les détecteurs photoélectriques.

**Interrupteurs de position:** sont employées dans la presse pour assurer la fonction détecter les positions.

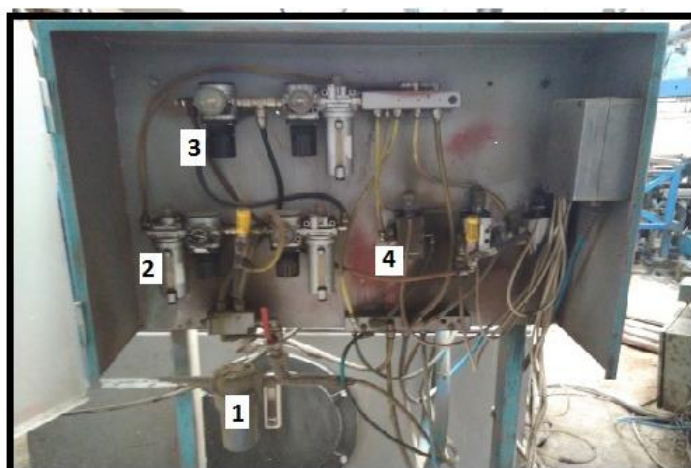
**Capteur inductif :** Cette capteur inductif est utilisés pour effectuer des mesures de position, Il envoie un champ magnétique pour arrêter le bras sans contact ni poussée.

**Les détecteurs photos-électriques :** portent aussi le nom de barrières lumineuses, ils sont de technologie électronique et délivrent une information ( 0 ou 1 ) chaque fois que le faisceau issu de la partie émettrice est interrompu par un obstacle quelconque occultant la partie réceptrice.

### II-13 Partie pneumatique :

La machine doit être alimentée par le réseau externe d'air comprimé. Le branchement est effectué dans la cabine pneumatique au moyen d'un raccord, L'air comprimé doit être le plus propre et sec possible. En conséquence, il doit entrer dans le circuit de la presse à travers un filtre doté d'un système de décharge de l'eau condensée. La pression pneumatique ne doit jamais être inférieure à 8 bar. Les valeurs correctes de la pression pour les différents dispositifs/mouvements sont :

MOUVEMENTS GENERAUX	: 4 -6 bar
FREINAGE DE LA TABLE	: 3 -4 bar
FERMETURE DES MOULES	: 4 -6 bar
RACLEUR DE LA TREMIE	: 3 -4 bar
PELLE DU CHARIOT	: 3 -6 bar



1- filtre +décharge d'eau      3- régulateur de pression  
2- lubrificateur                4- distributeur

FIG (II - 13 ) : La cabine pneumatique.

### II-14 Partie hydraulique :

Les centrales hydrauliques de la presse (FIG (II - 12 ) ) et du doseur sont normalement dotées d'échangeurs de chaleur eau-huile pour le refroidissement ,La pression de l'eau doit être de 1 - 1,5 bar.

Les dispositifs hydrauliques de la presse règlent la pression des différents dispositifs/mouvements entre les valeurs suivantes :

PREKIER PRESSAGE : NORMAL 150 BAR  
 DEUXIEME PRESSAGE : NORMAL 180 BAR  
 VERIN DU DOSEUR : NORMAL 100 BAR

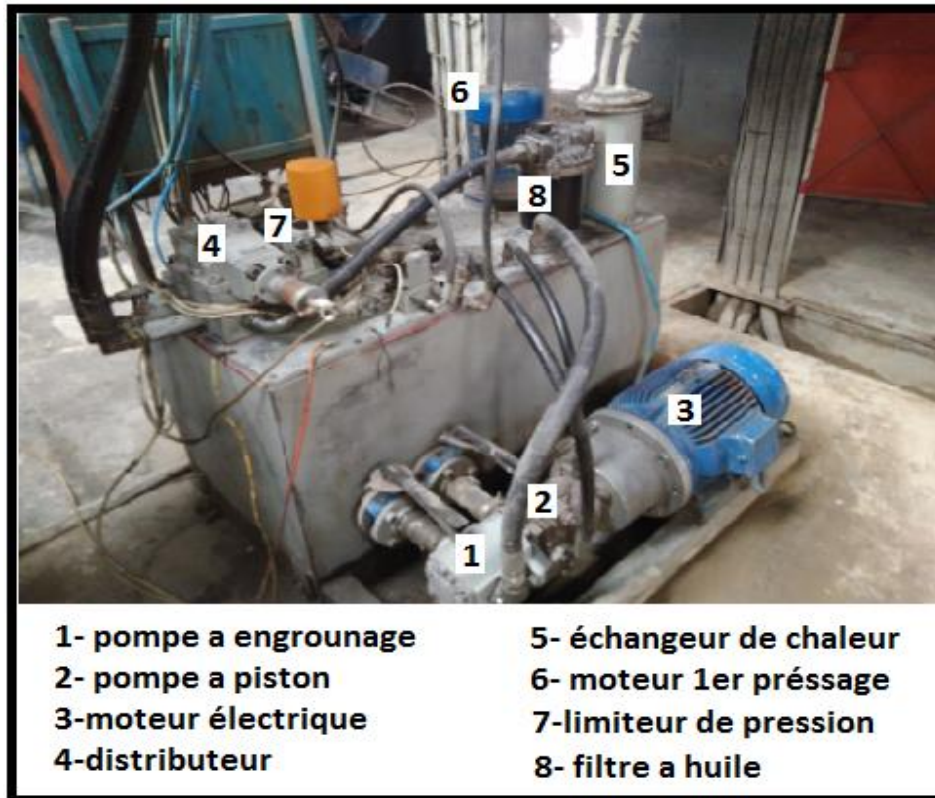


FIG (II - 14 ) : centrale hydraulique de pressage

La valeur de la pression dans le circuit hydraulique du deuxième pressage doit être réglée en fonction des éléments suivants :

- dimension totale de la surface à presser dans le moule (surface de chaque dalle multipliée par le nombre de dalles dans le moule).
- épaisseur des carreaux.
- caractéristiques du matériau utilisé, en particulier dans la première couche (granulométrie, résistance, humidité, etc.).

Le pressage est normalement effectué à une pression de 70 kg/cm<sup>2</sup> pour le "granito" à 120 kg/cm<sup>2</sup> et plus pour certaines dalles d'extérieur.

**II-15 Problématique :**

La commande électrique actuelle de la presse rotative est une technologie commande à relais et contacteurs ( logique câblée ) qui présente les inconvénients.

Le tableau (Tableau ( II-1 ) ) suivent présentée la comparaison entre la logique câblée et l'automate programmable :

logique câblée	Automate programmable
<ul style="list-style-type: none"> <li>• L'armoire est très encombrée et d'un accès très difficile.</li> <li>• Les composants électriques sont obsolètes et volumineux et d'un approvisionnement difficile.</li> <li>• Connexion figée, n'autorisant pas un développement de l'application.</li> <li>• Mise au point difficile.</li> <li>• Plus de consommation d'énergie relativement aux commandes et technologie électrique numérique et par conséquent un échauffement de l'armoire de commande.</li> <li>• Les relais électromagnétiques sont soumis à un fort cycle de commutation et une usure importante et un fort taux de pannes.</li> <li>• Le traitement de l'information se fait lentement à cause du temps de commutation des relais.</li> <li>• L'environnement de poussière et sable (Sahara) réduit la durée de leur fonctionnement des contacteurs et relais électromagnétique utilisée dans la commande électrique.</li> </ul>	<ul style="list-style-type: none"> <li>• Le programme est établi à partir d'un langage compris puis rangé suivant un ordre dans les mémoires.</li> <li>• Ce programme est divisé en une succession d'ordres. Les fonctions logiques ne sont pas dépendantes du câblage mais sont reliées entre elles par un câblage binaire.</li> <li>• Grâce à l'API, l'adaptation à une nouvelle fabrication est rapide.</li> <li>• Moins coûteuse.</li> <li>• L'automate étant ; modifiable à volonté, il est programmable.</li> <li>• La flexibilité, la souplesse vis-à-vis de la mise au point de commande et de l'évolution de celles-ci et la performance font de la logique programmée, une solution préférable.</li> <li>• Toutes les connexions sont en soft (logique).</li> <li>• Diagnostic rapide :(recherche d'éventuelles pannes).</li> <li>• Réduction du volume des composants</li> </ul>

Tableau ( II-1 ) : la comparaison entre la logique câblée et l'automate programmable

**Conclusion :**

L'expérience a montré que la commande des processus industriels plus ou moins complexes par la logique câblée a généré plusieurs difficultés à cause de leur encombrement et leur manque de souplesse et c'est pour ces raisons. La partie commande de la presse rotative se trouve obligée de s'orienter vers des solutions programmées qui offrent beaucoup plus de facilité de mise en œuvre, d'entretien, de souplesse et elles sont surtout moins encombrantes, moins coûteuses et nécessitent moins d'intervention humaine.

# Chapitre III

## Automatisation de la presse avec l'API S7 300

**Introduction :**

Après avoir effectué une description complète de la presse rotative dans le chapitre précédent et bien détaillé son principe de fonctionnement, dans ce chapitre nous décrirons l'automate choisi, les différentes parties qui le constitue ainsi qu'une présentation du logiciel qui permet sa configuration et sa programmation, puis nous procédons dans ce chapitre à l'élaboration d'une programmation relative aux modifications que nous avons proposés.

**III-1 Présentation de la gamme SIMATIC de SIEMENS:**

Siemens propose une large gamme de produits pour l'automatisation industrielle, par le biais de sa gamme SIMATIC, L'intégration globale de tout l'environnement d'automatisation est réalisée grâce à :

- Une configuration et une programmation homogène des différentes unités du système.
- Une gestion cohérente des données.
- Une communication globale entre tous les équipements d'automatisme mise en œuvre.

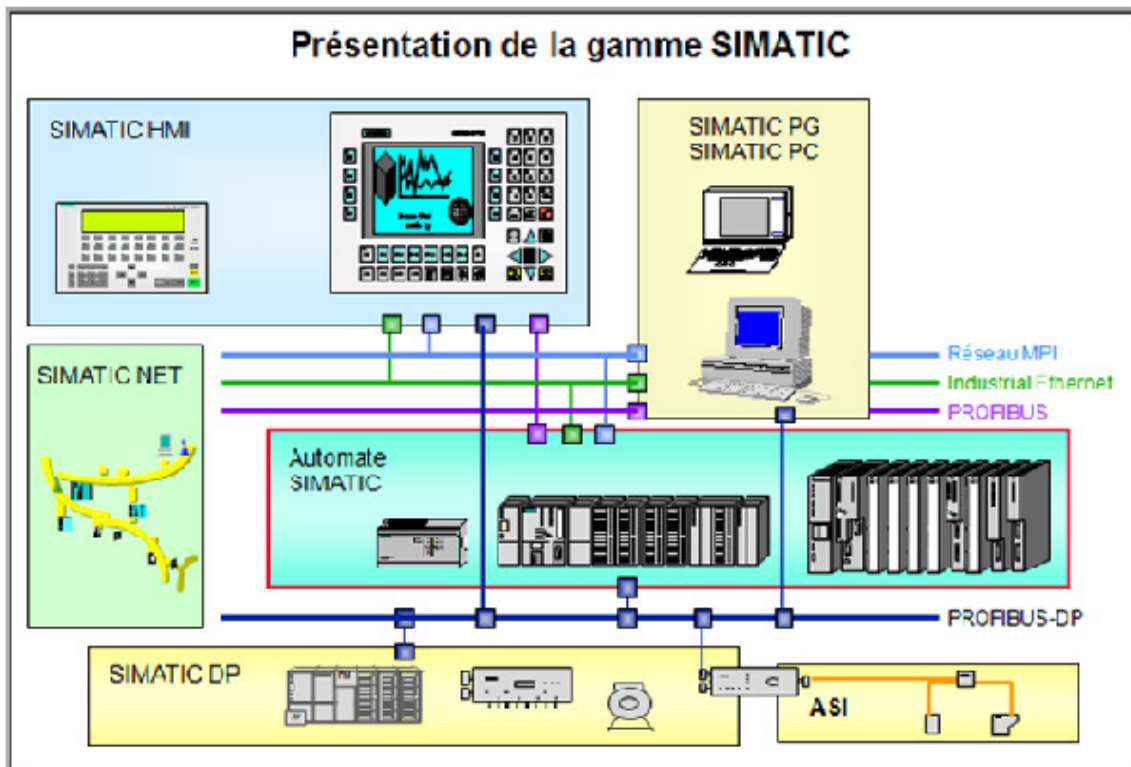


FIG ( III-1) : Présentation de la gamme SIMATIC

## **III-2 Les différentes variantes dans la gamme SIMATIC:**

### **III-2-1 SIMATIC S7:**

Cette gamme d'automates comporte trois familles :

-**S7-200** qui est un Micro-automate modulaire pour les applications simples, avec possibilité d'extensions jusqu'à 7 modules, et une mise en réseau par l'interface multipoint (MPI) ou PROFIBUS.

-**S7-300** est un Mini-automate modulaire pour les applications d'entrée et de milieu de gamme, avec possibilité d'extensions jusqu'à 32 modules, et une mise en réseau par l'interface multipoint (MPI). PROFIBUS et Industriel Ethernet.

-**S7-400** est un automate de haute performance pour les applications de milieu et haut de gamme, avec possibilité d'extension à plus de 300 modules, et une possibilité de mise en réseau par l'interface multipoint (MPI), PROFIBUS ou Industriel Ethernet.

### **III-2-2 SIMATIC C7:**

Le SIMATIC C7 combine automate programmable et panneau opérateur dans une seule unité, L'automate compte la CPU, les modules d'entrées: sorties. et le panneau opérateur qui est utilisé conune une interface Homme Machine ( HMI ).

### **III-2-3 SIMATIC M7:**

Les SIMATIC M7 sont des calculateurs industriels compatibles PC. Il s'agit d'un système modulaire sous boîtier, construit dans la technique des automates SIMATIC S7, il peut être intégré dans un automate S7 300400 ou être utilisé comme système autonome avec une périphérie choisie dans la gamme S7.[9]

## **III-3 Choix d'automate:**

Après une analyse bien détaillée et recensement des Entrées Sorties de chaque tache de la station, on a plus de 100 Entrées et Sorties, on va choisir l'automate S7-300 selon les avantages suivants :

- Une construction compacte et modulaire, libre de contraintes de configuration.
- Une riche gamme de modules adaptés à tous les besoins du marché est utilisable en architecture centralisée ou décentralisée.

- Une large gamme de CPU a été adaptée à toutes les demandes de performances pour pouvoir d'obtenir des temps de cycle machines courts, certaines étant dotées de fonctions technologiques intégrées comme par ex. le comptage, la régulation ou le positionnement.
- Une économie d'ingénierie en utilisant les outils orientés application et normalisés CEI 1131-3 tels que les langages évolués SCL ou des logiciels exécutifs orientés technologie pour le contrôle des mouvements.

**III-4 Présentation de l'automate S7-300:**

L'automate programmable S7-300 est un automate modulaire peut comporter des modules PS (alimentation), CPU (unité centrale), SM (module de signaux d'entrées sorties), modules de fonctions FM pour les fonctions spéciales (par ex. : commande de moteur pas à pas), processeurs de communication CP pour les liaisons réseau, Chaque module est repéré par un emplacement.[7]

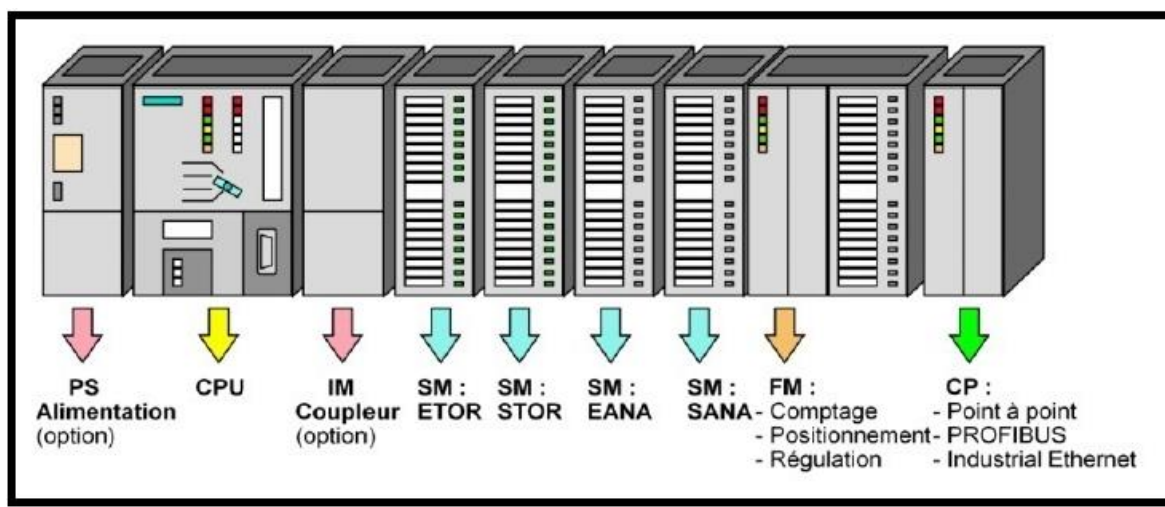


FIG ( III-2) : L'automate programmable s7-300

**III-4-1 Module d'alimentation (PS) :**

Il convertit la tension secteur (AC 120:230 V) en une tension d'alimentation (24 VCC) pour les modules de l'automate, Les tensions pour les capteurs, CPU, actionneurs....qui dépassent le 24 V sont fournies par des blocs d'alimentation ou transformateurs supplémentaires.

Le module d'alimentation utilisé dans ce projet est : PS 307 2A .

### III-4-2 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 permettre aussi le réglage du comportement au démarrage et le diagnostic des défauts par les LEDS.

Elle est disponible en différentes versions dont certains :

- A utilisateur standard : CPU 313. CPU 314. CPU 315. CPU 316.
- Avec fonctions intégrées : CPU 312 IFM. CPU 314 IFM.
- Avec interfaces PROFIBUS-DP : CPU 315-DP. CPU 316-2 DP. CPU 318-2 DP.

La CPU 313 (FIG ( III-3) ) utilisée dans notre étude est caractérisée par :

- Une mémoire de travail RAM 96 KO.
- Un logement pour une cartouche mémoire EPROM. EEPROM.
- Temps de traitement 0.1 ms:'kilo-instructions.
- Un connecteur pour console de programmation. terminal de dialogue (MPI). bus.
- Un connecteur pour le bus interne.
- Un logement pour une pile de sauvegarde.

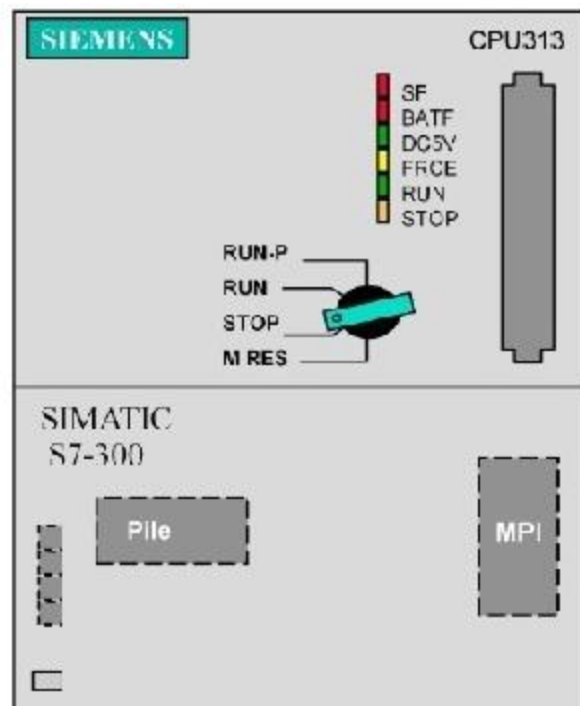


FIG ( III-3) : L'unité centrale CPU313.

Le module CPU comporte les éléments suivants la FIG(III-3) et les caractéristique sont données dans le :

**A-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.

**B- 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 programune est exécuté. accès en écriture et en lecture avec une PG : la CPU traite le programme.

**C- Connexion MPI:**

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

**D- 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.

**E- 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-4-3 Carte couplage IM:**

Les coupleurs permettent de disposer d'une configuration à plusieurs châssis .

**III-4-4 Module de signaux (SM):**

Ils servent d'interface entre le processus et l'automate, il existe des modules d'entrée TOR. des modules de sortie TOR ainsi que des modules d'entrée et de sortie analogiques. Les modules d'entrée sortie sont des interfaces entre les capteurs et les actionneurs d'une machine.

Après le recensement des Entrées / Sorties de notre processus on a trouvé Les entrées Logiques :  $74 + 20\%$  (réserves) = 86, Les sorties Logiques :  $37 + 20\%$  (réserves) = 40.

**A- Les entrées Tout Ou Rien (TOR) :**

Les modules d'entrée tout ou rien permettent de raccorder a l'automate les différents capteurs logiques, elles assurent l'adaptation.,l'isolement,le filtrage et la mise en forme des signaux électroniques, l'état de chaque entrée est donné par une diode électroluminescente situant sur la carte, le nombre d'entrées sur une carte est de: 16 31.les tensions d'entrées sont de : 24. 48. 110. 220 volts en courant continu ou alternatif

**B- Les sorties Tout Ou Rien :**

Les modules de sorties tout ou rien permettent de raccorder a l'automate les différents prés actionneurs, les tensions de sorties usuelles sont de 5. 24. 48. 110 ou 220 volts en continu ou en alternatif, Les courants vont de quelque mA a quelque Ampères, Ces modules possèdent des relais ou bien des triacs des transistors. L'état de chaque sortie est visualisé par une diode électroluminescente.

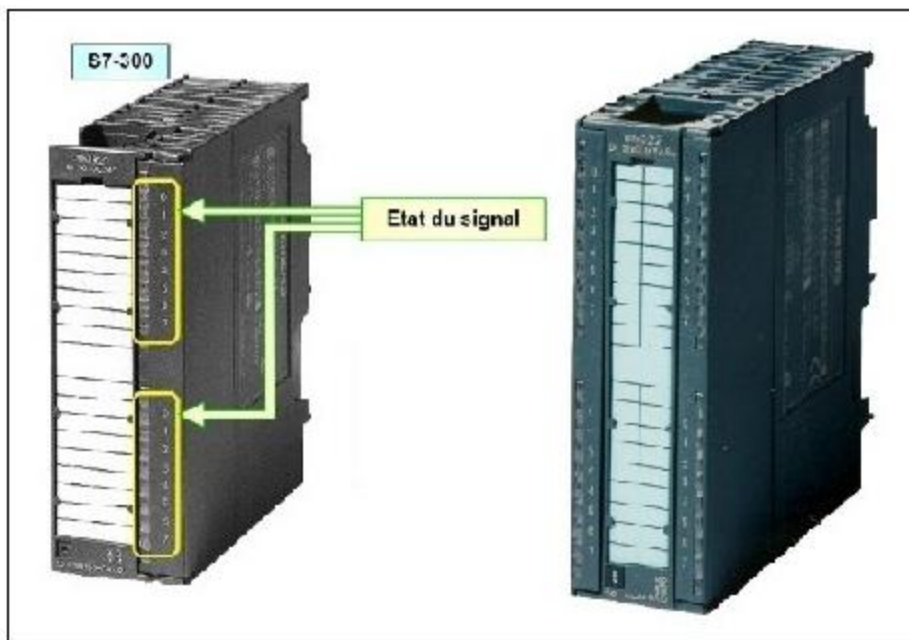


FIG ( III-4) : Module de signaux (SM)

#### III-4-5 Capacité d'extension du S7-300:

La figure( III- 5 ) montre un S7-300 en configuration maximale,La configuration globale admet jusqu'à 32 modules, 8 modules max par châssis (rail profilé), Aucune règle d'emplacement ne s'applique pour les modules de signaux, les modules de fonction et les processeurs de communication, ils peuvent donc être montés à un emplacement quelconque ,les coupleurs IM permettent de boucler le bus entre les différents châssis, le coupleur IMS représente l'émetteur et IMR le récepteur, les coupleurs doivent être montés aux emplacements prévus.[10]

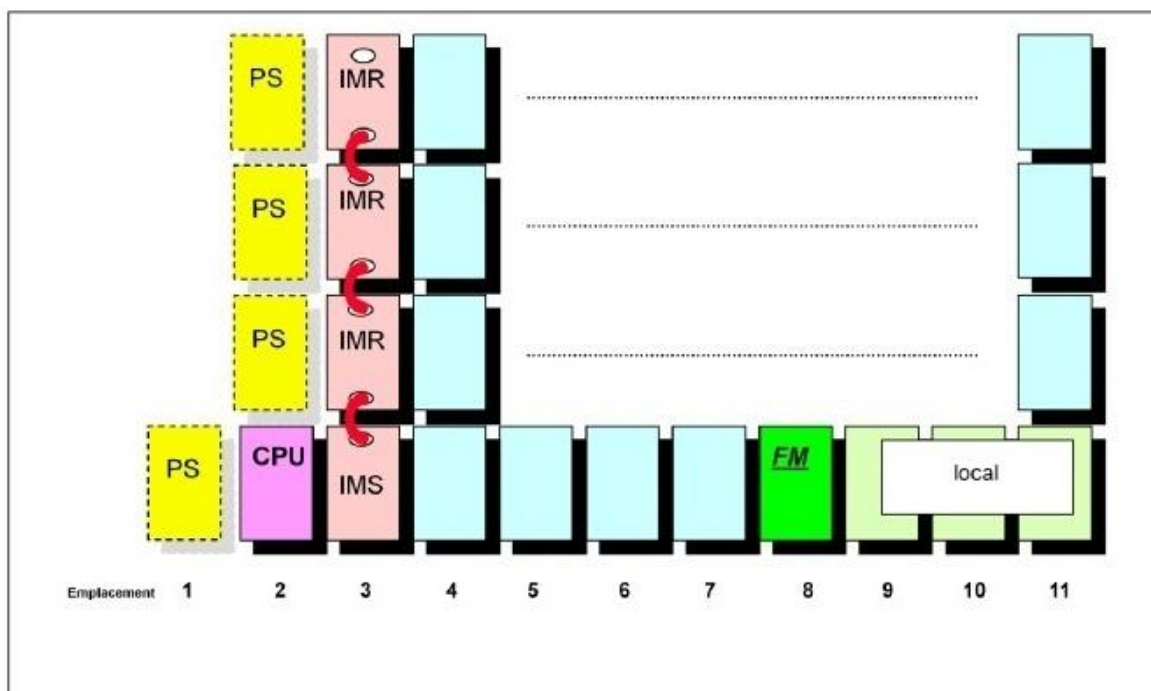


FIG ( III-5) : Capacité d'extension du S7-300.

Les règles suivantes valent pour l'enfichage :

Emplacements 1 à 3 (affectation fixe):

- Emplacement 1 : PS (alimentation), si disponible.
- Emplacement 2 : CPU (unité centrale), si disponible.
- Emplacement 3 : IM (coupleur), si disponible.

Emplacements 4 à 11 (affectation libre):

SM, FM, CP enfichables à un emplacement quelconque.

### III-5 Matériel utilisé:

Les choix du matériel est base d'une part, sur les équipements déjà existant au de la presse, et d'autre part sur le nombre Entrées Sorties et sur la valeur de la tension d'alimentation au niveau de l'armoire de commande, Les équipements choisis sont :

- 01 alimentation: PS307 2A .
- 01 unité de traitement : CPU 313.

- 03 modules de Entrées digitale : DI 32.DC 24V .
- 02 modules de Sorties digitale : DO32.DC 24V 0.8A .

### III-6 Console de programmation:

Le programme d'automatisation est créé au PC ( CONSOLE DE PROGRAMATION ) avec le logiciel STEP 7 et y est temporairement enregistré,une fois que vous avez relié le PC à l'interface MPI de l'automate, le programme peut alors être chargé à l'aide de la fonction de chargement dans la mémoire de chargement de l'automate (FIG ( III-6) ).

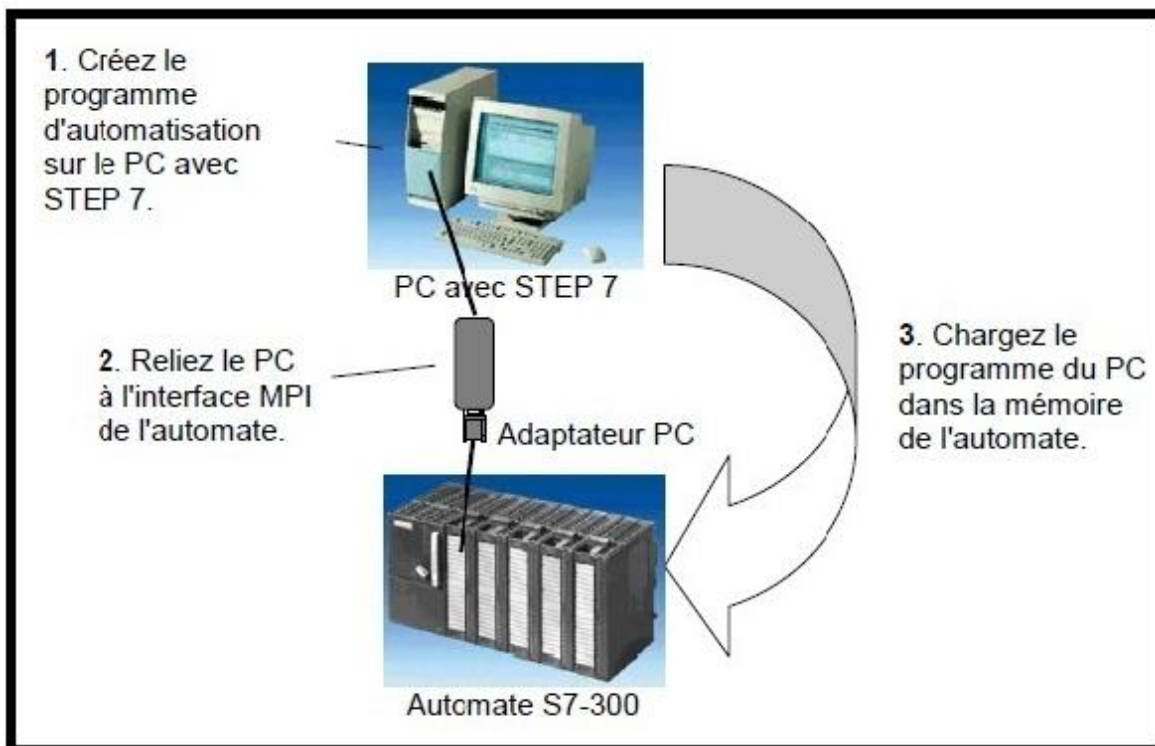


FIG ( III-6) : Console de programmation

### III-7 Logiciel de programmation « step7 »:

Step7 est le nom du logiciel permettant de configurer et de programmer les automates programmables S7-300/400 et les systèmes d'automatisation M7, ainsi que les systèmes intégrés compact C7, STEP7 offre toutes les fonctionnalités nécessaires pour configurer,paramétrer et programmer S7-300, Les caractéristiques de STEP 7 facilitent la tâche de programmation pour l'utilisateur, Step7 est constitué d'un logiciel de base et des logiciels optionnels s'exécutant sous système d'exploitation.[7]

### III-8 La conception d'un programme avec Step7:

La conception d'une solution d'automatisation se fait par deux alternatives. soit on commence par la programmation ou par la configuration matérielle, dans notre cas on a commencé par la configuration.

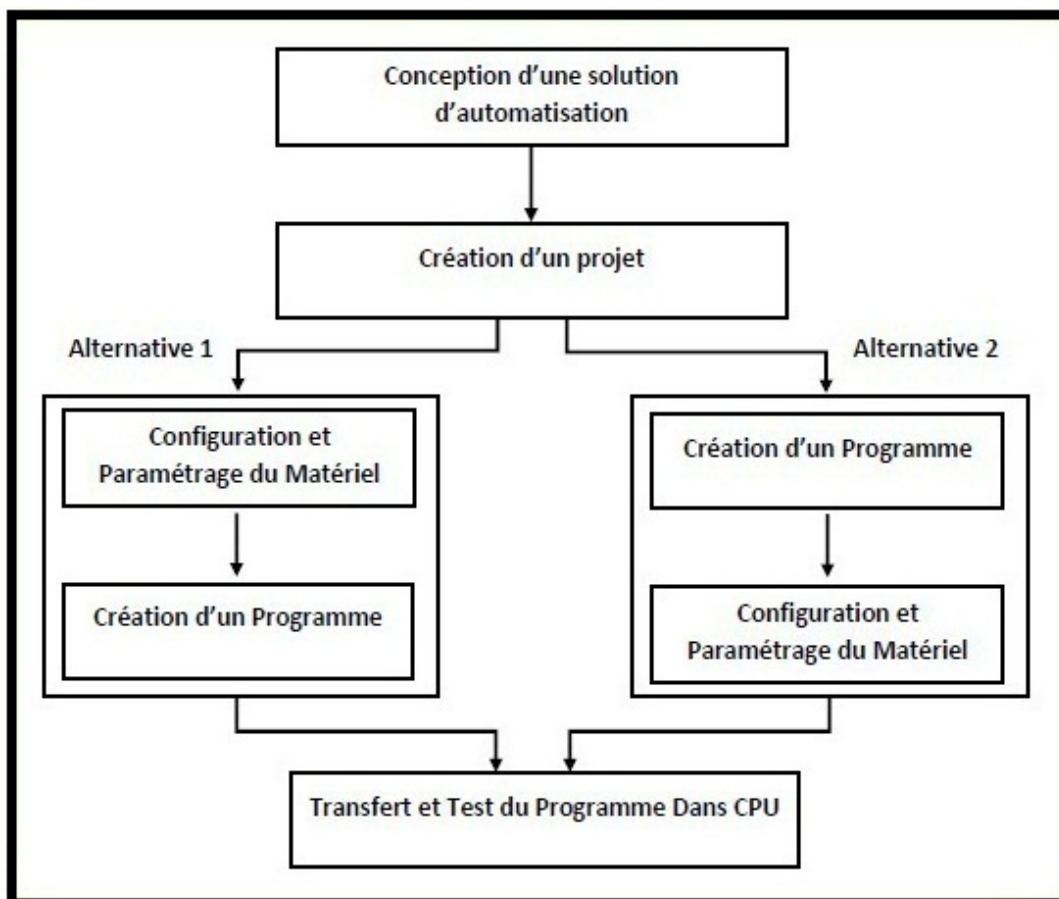


FIG ( III-7) : Organigramme pour la création de projets sous Step7

La stratégie à suivre pour faire la conception d'un programme en utilisant Step7 est :

- La création d'un nouveau projet .
- La configuration matérielle .
- La compilation et le chargement de la configuration .
- La création de la table des mnémoniques .
- L'élaboration du programme .
- La simulation avec le logiciel .

- La visualisation d'état du programme (le test) .

### III-8-1 Création du projet avec STEP7:

L'application centrale de STEP 7 est le “ SIMATIC Manager ” ,Vous l'ouvrez en double-cliquant sur son icône ( ---→ SIMATIC Manager ).

Pour créer un projet avec STEP 7, on peut lancer l'assistant de création de projet de STEP7 ou créer directement un projet que l'on configurera soi même, la deuxième méthode est un peu plus compliquée mais permet de mieux gérer le projet.

Dans la fenêtre SIMATIC Manager cliquer sur Fichier >Nouveau (ou encore CTRL+N), une fenêtre demandant un nom de projet s'ouvre,Il faut donc donner un nom au projet puis valider par OK, La fenêtre du projet s'ouvre, Le projet est vide, il faut lui insérer une station SIMATIC. cela est possible en cliquant sur le projet avec le bouton droit puis Insérer un nouvel objet >station SIMATIC 300, La station SIMATIC n'est toujours pas configurée, il faut passer à l'étape de configuration matérielle.[11]

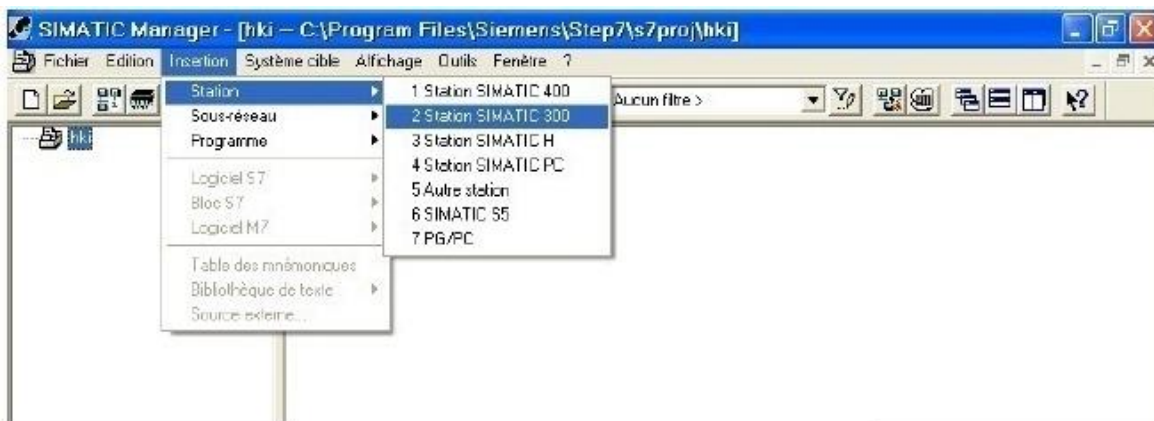


FIG ( III-8) : Création du projet avec STEP7

### III-8-2 Configuration matérielle:

La configuration matérielle est une étape très importante. elle permet de reproduire à l'identique le système utilisé (alimentation. CPU. modules etc....). Pour effectuer cette configuration il faut procéder aux étapes suivantes (FIG ( III-9)) :

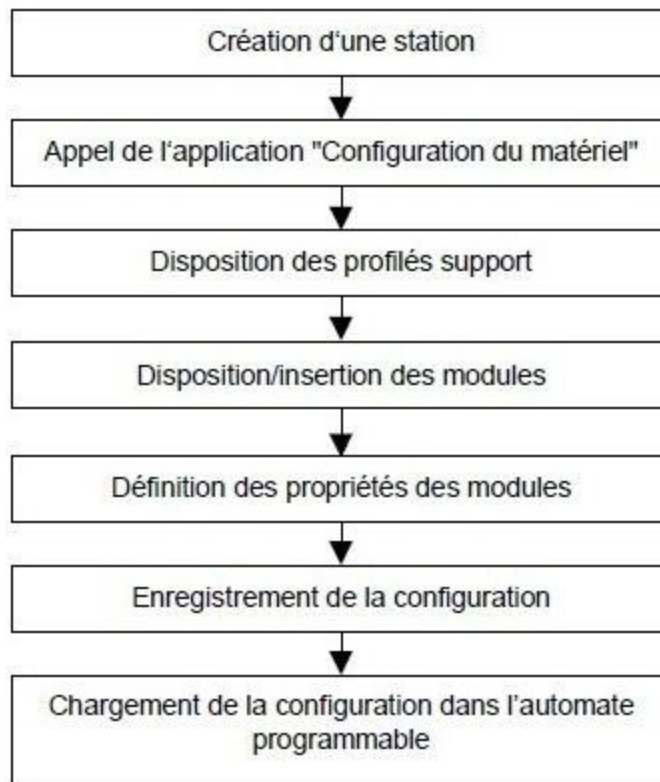


FIG ( III-9) : Les étapes de la configuration matériel.

il faut aller sur l'icône Station SIMATIC300, sur la fenêtre de droite s'affichent deux icônes « Matériel » et le nom de la CPU, ouvrir l'icône matériel : la fenêtre HW Config s'ouvre.

Nous avons tout d'abord besoin d'un châssis ou RACK puis d'une alimentation, dans le catalogue ouvrir le dossier PS-300 qui se trouve dans SIMATIC-300 puis choisir un modèle PS 307 2A, Pour l'installer sur le châssis. sélectionner l'emplacement 1 à l'aide de la souris, puis double-cliquer sur l'alimentation, on peut de la même manière insérer des composants sur le châssis en fonction de la configuration réelle.

Dans notre cas, l'insertion de la CPU S7 313 se fera dans l'emplacement 2.

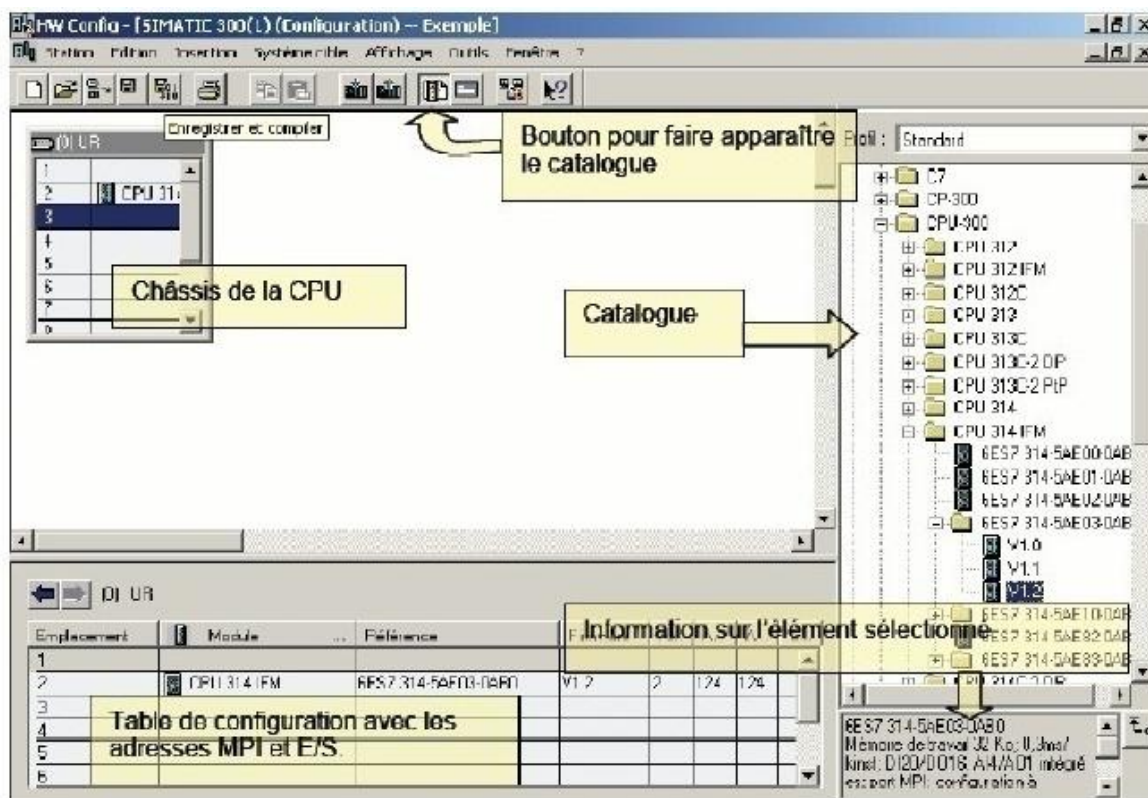


FIG ( III-10 ) : La fenêtre HW Config.

Le paramétrage de la CPU à l'aide de menu permet de définir des caractéristique, telles que : le comportement à la mise en route, la surveillance du temps de cycle ainsi que l'activation et la désactivation des plages de rémanences et les fonctions intégrées, ces données sont enregistrées dans les blocs de données systèmes.[9]

Le paramétrage des modules est réalisé automatiquement au démarrage de la CPU, ainsi le remplacement d'un module est ainsi possible sans nouveau paramétrage.

A la fin de la configuration, il suffit de cliquer sur Station>Enregistrer et compiler pour valider les changements apportés au châssis, de cette manière les changements seront pris en compte dans le reste du projet.

**Remarque:** L'emplacement 4 est le premier emplacement disponible pour des modules d'E/S, Une entrée ou une sortie est désignée dans le programme à l'aide d'une adresse qui indique clairement quel est son emplacement sur l'automate, Cette adresse est composée :

- D'une lettre indiquant la nature de la variable :
  - E pour une entrée
  - > A pour une sortie
- d'un chiffre appelé adresse d'octet qui indique l'emplacement du module (de 0 à 31)
- d'unpoint (.)
- d'un chiffre appelé adresse de bit qui indique l'emplacement de la variable sur le module (de 0 à 7).

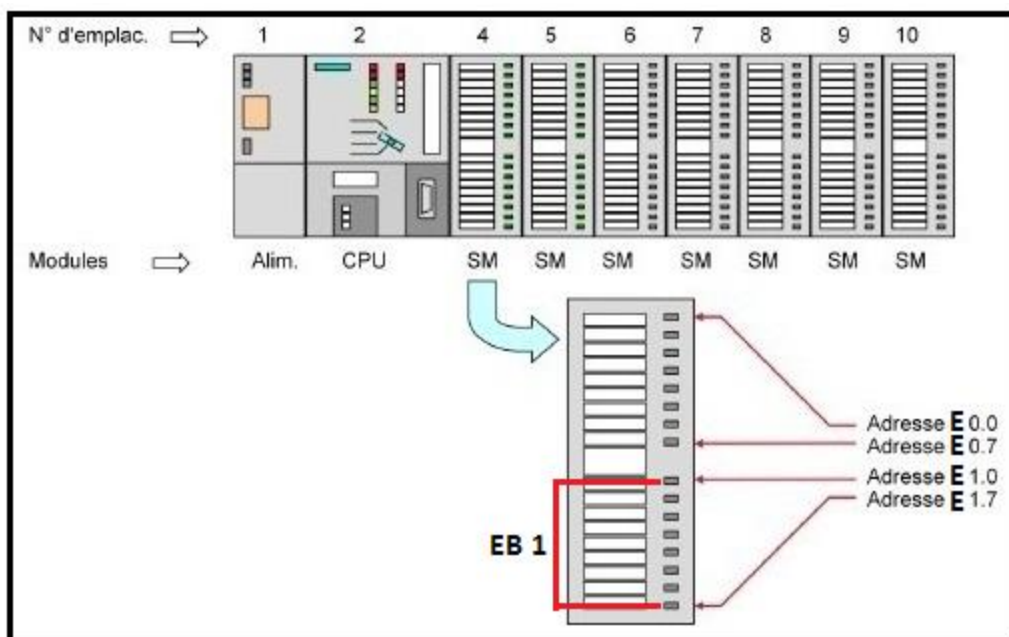


FIG ( III-11 ) : Adressage des modules S7-300 .

### III-8-3 Adressage absolu des variables:

Outre les blocs de programme, un programme utilisateur comporte aussi des données sur l'état du processus, des signaux, etc., pouvant être traitées par les opérations du programme utilisateur, les données sont stockées dans des zones de mémoire de l'automate sous forme des variables, une variable étant caractérisée de manière univoque par :

- 1- le chemin d'accès à la zone mémoire (opérande : par exemple Périphérie, MIE, MIS, mémentos, pile L, DB).
- 2- le type de données (type de données élémentaire ou complexe, type de paramètre).

**A-le chemin d'accès à la zone mémoire ( opérande ) :**

L'API S7 300 dispose de différentes zones mémoires :

- Zone E : Mémoire image des entrées.
- Zone A : Mémoire image des sorties.
- Zone M : Mémoire utilisateur.
- Zone L : Mémoire locale. associée à un module de programme.
- Zone P : Accès à la périphérie.
- Zone T : Mémoire des temporisations.
- Zone Z : Mémoire des compteurs.
- Zone DB : Mémoire utilisateur ou système structuré dans des blocs de données.

les variables E, A, M, DB, PE et PA sont rangés dans des octets (8 bits). on peut accéder à un BIT, à un OCTET, à un MOT de 16 bits ou à un DOUBLE MOT (32 bits ).

**Exemple:**

M4.3 :correspond au bit 3 de l'octet 4. L'adresse de l'octet et l'adresse du bit sont toujours séparées par un point.

MB4 :correspond aux 8 bits de l'octet 4,on peut comparer sa valeur,on peut l'additionner, le soustraire, le multiplier, le diviser, on peut lui affecter le résultat d'une opération.

MW2 : correspond au 16 bits constitué par les octets 2 (poids fort) et 3 (poids faible)

MDI : correspond au 32 bits constitué des octets 1, 2, 3 et 4.

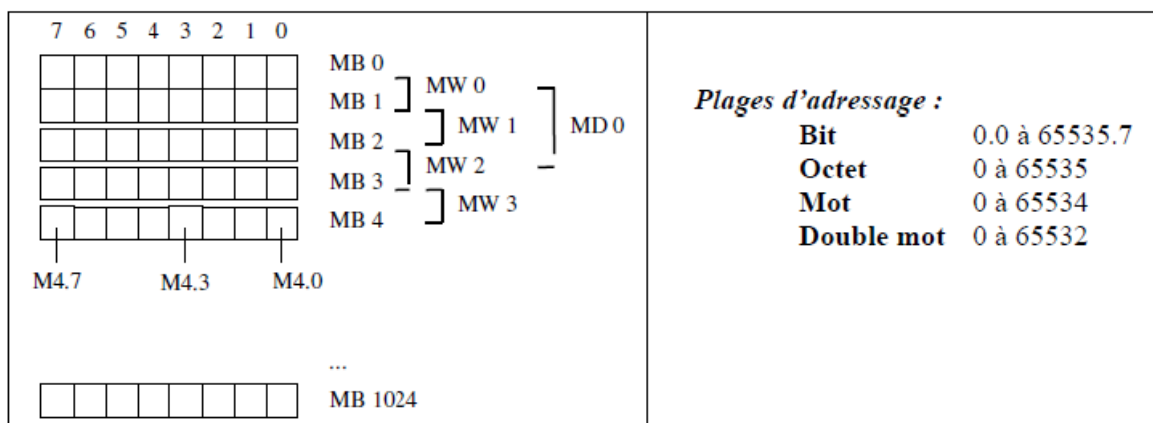


FIG ( III-12 ) : Adressage des éléments de mémoire utilisateur.

Une adresse absolue est composée d'un identificateur d'opérande et d'une adresse (par exemple À 4.0, E 1.1, M 2.0, FB21 ).

**B-Types de données utilisées:**

Les types de données simples utilisées lors de la programmation sont:

types de données	taille du type de données	Description	Plage
<b>BOOL</b>	1 bit	bit	0 ou 1
<b>BYTE</b>	8 bits	Octet, utilisé pour les caractères ASCII et les entiers non signés	0 à 255
<b>WORD</b>	16 bits	Nombre entier non signé codé sur 16 bits (mot)	0 à 65 535
<b>INT</b>	16 bits	Nombre entier signé codé sur 16 bits (mot)	- 32 768 à + 32 767
<b>DWORD</b>	32 bits	Nombre entier non signé codé sur 32 bits (double mot)	0 à $(2^{32} - 1)$
<b>DINT</b>	32 bits	Nombre entier signé codé sur 32 bits (double mot)	- $2^{31}$ à $(2^{31} - 1)$
<b>REAL</b>	32 bits	Nombre réel, codé en nombre à virgule flottante	- $10^{38}$ à $10^{38}$

TABLEAU (III -1) : Les types de données simples dans step7.

Pour visualiser les valeurs de taille octet. mot ou double mot. on peut utiliser différents formats:

- format décimal: +20047
- format hexadécimal: DW#16#4E4F
- format ASCII: 'texte entre apostrophes'
- format binaire: W#2#1010 0101 1010 0101
- format réel ou virgule flottante: +1.17549E-38

**III-8-4 Adressage mnémorique( symboliques ):**

Les mnémoniques sont des noms que l'on peut donner aux variables afin de faciliter la programmation en affectant des noms parlant plus faciles à retenir, ils améliorent la lisibilité du programme et servent également de documentation.

L'adressage mnémorique permet d'associer une adresse absolue définie à un nom mnémorique, Par exemple. on peut attribuer à l'entrée **E 0.0** le nom **END \_STOP** et au type de données **BOOL**.

On distingue les mnémoniques locaux et les mnémoniques globaux dans un programme STEP 7, Le tableau (III. ) illustre une comparaison entre ces deux type de mnémoniques.

	Mnémoniques globaux	Mnémoniques locaux
Domaine de validité	<ul style="list-style-type: none"> <li>ils sont valables dans l'ensemble du programme utilisateur,</li> <li>ils peuvent être utilisés par tous les blocs,</li> <li>leur signification est la même dans tous les blocs,</li> <li>leur nom doit être univoque dans l'ensemble du programme utilisateur.</li> </ul>	<ul style="list-style-type: none"> <li>ils sont connus uniquement dans le bloc dans lequel ils ont été définis,</li> <li>vous pouvez utiliser le même nom dans différents blocs à des fins différentes.</li> </ul>
Caractères autorisés	<ul style="list-style-type: none"> <li>lettres, chiffres, caractères spéciaux,</li> <li>trémas à l'exclusion de 0x00, 0xFF et des guillemets,</li> <li>lorsque vous utilisez des caractères spéciaux dans un mnémotique, ce dernier doit être placé entre guillemets.</li> </ul>	<ul style="list-style-type: none"> <li>lettres,</li> <li>chiffres,</li> <li>caractère de soulignement ( _ ),</li> </ul>
Utilisation	<p>Vous pouvez définir des mnémoniques globales pour :</p> <ul style="list-style-type: none"> <li>entrées/sorties (E, EB, EW, ED, A, AB, AW, AD)</li> <li>entrées, sorties de périphérie (PE, PA)</li> <li>mémentos (M, MB, MW, MD)</li> <li>temporisations (T)/ compteurs (Z)</li> <li>blocs de code (OB, FB, FC, SFB, SFC)</li> <li>blocs de données (DB)</li> <li>types de données utilisateur</li> <li>table des variables (VAT)</li> </ul>	<p>Vous pouvez définir des mnémoniques locales pour :</p> <ul style="list-style-type: none"> <li>paramètres de blocs (paramètres d'entrée, de sortie, d'entrée/sortie),</li> <li>données statiques d'un bloc</li> <li>données temporaires d'un bloc</li> </ul>
Endroit de définition	table des mnémoniques	table de déclaration des variables du bloc

TABLEAU (III -2) : Comparaison entre les mnémoniques globales et locales [8].

**III-8-5 Table des mnémoniques globale:**

Pour améliorer la lisibilité et la clarté de notre programme. nous avons utilisé des mnémoniques à la place des adresses absolues, pour cela nous avons créé une table de mnémoniques dans laquelle nous avons défini pour chaque opérande utilisée un nom d'adresse absolue, le type de données ainsi qu'un commentaire.

Dans Le STEP 7 Une table des mnémoniques (objet "Mnémoniques") vide est automatiquement générée lorsque vous créez un programme S7.

Le tableau suivant (FIG ( III-13)) représente la table des mnémoniques qu'on a utilisées dans notre programme :

Éditeur de mnémoniques - [Programme S7(1) (Mnémoniques) -- OPA 650\SIMATIC 300(1)\CPU 313]

Table Edition Insertion Affichage Outils Fenêtre ?

Tous les mnémoniques

	Mnémonique	Opérand	Type de	Commentaire
1	ROTATION	A 12.0	BOOL	CNT rotaion la table
2	CNT vibr fixe tremie	A 12.1	BOOL	CNT vibrateur fixe de la tremie
3	INT vibreur moules	A 12.2	BOOL	INT vibreur moules
4	CNT FREIN vibra moule	A 12.3	BOOL	CNT FREINAGE vibration moules
5	CNT chariot B .M	A 12.4	BOOL	CNT chariot B .M
6	CNT tapais doseur	A 12.5	BOOL	CNT tapais doseur
7	CNT vibreur mob tremie	A 12.6	BOOL	CNT vibreur mobile tremie
8	CNT inverter tremie	A 12.7	BOOL	CNT inverter movmment tremie
9	CNT frein mot rot table	A 13.0	BOOL	CNT freinage moteur rotation table
10	INT chariot position ar	A 13.1	BOOL	INT chariot position en arriere
11	INT demoleur pos bas	A 13.2	BOOL	INT demoleur position en bas
12	LAMP rotation auto table	A 13.3	BOOL	LAMP signalisation rotation auto table
13	LAMP sig vibration moule	A 13.4	BOOL	LAMP signalisation vibration moules
14	signal acoustic rotation	A 13.5	BOOL	signal acoustique rotation la table /temp huile /press air
15	EV premiere pressage	A 13.6	BOOL	EV premiere pressage
16	EV pressage	A 13.7	BOOL	EV pressage
17	lampe automatique	A 14.0	BOOL	lampe automatique
18	EV souffleur doseur	A 14.1	BOOL	EV souffleur doseur
19	EV desente vibreur polon	A 14.2	BOOL	EV desente vibreur polongent
20	EV oscillation vib polon	A 14.3	BOOL	EV oscillation vibreur polongent
21	EV movv avant doseur	A 14.4	BOOL	EV mouvement avant doseur
22	EV movv arriere doseur	A 14.5	BOOL	EV mouvement arriere doseur
23	EV ferme croche ejector	A 14.6	BOOL	EV fermee crochets demoleur
24	EV souffle pelle auto	A 14.7	BOOL	EV soufflerie pelle automatique
25	EV elevation pelle auto	A 15.0	BOOL	EV elevation pelle automatique
26	EV abaissement pelle	A 15.1	BOOL	EV abaissement pelle automatique
27	EV souffle nett.fonds	A 15.2	BOOL	EV souffle nett.fonds moule station demoleur
28	INT stop pump hydr extra	A 15.3	BOOL	INT stop hydraulique pompe par extrapressage
29	EV centrage table	A 15.4	BOOL	EV centrage table
30	EV freinage table	A 15.5	BOOL	EV freinage table
31	EV desente demoleur	A 15.6	BOOL	EV desente demoleur
32	EV montee demoleur	A 15.7	BOOL	EV montee demoleur
33	EV descente plaque tenue	A 16.0	BOOL	EV descente plaque de tenue
34	EV montee plaque tenue	A 16.1	BOOL	EV montee plaque de tenue
35	INT stop pompe doseur	A 16.2	BOOL	INT stop pompe malaxeur doseur
36	INT tremie deceleration	A 16.3	BOOL	INT inverter tremie deceleration
37	CNT vis sans fin	A 16.4	BOOL	CNT vis sans fin
38	PS marche rotation table	E 0.0	BOOL	PS marche rotation table
39	PS arret rotation table	E 0.1	BOOL	PS arret rotation table
40	SL rotatio table/chariot	E 0.2	BOOL	SL rotation table avec chariot
41	SL rotation sans chariot	E 0.3	BOOL	SL rotation table sans chariot
42	SL travail / entretien	E 0.4	BOOL	SL travail / entretien
43	SL tremie 1 course	E 0.5	BOOL	SL tremie 1 course
44	SL tremie 2 course	E 0.6	BOOL	SL tremie 2 course
45	PS impulsion man tremie	E 0.7	BOOL	PS impulsion manulle tremie
46	PS marche vibreur moules	E 1.0	BOOL	PS marche vibreur moules
47	PS arret vibreur moules	E 1.1	BOOL	PS arret vibreur moules
48	relais ther vibr moule	E 1.2	BOOL	relais thermique vibreur moules
49	SL premiere pressage	E 1.3	BOOL	SL premiere pressage
50	sonde level min 2 couch	E 1.4	BOOL	sonde de niveau mininom 2eme couche tremie
51	sonde de niveau doseur	E 1.5	BOOL	sonde de niveau doseur
52	CNT pompe hydrolique	E 1.7	BOOL	CNT pompe hydrolique
53	SL branche crochets	E 2.0	BOOL	SL branche crochets demoleur
54	SL commande manuelle	E 2.1	BOOL	SL commande manuelle
55	PS chariot course en AV	E 2.2	BOOL	PS chariot course en avant

56	PS chariot course en AR	E	2.3	BOOL	PS chariot course en arriere
57	PS selevement pelle	E	2.4	BOOL	PS selevement pelle automatique
58	ps abaissement pelle	E	2.5	BOOL	ps abaissement pelle automatique
59	PS desente demouleur	E	2.6	BOOL	PS desente demouleur
60	PS montee demouleur	E	2.7	BOOL	PS montee demouleur
61	PS repetition pressage	E	3.0	BOOL	PS repetition pressage
62	INT consentement linear	E	3.1	BOOL	INT consentement linear
63	SL cycle automatique	E	3.2	BOOL	SL cycle automatique
64	PS START cycle auto	E	3.3	BOOL	PS demarrage cycle automatique
65	CNT malaxeur doseur	E	3.4	BOOL	CNT malaxeur doseur
66	CNT PUMP hydrolic doseur	E	3.5	BOOL	CNT pompe hydrolique doseur
67	PS desente vibreur polEN	E	3.6	BOOL	PS desente vibreur polengeant
68	PS montee vibreur polen	E	3.7	BOOL	PS montee vibreur polengeant
69	SL vibreur plongeant	E	4.2	BOOL	SL vibreur plongeant
70	SL dosage pos 1	E	4.3	BOOL	SL dosage pos 1
71	SL dosage pos 2	E	4.4	BOOL	SL dosage pos 2
72	PS manuelle doseur	E	4.5	BOOL	PS commande manuelle doseur
73	FC doseur dech pos2/pos1	E	4.6	BOOL	FC doseur position decharge pos 2/pos 1
74	FC doseur dech pos1/pos2	E	4.7	BOOL	FC doseur position decharge pos 1 / pos 2
75	FC securite tremie	E	5.0	BOOL	FC securite tremie
76	SL vibreur fixe tremie	E	5.1	BOOL	SL vibreur fixe tremie
77	SL rotation table/tremie	E	5.2	BOOL	SL rotation table avec tremie
78	SL rotation sans tremie	E	5.3	BOOL	SL rotation table sans tremie
79	FC vibreur mobile tremie	E	5.4	BOOL	FC vibreur mobile tremie
80	FC chariot en avant	E	5.5	BOOL	FC chariot en avant
81	FC securite chariot	E	5.6	BOOL	FC securite chariot
82	FC ralentissement tremie	E	5.7	BOOL	FC ralentissement tremie
83	FC securite 1 pressage	E	6.0	BOOL	FC securite 1 pressage
84	FC extra 1er pressage	E	6.1	BOOL	FC extra pressage 1er pressage
85	FC securite centrage	E	6.2	BOOL	FC securite centrage
86	FC centrage branche	E	6.3	BOOL	FC centrage branche
87	FC freinage vibr moule	E	6.4	BOOL	FC freinage vibreur moule
88	FC securite protection	E	6.5	BOOL	FC securite protection
89	pressostate montee moule	E	6.6	BOOL	pressostate air montee moule station nettoyage
90	pressostate monte moule	E	6.7	BOOL	pressostate air montee moule station pressage
91	FC freinage table	E	7.0	BOOL	FC freinage table
92	FC centrage table	E	7.1	BOOL	FC centrage table
93	FC inpultion tremie	E	7.2	BOOL	FC inpultion tremie
94	FC arret reducteur	E	7.3	BOOL	FC arret reducteur en position
95	FC moule bas 2eme pres	E	7.4	BOOL	FC moule bas station 2 eme pressage
96	FC moule bas doseur	E	7.5	BOOL	FC moule bas station doseur
97	FC moule bas nettoyage	E	7.6	BOOL	FC moule bas station nettoyage
98	FC securite plaque dosor	E	7.7	BOOL	FC securite nettoyage plaque tenue doseur
99	FC securite demouleur	E	8.0	BOOL	FC securite demouleur
100	FC demouleur en bas	E	8.1	BOOL	FC demouleur en bas
101	FC securite vibreur pol	E	8.2	BOOL	FC securite vibreur polongeat
102	FC moule haut demouleur	E	8.4	BOOL	FC moule haut station demouleur
103	FC1 crochet OV demouleur	E	8.5	BOOL	FC 1 crochet ouvert demouleur
104	FC2 crochet OV demouleur	E	8.6	BOOL	FC 2 crochet ouvert demouleur
105	FC1 crochet FR demouleur	E	8.7	BOOL	FC 1 crochet ferme demouleur
106	FC2 crochet FR demouleur	E	9.0	BOOL	FC 2 crochet ferme demouleur
107	FC securite 2 eme press	E	9.1	BOOL	FC securite 2 eme pressage
108	FC extra 2eme presSAGE	E	9.2	BOOL	FC extra pressage 2eme presage
109	sonde ther temp huile	E	9.3	BOOL	sonde thermique temp huile centrale hydrolique
110	pressostate centrale	E	9.4	BOOL	pressostate centrale hydrolique
111	pressostate frein table	E	9.5	BOOL	pressostate air freinage table
112	SL clef dech man doseur	E	9.6	BOOL	SL a clef decharge manuel doseur
113	SL clef CHARG man doseur	E	9.7	BOOL	SL a clef charge manuel doseur
114	SL clef desente doseur	E	10.0	BOOL	SL a clef desente plaque tenue doseur
115	SL clef MONTE doseur	E	10.1	BOOL	SL a clef montee plaque tenue doseur
116	FC 1 chute carreaux	E	10.2	BOOL	photocell 1 controle chute carrellage

117	FC 2 chute carreaux	E	10.3	BOOL	photocell 2 controle chute carrellage
118	FC1 station identifier	E	11.1	BOOL	LS proximity 1 station identification
119	FC2 station identifier	E	11.2	BOOL	LS proximity 2 station identification
120	FC3 station identifier	E	11.3	BOOL	LS proximity 3 station identification
121	FC4 station identifier	E	11.4	BOOL	LS proximity 4 station identification
122	vib/pres/ table rot	FC	1	FC 1	
123	chariot	FC	2	FC 2	
124	compteur	FC	3	FC 3	
125	doseur	FC	4	FC 4	
126	nettoyage	FC	5	FC 5	
127	demoleur hydraulique	FC	6	FC 6	
128	tremie	FC	7	FC 7	
129	nettoyage station demoul	FC	8	FC 8	
130	vibreur plongeat	FC	9	FC 9	
131	pas disp	FC	10	FC 10	
132	start stop autom cycl	FC	11	FC 11	
133	nettoyage table	FC	12	FC 12	
134	conversion ASI	FC	13	FC 13	
135	alarmes	FC	14	FC 14	
136	identification	FC	16	FC 16	
137	nettoyage moule demoul	FC	25	FC 25	
138	guide opirateur	FC	26	FC 26	
139	time out	FC	27	FC 27	
140	fonction complete	FC	28	FC 28	
141	position 1	M	0.1	BOOL	
142	position 2	M	0.2	BOOL	
143	position3	M	0.3	BOOL	
144	position4	M	0.4	BOOL	
145	position 5	M	0.5	BOOL	
146	position6	M	0.6	BOOL	
147	position7	M	0.7	BOOL	
148	doseur commande	M	10.1	BOOL	memory doseur commande
149	vibreur plon commande	M	10.2	BOOL	memory vibreur plongeant commande
150	station nettoy commande	M	10.3	BOOL	memory station nettoyage commande
151	tremie commande	M	10.4	BOOL	memory tremie commande
152	1st press commande	M	10.5	BOOL	memory premiere pressage commande
153	pressage commande	M	10.6	BOOL	memory pressage commande
154	chariot /demou commande	M	10.7	BOOL	memory chariot /demoleur commande
155	OK STN.1	M	20.1	BOOL	
156	OK STN .2	M	20.2	BOOL	
157	OK STN .3	M	20.3	BOOL	
158	OK STN .4	M	20.4	BOOL	
159	OK STN .5	M	20.5	BOOL	
160	OK STN .6	M	20.6	BOOL	
161	OK STN .7	M	20.7	BOOL	
162	depart automatique	M	31.0	BOOL	mem depart automatique
163	arret automatique	M	41.0	BOOL	mem arret automatique
164	M64.0	M	64.0	BOOL	
165	tremie 1course	M	64.1	BOOL	mem tremie 1course
166	tremie 2 course	M	64.2	BOOL	mem tremie 2 course
167	elevation pelle auto	M	65.0	BOOL	mem aux elevation pelle automatique
168	commande man vibr plon	M	65.1	BOOL	mem commande manul vibreur plongeant
169	rotation auto table	M	65.2	BOOL	mem rotation automatique table
170	5 rotation table	M	65.3	BOOL	mem aux 5 rotation table
171	anterpritation tremie	M	65.4	BOOL	mem anterpritation tremie
172	1fin de course securite	M	65.5	BOOL	mem 1fin de course securite
173	2 fin de course securite	M	65.6	BOOL	mem 2 fin de course securite

174	aux 4rotation table	M	65.7	BOOL	mem aux 4rotation table
175	aux 1 descente demouleur	M	66.3	BOOL	mem aux 1 descente demouleur
176	1 alarme press frein tab	M	66.4	BOOL	mem 1 alarme pression air frein table
177	3 alarme press frein tab	M	66.5	BOOL	mem 3 alarme pression air frein table
178	pelle autom en avant	M	66.6	BOOL	mem aux pelle autom en avant
179	pelle autom en arriere	M	66.7	BOOL	mem aux pelle autom en arriere
180	cycl de prolevment	M	67.0	BOOL	cycl de prolevment
181	mém mouvement chariot	M	67.1	BOOL	mém mouvement chariot
182	mém 1rotation table	M	67.2	BOOL	mém 1rotation table
183	mém monte démouleur	M	67.3	BOOL	mém monte démouleur
184	mém desent vibreur polon	M	67.4	BOOL	mém desent vibreur polongeant
185	mém alarme careau	M	67.6	BOOL	mém alarme careau
186	mém level min trémie	M	68.2	BOOL	mém level min trémie
187	moule no bas dans doseur	M	68.3	BOOL	mém moule non en bas station doseur
188	arret rotation et vibr	M	68.4	BOOL	mém arret rotation et vibreur moue provoque par arret produ...
189	alarm batterie CPU	M	68.5	BOOL	mém alarm batterie CPU
190	2 rotation table	M	68.6	BOOL	mém aux 2 rotation table
191	rotation table	M	68.7	BOOL	mém aux 3 rotation table
192	3 rotation table	M	69.0	BOOL	mém arret rotation table provoque par tapais vide
193	comande doseur	M	69.1	BOOL	mém comande doseur
194	cuve doseur pleine	M	69.2	BOOL	mém cuve doseur pleine
195	comande doseur2	M	69.6	BOOL	mém aux comande doseur
196	mém cycle auto	M	70.0	BOOL	mém cycle auto
197	comande urgence	M	70.1	BOOL	mém comande urgence
198	alarme rotat/temp huile	M	70.2	BOOL	mém alarme rotation /temp huile/pression air
199	comande nettoyage moule	M	70.3	BOOL	mém comande nettoyage moule démouleur
200	1oscillation vibr polo	M	70.6	BOOL	mém aux 1oscillation vibreur polongeant
201	2oscillation vibr polo	M	70.7	BOOL	mém aux 2oscillation vibreur polongeant
202	vibration moules	M	74.0	BOOL	mém vibration moules et lampe
203	antérépétition pressage	M	74.3	BOOL	mém antérépétition pressage
204	2ala pression frein tab	M	74.4	BOOL	mém aux 2alarme pression air frein table
205	antérépétition déméleur	M	74.5	BOOL	mém antérépétition déméleur
206	antérépétition chariot	M	74.6	BOOL	mém antérépétition chariot
207	préssage bref	M	74.7	BOOL	mém préssage bref
208	M75.0	M	75.0	BOOL	
209	nettoy fonds de moule	M	75.1	BOOL	mém aux nettoyage fonds de moule stasjon démouleur
210	mouvment trémie	M	75.2	BOOL	mém aux mouvment trémie
211	ferme crochets démou	M	75.3	BOOL	mém aux fermeture crochets démouleur
212	M75.4	M	75.4	BOOL	
213	premier préssage	M	75.5	BOOL	mém aux premier préssage
214	désente démouleur	M	75.6	BOOL	mém aux désente démouleur
215	préssage	M	75.7	BOOL	mém aux préssage
216	fonctionement doseur	M	76.6	BOOL	mém aux fonctionnement doseur
217	alarme level min trémie	M	200.0	BOOL	mém alarme niveau min trémie
218	alarme arret production	M	200.1	BOOL	mém alarme arret production
219	moule non en bas doseur	M	200.2	BOOL	mém moule non en bas station doseur
220	alarme battrie CPU	M	200.3	BOOL	mém alarme battrie CPU
221	alarme préss frein table	M	200.4	BOOL	mém alarme préssion air frein table
222	M200.5	M	200.5	BOOL	
223	alarme dispo sécurité	M	201.0	BOOL	mém alarme dispositive sécurité
224	alarme pressostat huile	M	201.1	BOOL	mém alarme pressostat huile
225	alarme temp huile	M	201.2	BOOL	mém alarme temp huile
226	alarme tapais vide	M	201.3	BOOL	mém alarme arret rotation provoque par tapais vide
227	alarme pressage suppl 1	M	201.4	BOOL	mém alarme pressage suppl 1
228	alarme pressage suppl 2	M	201.5	BOOL	mém alarme pressage suppl 2
229	alarme chute careau	M	201.6	BOOL	mém alarme chute careau
230	alarme thermique vibreur	M	201.7	BOOL	mém alarme thermique vibreur moule
231	organisation	OB	1	OB 1	
232	PS FLT	OB	81	OB 81	

233	MOD ERR	OB	122	OB	122	
234	course simple tremie	T	0	TIMER		course simple tremie
235	course double tremie	T	1	TIMER		course double tremie
236	pressage	T	2	TIMER		pressage
237	decharge/soufflet doseur	T	3	TIMER		dechargement/soufflet doseur
238	arret rota tapis vide	T	4	TIMER		arret rotation de table provoque par tapis vide
239	arret tapisdoseur pleine	T	5	TIMER		arret tapis provoque par cuve doseur pleine
240	oscillation vibreur plon	T	6	TIMER		oscillation vibreur plongeant
240	oscillation vibreur plon	T	6	TIMER		oscillation vibreur plongeant
241	alarme rotation table	T	7	TIMER		duree alarme rotation table
242	soufflet nettoyage pelle	T	8	TIMER		soufflet nettoyage pelle autom
243	ouverture crochets	T	9	TIMER		ouverture crochets demouleur
244	vibreux fixe tremie	T	10	TIMER		vibreux fixe tremie
245	soufflet nettoyage table	T	11	TIMER		soufflet nettoyage table
246	montee pelle auto	T	12	TIMER		montee pelle automatique
247	soufflet nettoyage stat	T	13	TIMER		soufflet nettoyage stat demouleur
248	arret rotat min tremie	T	14	TIMER		arret rotation provoque par niveau min tremie
249	arret production	T	15	TIMER		arret rotation vibr provoque par arret production
250	depart auto	T	18	TIMER		depart automatique
251	demarrage auto	T	22	TIMER		demarrage autom instalation
252	arret auto	T	23	TIMER		arret autom instalation
253	remise a zero compteur	T	28	TIMER		remise a zero compteur
254	controle chut de carr	T	29	TIMER		controle chut de carrelage
255	alarme press frein table	T	31	TIMER		alarme pression air frein table
256	comptage unite pressage	Z	1	COUNTER		comptage unite de pressage
257	comptage centaines press	Z	2	COUNTER		comptage centaines de pressage
258	comptage milliers press	Z	3	COUNTER		comptage milliers de pressage
259						

FIG ( III-13 ) : La table des mnémoniques.

### **III-8-6 L'élaboration du programme:**

Deux programmes différents s'exécutent dans une CPU d'API S7 300: le système d'exploitation et le programme utilisateur.

#### **III-8-6-1 Système d'exploitation:**

Le système d'exploitation, contenu dans chaque CPU, organise toutes les fonctions et procédures dans la CPU qui ne sont pas liées à une tâche d'automatisation spécifique. Ses tâches sont les suivantes :

- le déroulement du démarrage et du redémarrage,
- l'actualisation de la mémoire image des entrées et l'émission de la mémoire image des sorties,
- l'appel du programme utilisateur.
- l'enregistrement des alarmes et l'appel des OB d'alarme.
- la détection et le traitement d'erreurs.
- la gestion des zones de mémoire.
- la communication avec des consoles de programmation et d'autres partenaires de communication.

#### **III-8-6-2 Programme utilisateur:**

Vous devez créer votre programme utilisateur et le charger dans la CPU, il contient toutes les fonctions nécessaires au traitement de votre tâche d'automatisation spécifique. Il doit entre autres :

- déterminer les conditions pour le démarrage et le redémarrage de la CPU (par exemple, initialiser des signaux),
- traiter des données du processus (par exemple, combiner des signaux binaires, lire et exploiter des valeurs analogiques, définir des signaux binaires pour la sortie, écrire des valeurs analogiques),
- réagir aux alarmes.
- traiter les perturbations dans le déroulement normal du programme.

**III-8-6-3 Organisation du programme utilisateur:**

Le logiciel de programmation STEP 7 vous permet de structurer votre programme utilisateur, c'est-à-dire de le subdiviser en différentes parties autonomes, Il en résulte les avantages suivants :

- Ecrire des programmes importants mais clairs.
- Standardiser certaines parties du programme.
- Simplifier l'organisation du programme.
- Modifier facilement le programme.
- Simplifier le test du programme, car vous pouvez l'exécuter section par section.
- Faciliter la mise en service.

L'organisation des applications réalisées avec Step7 sont conçus à partir des blocs, On distingue :

- les blocs de code (OB. FB. SFB. FC. SFC) qui contiennent les programmes qu'on doit charger dans la CPU
- les blocs de données (DB d'instance et DB globaux) qui contiennent les paramètres du programme.

On peut citer les blocs importants suivants :

**A-Blocs d'organisations (OB):**

Les blocs d'organisation (OB) constituent l'interface entre le système d'exploitation et le programme utilisateur, ils sont appelés par le système d'exploitation et gèrent le traitement de programme cyclique et déclenché par alarme, ainsi que le comportement à la mise en route de l'automate programmable et le traitement des erreurs, vous pouvez programmer les blocs d'organisation et déterminer ainsi le comportement de la CPU.

Le bloc d'organisation OB1 est généré automatiquement lors de la création d'un projet. il représente le programme principal (MAIN dans le langage C), En effet c'est le programme appelé cycliquement par le système d'exploitation, Les autres blocs existant dans le projet seront exécutés à leur appel par l'OB1.[7]

**B-Blocs fonctionnels (FB):**

Un bloc fonctionnel contient un programme qui exécuté, lorsque ce bloc fonctionnel est appelé par un autre bloc de code, il facilite la programmation de fonctions complexes, comme la commande de moteur (accélération).

Un bloc de données d'instance est associé à chaque FB qui constitue la mémoire. Les paramètres transmis au FB ainsi que les variables statiques sont sauvegardés dans le bloc de données locales.

**C-Fonctions (FC):**

Les fonctions font partie des blocs que vous programmez vous-même, Une fonction est un bloc de code sans mémoire, les variables temporaires d'une fonction sont sauvegardées dans la pile des données locales, ces données sont perdues à l'achèvement de la fonction.

Les fonctions peuvent faire appel à des blocs de données globaux pour la sauvegarde des données.

**D-Blocs de données (DB):**

Les DB sont utilisés pour la mise à la disposition de l'espace mémoire pour les variables de type données. on a deux types de bloc :

- Bloc de données d'instance : il est associé à chaque appel de bloc fonctionnel transmettant des paramètres. il contient des paramètres effectifs et des données statique du FB, on appelle l'instance l'appel d'un FB pour un DB.
- Bloc de données globales (DB): contrairement aux blocs de code (OB, FC, FB) les DB ne contiennent pas d'instruction STEP7, ils servent à l'enregistrement de données utilisateur, ils contiennent des données variables que le programme utilisateur utilise, Tout FB, FC, OB peut lire les données contenues dans un DB global ou écrire des données dans un DB global, ces données sont conservées dans le bloc de données même lorsqu'on quitte le DB.

Le rôle des DB dans un programme utilisateur est illustré en figure suivante :

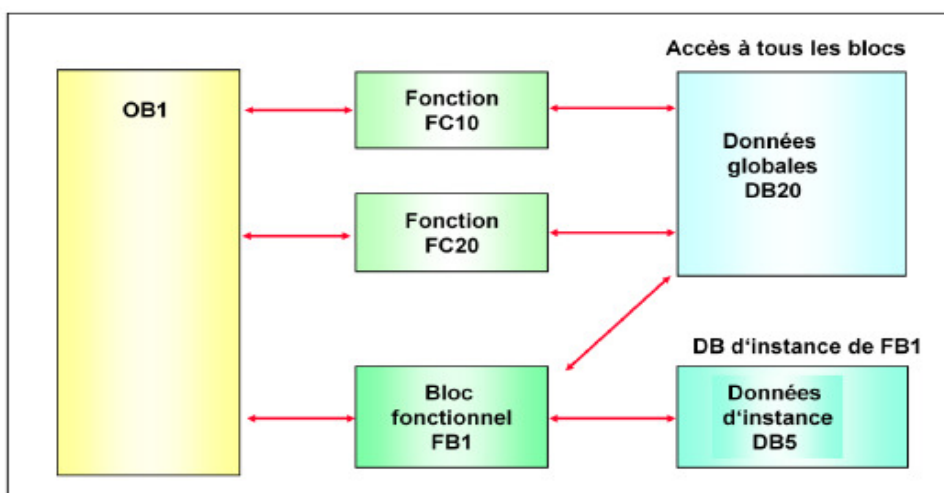


FIG (III-14) : blocs de programme dans step7

III-8-6-4 Création de blocs de code et blocs de données :

Les blocs de code (OB, FB, FC) comportent une section de déclaration de variables, une section d'instructions et possèdent également des propriétés.

Lors de la programmation, vous devez par conséquent éditer les trois parties suivantes :

-**table de déclaration des variables** : vous y définissez les paramètres, attributs système des paramètres ainsi que les variables locales du bloc.

-**section des instructions** : vous y programmez le code du bloc que l'automate programmable doit exécuter. La section des instructions comporte un ou plusieurs réseaux. Pour créer les réseaux, vous disposez par exemple des langages de programmation LIST (liste d'instructions), CONT (schémas à contacts) et LOG (logigramme).

-**propriétés de bloc** : elles contiennent des informations supplémentaires comme l'horodatage ou l'indication du chemin qui sont entrées par le système. Vous pouvez également entrer vous-même des données sur le nom, la famille, la version et l'auteur ou bien affecter des attributs système aux blocs.

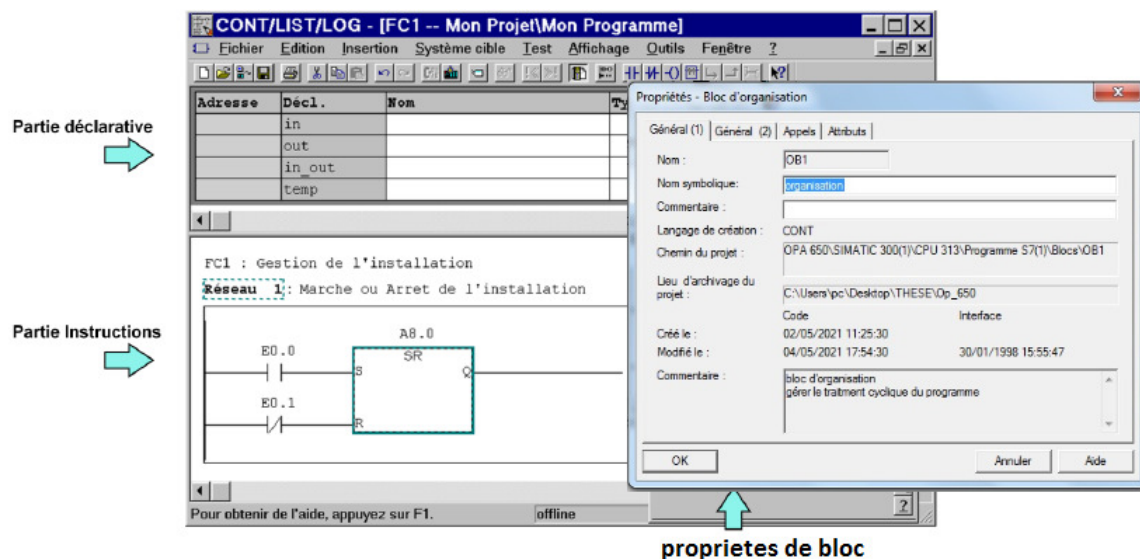


FIG (III-15) :Les parties de bloc de code.

pour créer un bloc de code ( par exemple FC1) Sélectionnez le dossier 'Blocs' pour insérer un nouveau bloc FC1 dans le projet, le bloc S7 **Fonction** est alors inséré dans le dossier Blocs.

Il est maintenant possible d'entrer un nom pour la fonction et de sélectionner diverses options pour la documentation des blocs.

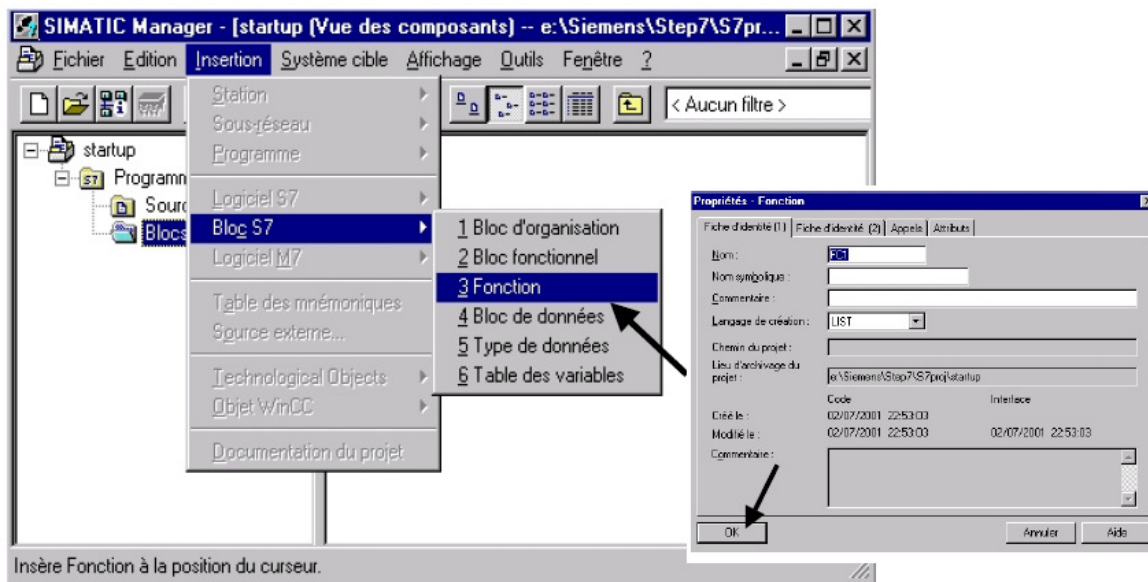


FIG (III-16) : création un bloc de code

Dans un bloc de données, vous stockez, par exemple, les valeurs auxquelles votre machine ou installation accède, contrairement au bloc de code le bloc de données ne comporte que la section de déclaration des variables, il ne possède pas de section d'instructions, et ne nécessite donc pas de programmation de réseaux.

pour créer un bloc de données, partir de l'éditeur CONT/LIST/LOG permet d'ouvrir un bloc de données existant ou d'en créer un nouveau, si vous cliquez sur le bouton "Nouveau", la boîte de dialogue "Nouveau" apparaît à l'écran, Sélectionnez le projet et le programme utilisateur puis tapez le "Nom de l'objet", par exemple DB2 (le champ Type d'objet devrait indiquer "Bloc de données" ou "Editables"), après confirmation avec le bouton "OK", la boîte de dialogue " Nouveau bloc de données" apparaît à l'écran.

Cette boîte de dialogue permet de définir le type de bloc de données à créer :

- bloc de données (bloc de données global).
- bloc de données associé à un type de données utilisateur (créé un DB selon un bloc UDT)
- bloc de données associé à un bloc fonctionnel (créé un DB d'instance pour un FB).

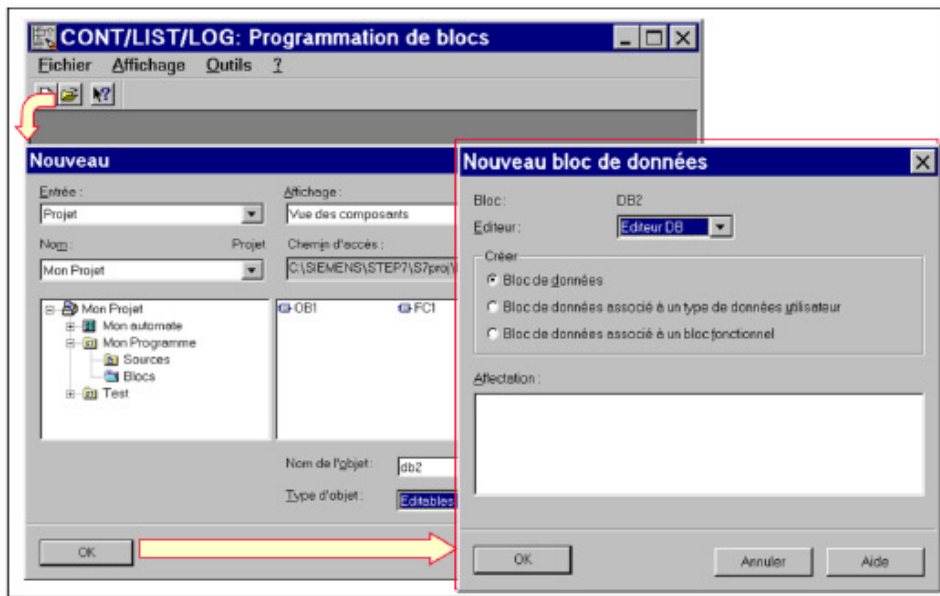


FIG (III-17) :Création un bloc de données

Nous avons structuré notre programme sous forme de structure arborescente comme illustre sur la figure( III- 18 ) suivante :

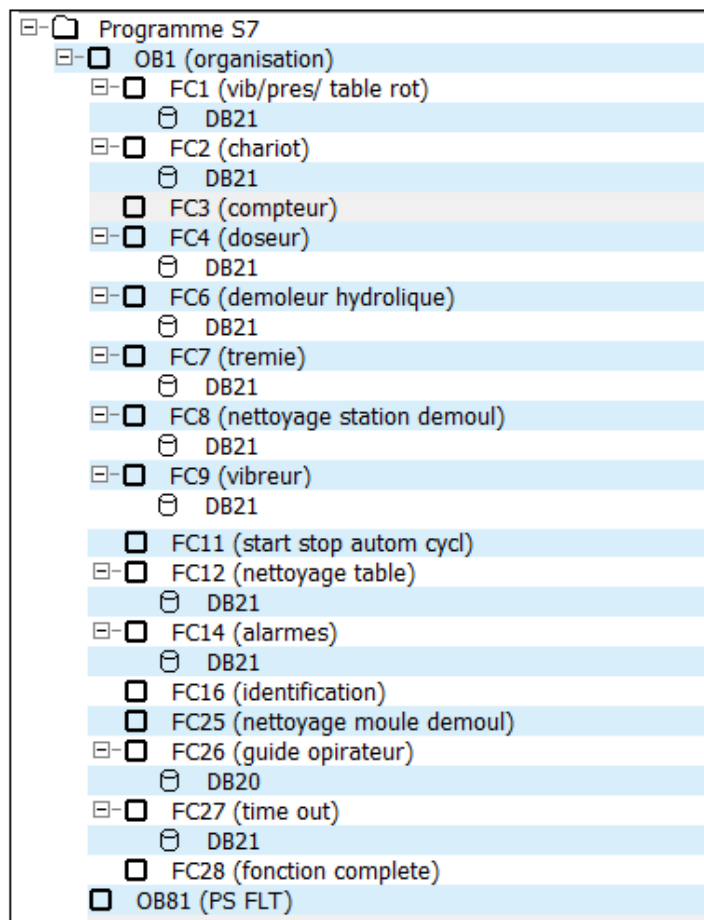


FIG (III-18) : structure arborescente de programme step7.

- Dans notre programme, on a utilisé les blocs (OB1) ,(OB81), 16 (FC) et 2 DB
- Un bloc d'organisation (OB1) pour gérer le traitement cyclique du programme.
- Un bloc d'organisation (OB81) pour gérer l'erreur d'alimentation.
- FC1,FC2 , jusqu'à FC28 sont les fonctions dont nous avons introduit nos programmes fonctionnels du système.
- DB20 et DB21 sont les blocs de données.

**III-8-6-5 Traitement cyclique du programme:**

À la mise sous tension ou à la mise en marche (RUN), la CPU procède à une initialisation complète (démarrage avec l'OB100), lors du démarrage, le système d'exploitation efface les mémentos, les temporisations et les compteurs non rémanents, les alarmes de processus et de diagnostic mémorisées et il lance le temps de surveillance du cycle.

Le fonctionnement cyclique de la CPU comprend 3 étapes principales :

- 1 - La CPU interroge l'état des signaux des modules d'entrées et actualise la mémoire image des entrées MIE.
- 2 - Elle exécute le programme utilisateur avec ses différentes opérations.
- 3 - Elle copie les valeurs de la mémoire image des sorties MIS dans les modules de sortie.

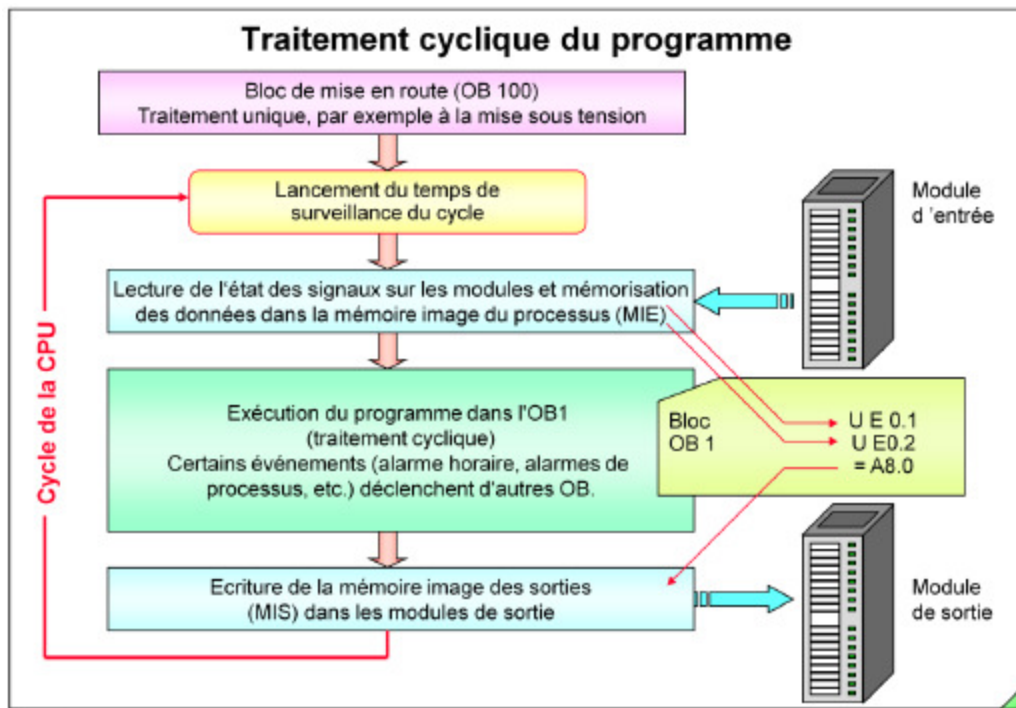


FIG (III-19) : Traitement cyclique du programme.[10]

**III-8-6-6 Langages de programmation:**

Le logiciel STEP7 offre quatre langages de programmation:

- CONT (LD Ladder Diagram).
- LOG (FBD Function Bloc Diagram).
- LIST (IL Instruction List).
- GRAPH (GRAFCET).

D'autre langage de programmation peuvent être procuré sous forme de logiciel additionnel : le SCL (ST Structured Text).

**A-Langage de programmation CONT (schéma à contacts):**

La représentation en langage de programmation CONT (schéma à contacts) s'inspire des schémas de circuits, les éléments d'un schéma de circuit, tels que contacts à fermeture et contacts à ouverture, sont rassemblés dans des réseaux, un ou plusieurs réseaux forment la section des instructions complète d'un bloc de code, la programmation en CONT est la méthode choisie par de nombreux programmeurs d'automates programmables et par le personnel de maintenance.

**B-Langage de programmation LOG (logigramme):**

Le langage de programmation LOG (logigramme) utilise les boites fonctionnelles graphiques de l'algèbre booléenne pour représenter des éléments logiques. Il permet en outre de représenter des fonctions complexes, telles que les fonctions mathématiques en les mettant directement en liaison avec ces boîtes logiques.

**C-Langage de programmation LIST (liste d'instructions):**

Le langage de programmation LIST (liste d'instructions) est un langage textuel proche du langage machine. Chaque instruction correspond à une étape de l'exécution du programme par la CPU. Vous pouvez regrouper plusieurs instructions en réseaux.

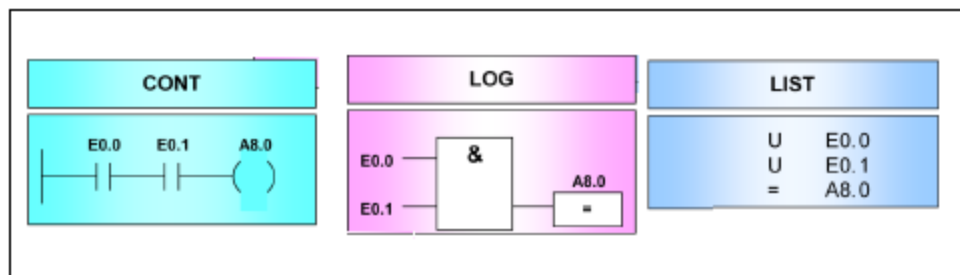


FIG (III-20) : Représentation des langages de programmation step7

**III-8-6-7 L'instruction:**

La tâche d'automatisation est divisée en plusieurs instructions en vue de son traitement par l'automate programmable, L'instruction est une unité autonome du programme d'automatisation, elle équivaut à un ordre pour l'unité de commande, les désignations, identificateurs et mnémoniques des instructions sont définis dans la norme DIN 19 235.

L'instruction se compose comme suit ( Tableau (III- ) ) :

Instruction		
Partie type opération	Partie opérande	
	Identificateur	Paramètre
<b>U</b>	<b>E</b>	<b>0.0</b>

Tableau (III- 3 ) : Les parties de l'instruction.

**A-Partie type opération:**

L'opération décrit la fonction à exécuter, la norme DIN 19 239 fait la distinction entre :

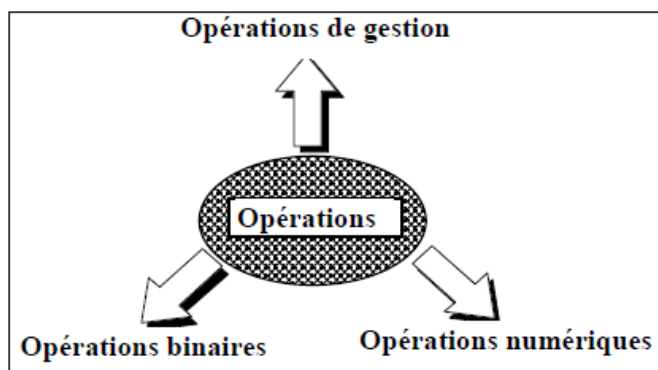


FIG (III-21) : Les types des opérations.

Exemple d'opérations numériques :

L.....charger.

T.....Transférer.

==R.....Comparaison "égale" à nombres réels.

Exemple d'opération binaires :

U.....ET.

N.....NON.

=.....AFFECTATION.

Exemples d'opérations de gestion :

- CC .....Appel conditionnel de bloc.
- UC .....Appel inconditionnel de bloc.
- AUF .....Ouverture du bloc.
- SPA..... Saut inconditionnel.
- SPB ..... Saut conditionnel.

**B-Partie opérande:**

La partie opérande contient toutes les informations nécessaires à l'exécution de la fonction, il indique à l'unité de commande sur quel objet l'opération doit porter, l'identificateur d'opérande indique le type de l'opérande, il s'agit par exemple : **E** pour entrées. **A** pour sorties **M** pour mémentos.

**III-8-6-8 L'éditeur de programme (CONT/LIST/LOG):**

C'est dans l'éditeur de programme CONT/LIST/LOG que vous programmez les blocs, Vous voyez représentée ici à titre d'exemple la vue CONT.

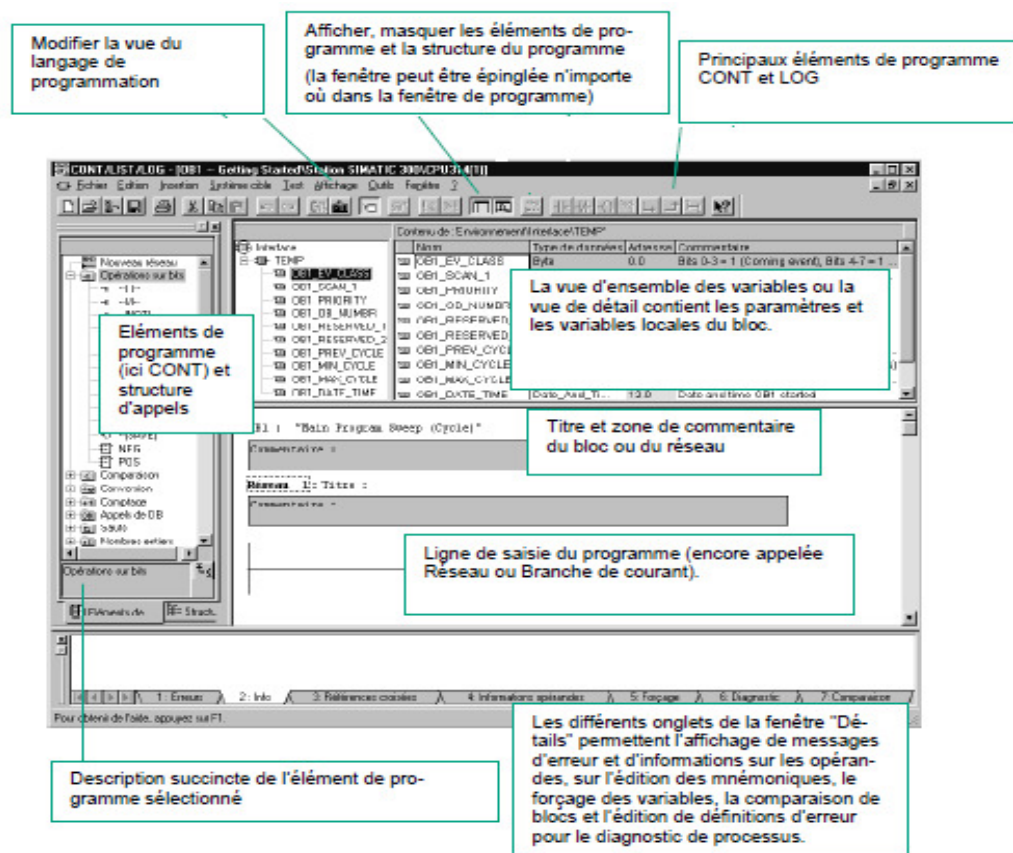


FIG (III-22) : Structure de l'éditeur de programme CONT/LIST/LOG.

Dans la section instructions, que vous entrez le programme de votre bloc scindé en réseaux, sous forme d'instructions LIST ou d'éléments CONT, l'éditeur incrémental LIST ou CONT vérifie la syntaxe après l'entrée de chaque instruction ou d'un élément CONT et affiche les erreurs éventuelles en rouge et en italique. Ces erreurs de syntaxe sont à corriger avant d'enregistrer le bloc.

Dans la section instructions d'un bloc, vous pouvez saisir l'intitulé du bloc, les commentaires du bloc, le nom des réseaux, les commentaires de réseaux ainsi que les instructions LIST ou les éléments CONT des différents réseaux ( figure (III- )).

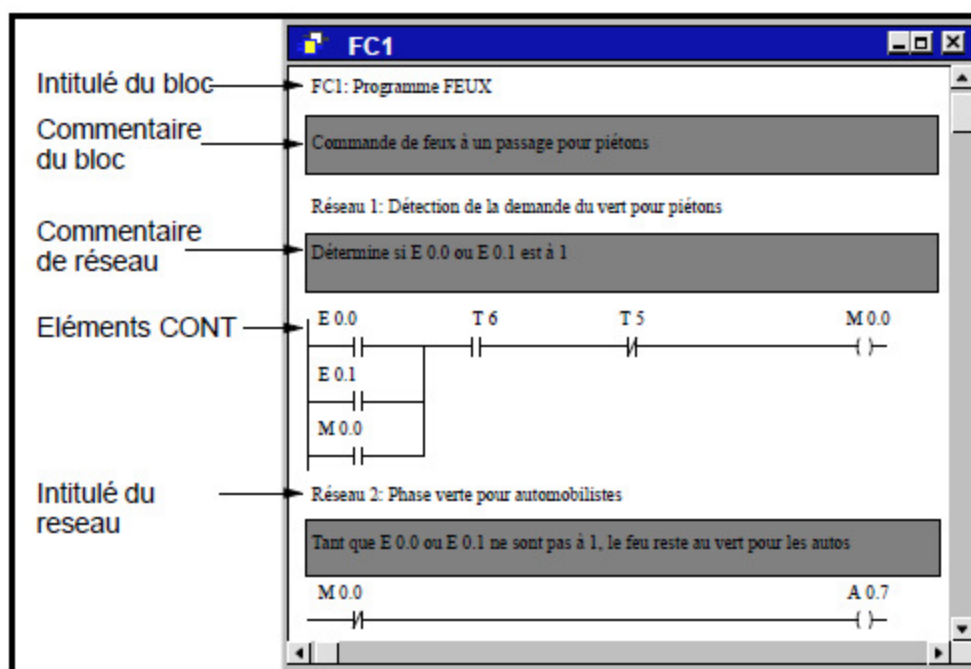


FIG (III-23) : Structure de la section instructions d'un bloc CONT

Nous avons choisi le langage à contact pour la programmation car il s'agit du langage le plus simple, il peut être compris par n'importe qui ayant des notions de logique, il est le plus facile à débogger et permet une vision facile en temps réel avec l'installation.

**III-8-6-9 Quelques exemples du programme :**

The screenshot displays the SIMATIC Manager interface for configuring the OB1 (Organization Block) in a SIMATIC 300 system. The main window shows a table of variables and their addresses, followed by a list of function calls (FC) for various machine functions. Below this, two ladder logic diagrams are shown: 'Réseau 2: cycle automatique' and 'Réseau 3: controle cycle automatique'.

Nom	Type de données	Adresse	Commentaire
TEMP0	Byte	0.0	bites 0-3=1 (coming event),bites 4-7=1(event class1)
TEMP1	Byte	1.0	1(cold restart scan 1 of OB 1),3(scan 2-n of OB1)
TEMP2	Byte	2.0	1(priority of Iis lower)
TEMP3	Byte	3.0	1(organisation bloc 1,OB1)
TEMP4	Byte	4.0	reserved for system
TEMP5	Byte	5.0	reserved for system
TEMP6	Int	6.0	cycle time of previous OB1 scan (milliseconds)
TEMP7	Int	8.0	minimum cycle time of OB1 (ms)
TEMP8	Int	10.0	maximum cycle time of OB1 (ms)
TEMP9	Date_And_Time	12.0	date time OB1 started

OB1 : bloc d'organisation  
 gérer le traitement cyclique du programme

Réseau : Titre :

```

CALL FC 1      vib/pres/ table rot
CALL FC 2      chariot
CALL FC 3      compteur
CALL FC 4      doseur
CALL FC 6      demoleur hydrolique
CALL FC 7      tremie
CALL FC 8      nettoyage station demoul
CALL FC 9      vibreur polongeat
CALL FC 11     start stop autom cycl
CALL FC 12     nettoyage table
CALL FC 14     alarmes
CALL FC 16     identification
CALL FC 25     nettoyage moule demoul
CALL FC 26     guide oprateur
CALL FC 27     time out
CALL FC 28     fonction complete
    
```

Réseau 2: cycle automatique

Réseau 3: controle cycle automatique

FIG (III-24) : Bloc de code OB1 (organisation)

Contenu de : 'Environnement\Interface\TEMP'

Nom	Type de données	Adresse	Commentaire
TEMP0	Byte	0.0	16#39,event class 3,entering event state,internal fault event
TEMP1	Byte	1.0	16#XX ,fault identification code
TEMP2	Byte	2.0	26/28 (priority of iis lowest)
TEMP3	Byte	3.0	81 (organisation bloc 81 ,OB81)
TEMP4	Byte	4.0	reserved for system
TEMP5	Byte	5.0	reserved for system
TEMP6	Int	6.0	reserved for system

OB81 : power supply fault

Commentaire :

```

#M69.1 : alarme batterie
L #TEMP0          #TEMP0      -- 16#39,event class 3,entering event state,internal fault event
L B#16#39
==I
SPBN M001
L #TEMP1          #TEMP1      -- 16#XX ,fault identification code
L B#16#22
==I
= "alarm batterie CPU"      M68.5  -- mém alarm batterie CPU
M001: NOP 0
    
```

Opérande /	Bloc	Type	Occurrence	Occurrence	Occurrence
alarm batterie...	alarmes	R	Ré 1 Inst 68 /U		
	PS FLT	W	Ré 1 Inst 8 /=		

FIG (III-25) : Bloc de code OB 81 (défaut d'alimentation)

Réseau 6 : EV MOUVEMENT AVANT DOSEUR

Diagramme de logique de commande montrant des contacts M69.1, E4.4, E4.3, E4.6, E9.7, E6.5, E7.7, M10.1, A14.5, et A14.4.

Opérande /	Bloc	Ty	Occurrence	Occurrence	Occurrence
DB21.DBW44	doseur	R	Ré 5 /L		
decharge/souf...	doseur	R	Ré 2 /U		
EV soufleur do...	doseur	R	Ré 4 /O	Ré 5 /U	
		W	RA 3 /=		

FIG (III-26) : Bloc de code FC 4 (doseur)

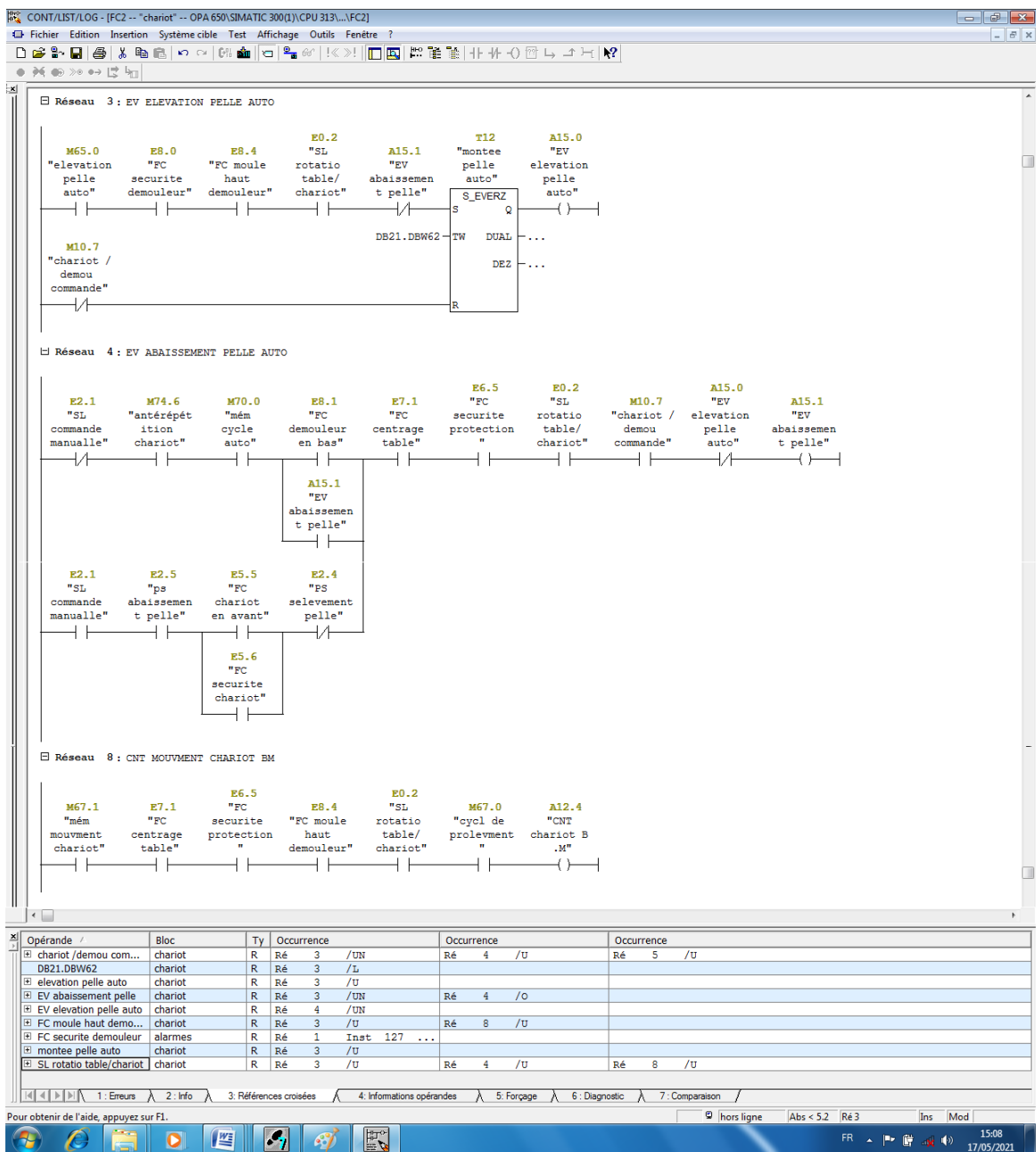


FIG (III-27) : Bloc de code FC2 ( chariot)

**III-8-7 Simulation de programme avec S7-PLCSINI:**

L'application de simulation de modules S7-PLCSIM permet d'exécuter et de tester le programme dans automate programmable (API) simulé dans un ordinateur ou dans une console de programmation, La simulation étant complètement réaliser au sein du logiciel STEP7. il n'est pas nécessaire une liaison soit établie avec un matériel S7 quelconque (CPU ou module des signaux), Ceci peut avoir différentes raisons :

- Petits blocs de programme qui ne peuvent pas encore être testés dans une séquence unique sur la machine.
- L'application est critique, car elle peut occasionner des dommages matériels ou blessures corporelles en cas d'erreurs de programmation , La simulation permet de supprimer ces erreurs dès la phase de test.

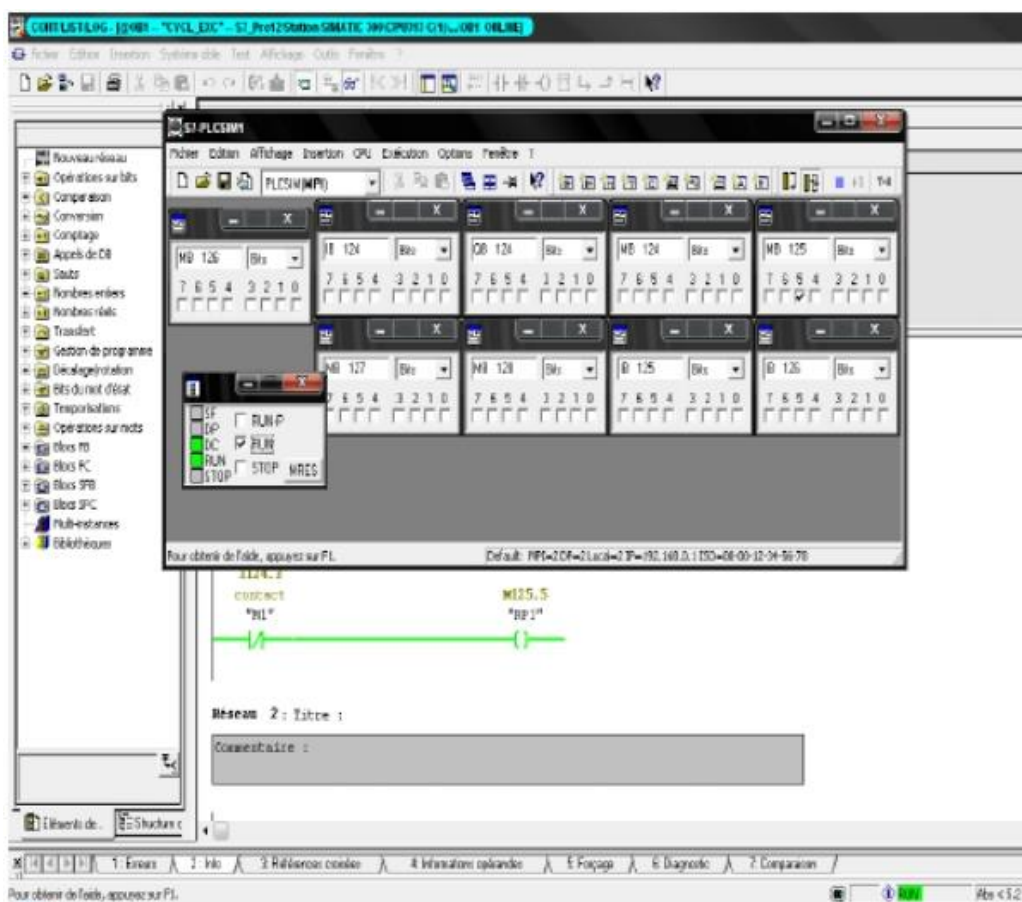


Fig (III-28) : lancement de l'application avec S7 –PLCSIM.

**Conclusion :**

Nous avons réalisé la tâche de programmation de l'automate programmable industriel par l'outil « STEP7 » de SIMATIC Manager, la conception d'une interface de communication entre l'opérateur et la presse est nécessaire pour l'efficacité de l'application déjà développée. La conception et la programmation de cette interface feront objet du chapitre suivant.

# Chapitre IV

# Supervision de la presse

**Introduction :**

Le système automatisé est conçu pour fonctionner de manière autonome sans intervention humaine, l'homme doit pouvoir communiquer avec ce système, pour échanger les informations. Pour cela l'échange d'information entre le processus et l'opérateur a été amélioré à travers l'Interface homme-machine.

Notre objectif dans ce chapitre est de réaliser un système de supervision pour la PRESSE ROTATIVE à l'aide du logiciel WinCC flexible.

**IV-1 Interface Homme Machine:**

À cause de la complexité du processus dans les industries, comme dans le cas de la presse rotative de carreau, l'opérateur a besoin de plus de transparence pour avoir une visualisation claire ; c'est à dire une hiérarchie des messages bien définie et des commandes intuitives.

Il est possible d'obtenir cette transparence grâce à l'interface homme-machine où la partie "Homme " est l'opérateur et la partie "Machine " le processus ou l'installation. Le principal objectif de cette interface est de couvrir toutes les tâches de contrôle commande et de maintenir les machines et les installations en état de marche.

**IV-2 La supervision :**

La supervision est une forme évoluée de dialogue homme-machine. Elle consiste à surveiller l'état de fonctionnement d'un procédé dont les possibilités vont bien au-delà de celles des fonctions de conduite et de surveillance réalisées avec les interfaces. Les fonctions de la supervision sont nombreuses, on peut citer quelques uns :

- Assiste l'opérateur dans les opérations de diagnostic et de maintenance.
- Assure la communication entre les équipements d'automatisme et les outils informatiques d'ordonnancement et de gestion de production.
- Elle répond à des besoins nécessitant en général une puissance de traitement importante.
- Coordonne le fonctionnement d'un ensemble de machines enchainées constituant une ligne de production, en assurant l'exécution d'ordre commun (marche-arrêt) et des tâches telles que la synchronisation

### **IV-3 Avantages de la supervision :**

Un système de supervision apporte une aide à l'opérateur dans la conduite du processus son but étant de présenter à l'opérateur des résultats expliqués et interprétés. Ses principaux avantages sont :

- La détection des défauts.
- Le diagnostic et le traitement des alarmes.
- Surveiller le processus à distance.

**Remarque :** Pour ce qui est de ce dernier point (à savoir la surveillance du Processus à distant), les logiciels de supervision ont des options qui permettent même la tél éconduite et la télémaintenance via internet (par exemple : Sm@rtservice et Sm@rtAccess de WinCC flexible).

### **IV-4 Structure d'un système de supervision:**

La plus part des systèmes de supervisions se composent d'un moteur central (logiciel), auquel se rattachent des données provenant des équipements (automates).

Le logiciel de supervision assure l'affichage, le traitement des données, l'archivage et la communication avec d'autres périphériques( FIG (IV-1) ).

#### **IV-4-1 Le module de visualisation (affichage) :**

Il permet d'obtenir et de mettre à la disposition de l'opérateur des éléments d'évaluation du procédé par ses volumes de données

#### **IV-4-2 Le module d'archivage :**

Il mémorise les données (alarme et événement) pendant une longue période. Il permet l'exploitation des données pour les applications spécifiques à des fins de maintenance ou de gestion de la production.

#### **IV-4-3 Le module de traitement :**

Il permet de mettre en forme les données afin de les présenter via le module de visualisation aux opérateurs sous une forme prédéfinie.

#### **IV-4-4 Module de communication :**

Il assure l'acquisition et le transfert de données, ainsi qu'il gère la communication avec les automates programmables industriels et autres périphériques.

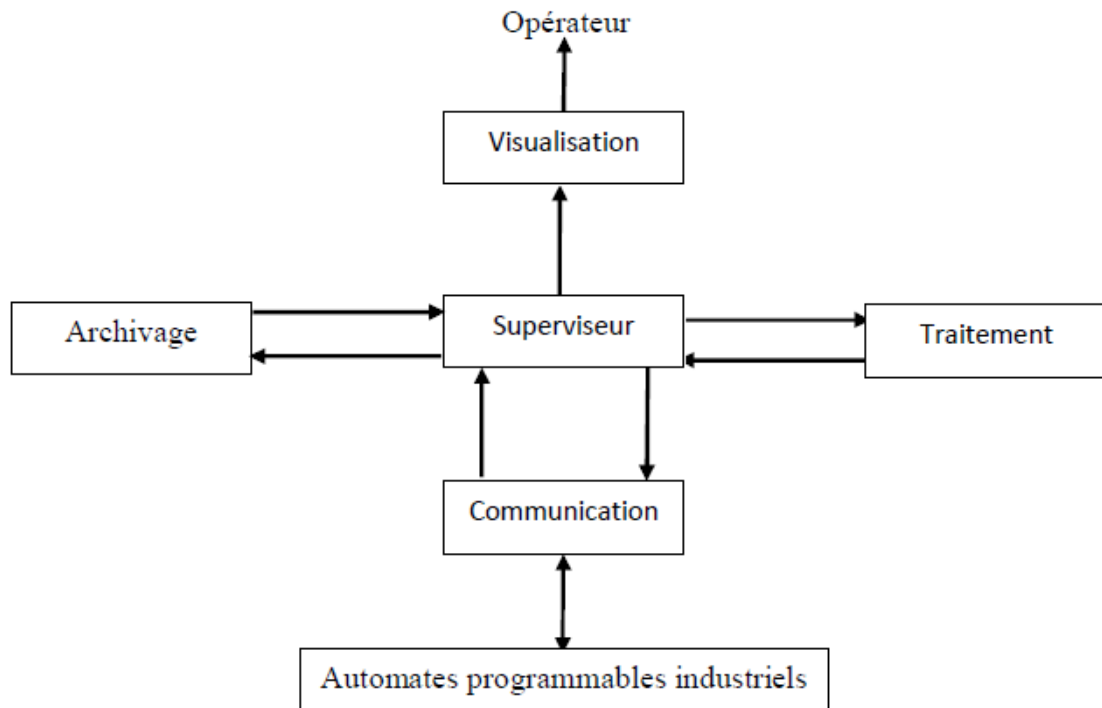


FIG (IV.1) : Structure d'un système de supervision.

### **IV-5 Choix de l'Interface Homme Machine :**

SIMATIC HMI (Human Machine Interface) propose une vaste gamme de pupitres opérateur et d'ordinateurs pour toutes les applications de conduite et de supervision. Il trouve dans cette gamme plusieurs types des pupitres programmables via trois logiciels ; protocol, WinCC, WinCC Flexible.

Nous avons choisi d'utiliser le logiciel WinCC pour la réalisation, par des moyens d'ingénierie simples et efficaces, de concepts d'automatisation évolutifs, au niveau machine. WinCC réunit les avantages suivants : • simplicité • ouverture • flexibilité

### **IV-6 Supervision sous Wincc flexible :**

#### **IV-6-1 Description de Wincc :**

Wincc (Windows Control Center) est un logiciel de supervision développé par la firme SIEMENS, il est caractérisé par sa flexibilité, c'est-à-dire qu'il peut être utilisé par un composant hors SIEMENS.

WINCC (Windows Control Center) est la première interface homme machine (IHM)

qui intègre véritablement la supervision et le processus d'automatisation. Il permet la saisie, l'affichage et l'archivage des données tout en facilitant les tâches de conduite et de surveillance aux exploitants.[5]

Ce logiciel offre une bonne solution de supervision car il met à la disposition de l'opérateur des fonctionnalités adaptées aux exigences d'une installation industrielle. Il est composé de deux éléments principaux :

**a- WinCC flexible Engineering System :** WinCC flexible Engineering System est le logiciel avec lequel on réalise toutes les tâches de configuration requises après que l'édition WinCC flexible détermine les pupitres opérateurs de la gamme SIMATIC HMI à configurer.

**b- WinCC flexible Runtime :** WinCC flexible Runtime est le logiciel de visualisation de processus, ou on exécute le projet en mode processus.

#### IV-6-2 Choix de pupitre de commande :

Le pupitre de commande permet à l'opérateur de visualiser à partir de synoptiques, l'évolution des différents paramètres du système. Il est l'interface de dialogue homme machine. En effet, c'est à travers lui que l'opérateur communique avec le système.

Pour notre application nous avons choisi le pupitre opérateur SIMATIC Multi Panel de type MP2708 TOUCH . La série SIMATIC MP de Siemens est livrée dans une variété d'écrans avec fonctionnement tactile. Tous les panneaux SIMATIC MP sont conformes à la norme IP65 pour des conditions difficiles et sont flexibles pour n'importe quelle application en extérieur. Ces nouvelles interfaces graphiques sont capables de coordonner et d'arrêter leurs écrans de manière centralisée via PROFIenergy pendant les pauses, afin de réduire la consommation d'énergie par rapport aux panneaux SIMATIC précédents.

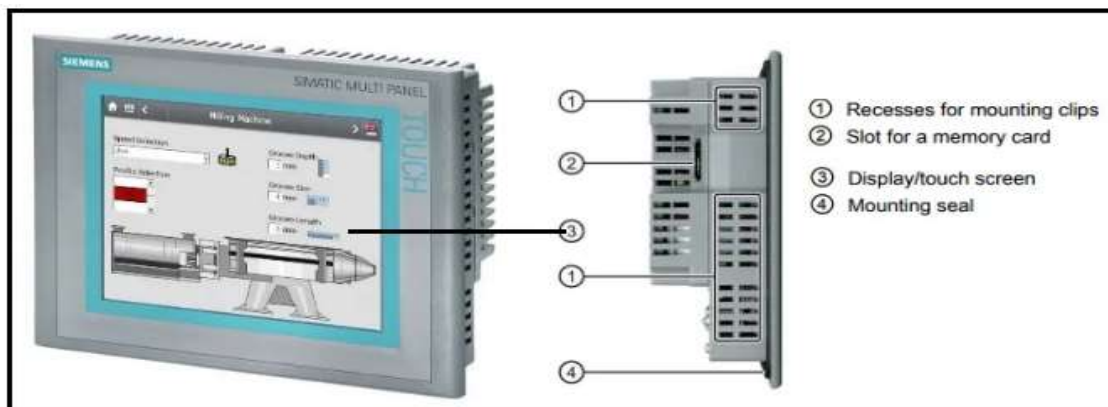


FIG (IV-2): Le pupitre opérateur MP270 8 touch

### IV-6-3 Intégration de WinCC flexible dans STEP7 :

WinCC flexible peut être intégré au logiciel de configuration SIMATIC STEP7, cela permet de choisir des mnémoniques et blocs de données de SIMATIC STEP7 comme variables dans WinCC flexible, on économise non seulement le temps et l'argent, mais on évite des sources d'erreurs dues à la répétition de la saisie.

#### IV-6-3-1 Avantages de l'intégration au STEP7 :

Lors de la configuration intégrée, on a accès aux données de configuration que l'on a créés lors de la configuration de l'automate avec STEP 7, on profite des avantages suivants :

- On peut utiliser le gestionnaire SIMATIC Manager comme poste central de création, d'édition et de gestion des automates SIMATIC et des projets WinCC flexible.
- Les paramètres de communication de l'automate sont entrés par défaut lors de la réaction du projet WinCC flexible, Toute modification sous STEP 7 se traduit par une mise à jour des paramètres de communication sous WinCC flexible.
- Lors de la configuration de variables et de pointeurs de zone, on peut accéder sous WinCC flexible directement aux mnémoniques de STEP 7, on sélectionne simplement sous WinCC flexible le mnémonique STEP 7 auquel on veut affecter une variable. Les modifications de mnémonique sous STEP 7 sont mises à jour sous WinCC flexible.
- Il suffit de définir les mnémoniques une seule fois sous STEP 7 pour pouvoir les utiliser sous STEP 7 et sous WinCC flexible.
- Les alarmes ALARM\_S et ALARM\_D configurées sous STEP 7 sont prises en charge sous WinCC flexible et peuvent être affichées sur le pupitre opérateur

### IV-6-4 Application la supervision sous WinCC Flexible 2008 :

#### IV-6-4-1 Insertion d'une station HMI dans SIMATIC manager :

Pour insérer une station HMI, il faut Suivez ces étapes (FIG(IV-3)) :

1. Dans le simatic manager Cliquer sur «projet » .
2. Cliquer sur «insertion ».
3. Cliquer sur «Station ».
4. Choisir «Station SIMATIC HMI ».

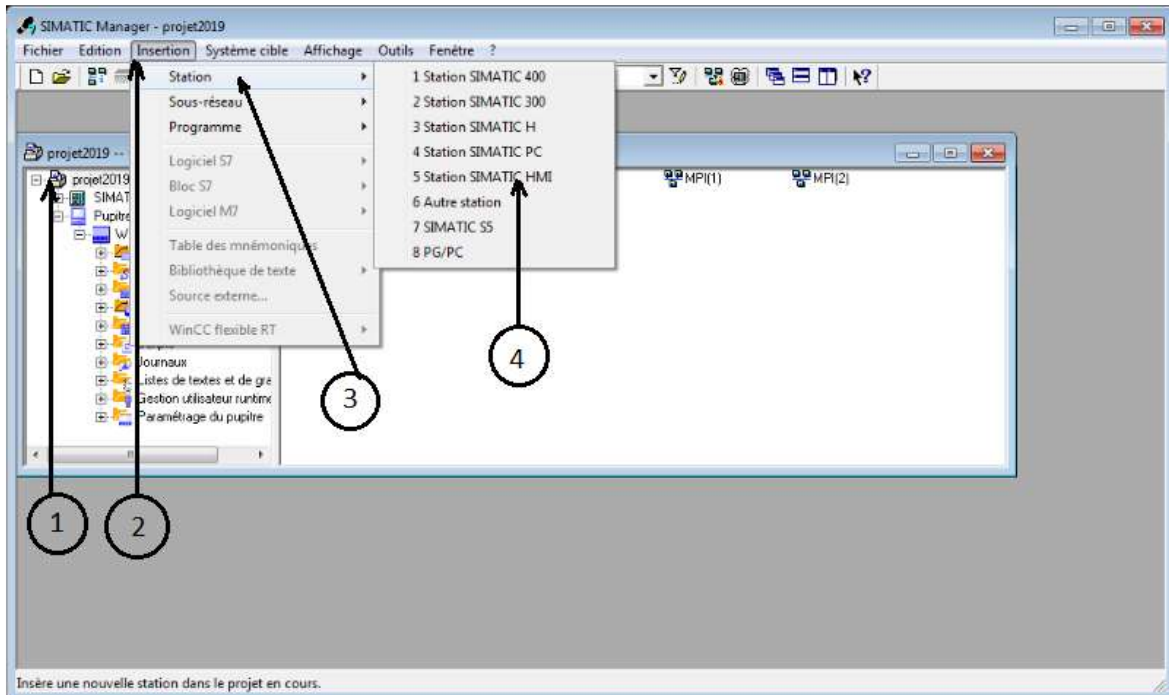
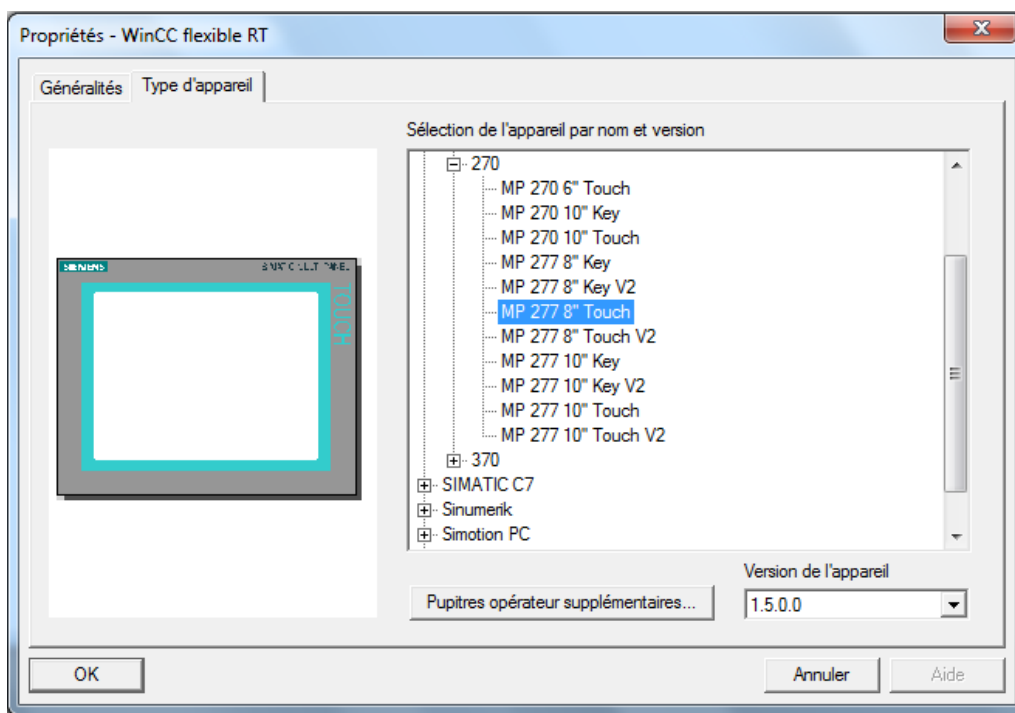


FIG (IV-3): Insertion d'un nouveau Station.

**IV-6-4-2 Choix de la station (pupitre) :**

Une fenêtre de sélection du type de pupitre s'ouvre pour nous permettre de choisir un pupitre adéquat à notre application (FIG IV-4).



FIG( IV-4) : Choix de pupitre.

Après validation du type du pupitre l'espace de travail WinCC Flexible Advanced projet-HMI s'ouvre.

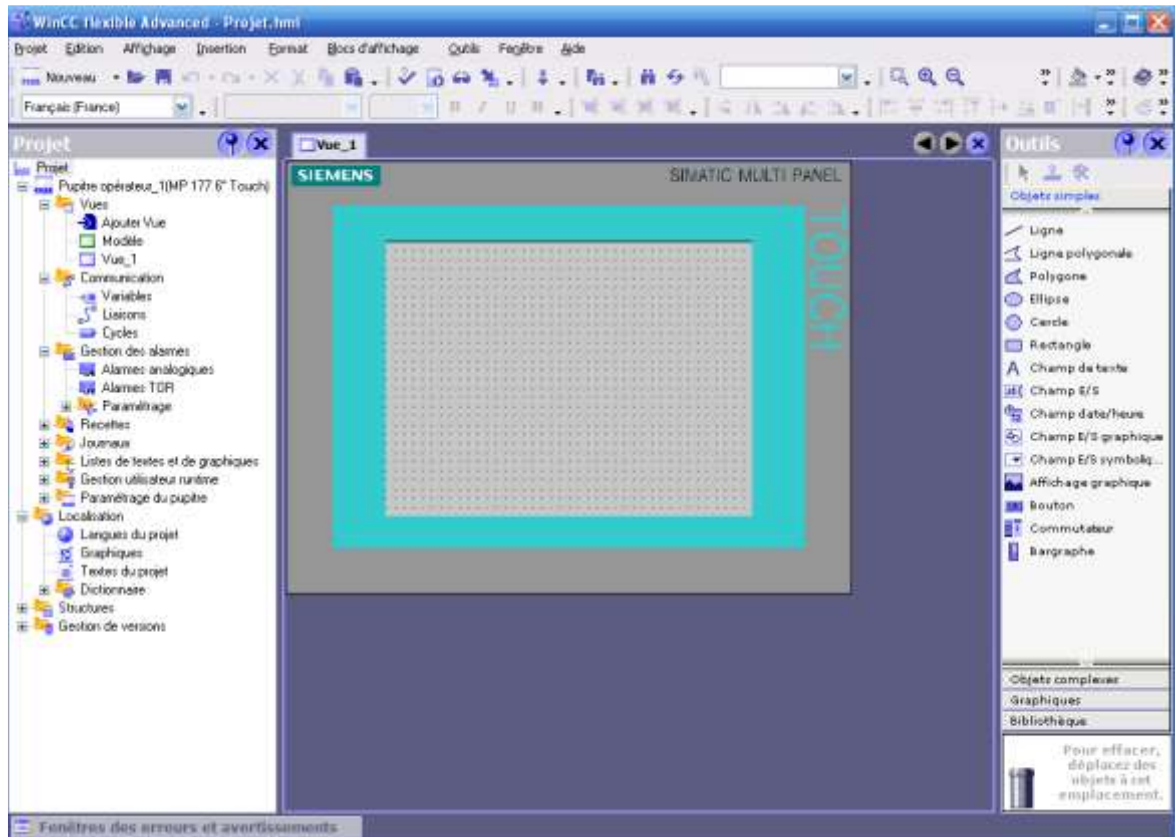


FIG (IV -5) : Espace de travail

L'espace de travail de WinCC flexible nous offre tous les outils nécessaires à la présentation d'un quelconque système automatique, mécanique, hydraulique et autres.

L'insertion de la maquette du système sera suivie des configurations de ces différents paramètres soit de mise en marche ou de communication avec l'automate programmable grâce à des éditions des différentes variables intervenant dans le processus du système à automatiser.

#### IV-6-4-3 Configuration du réseau :

Pour la configuration des réseaux de WinCC flexible, vous devez suivre les étapes suivantes (montrées dans la FIG IV-6) :

- 1- Cliquer sur «Pupitre opérateur 1».
- 2- Cliquer sur «Configuration».
- 3- Cliquer sur «NETPRO».

4. Cliquer sur le carré ROUGE, ensuite glisser vers la ligne ROUGE .
5. Enfin enregistrer et compiler .

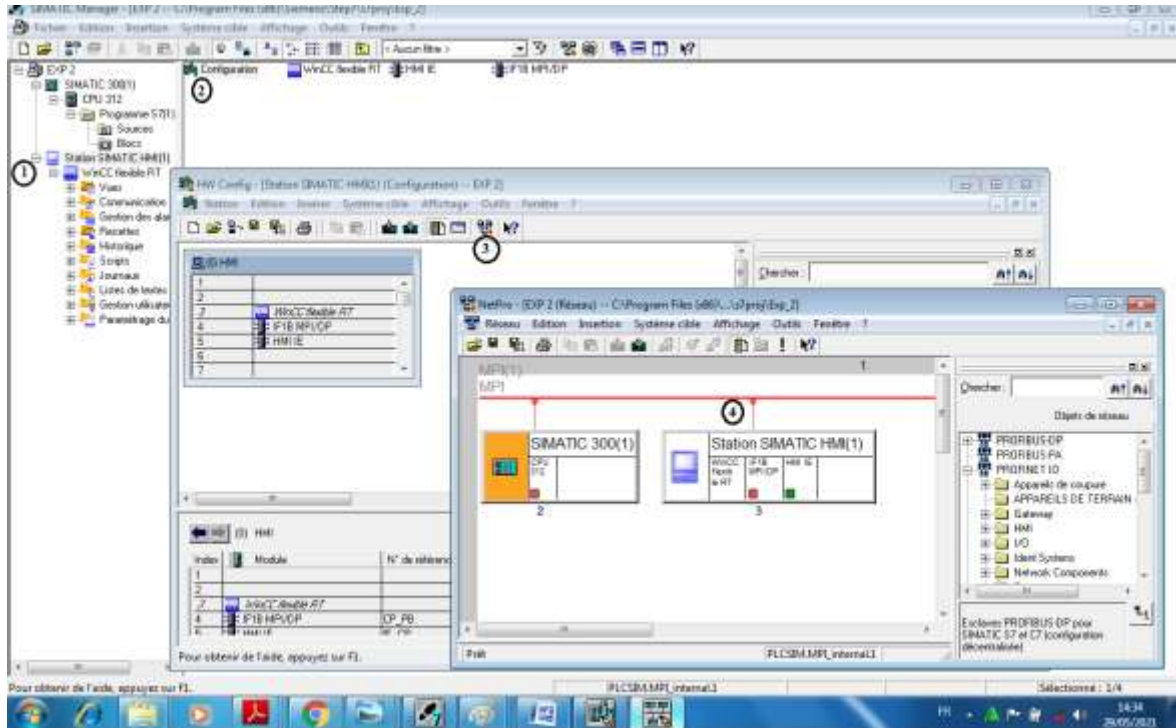


FIG (IV -6) : la configuration des réseaux

#### IV-6-4-4 Création des vues :

La supervision de notre projet est constituée de quatre vues :

- vue d'accueil.
- vue de temporisation.
- vue d'état des entrées /sorties.
- vue des alarmes.

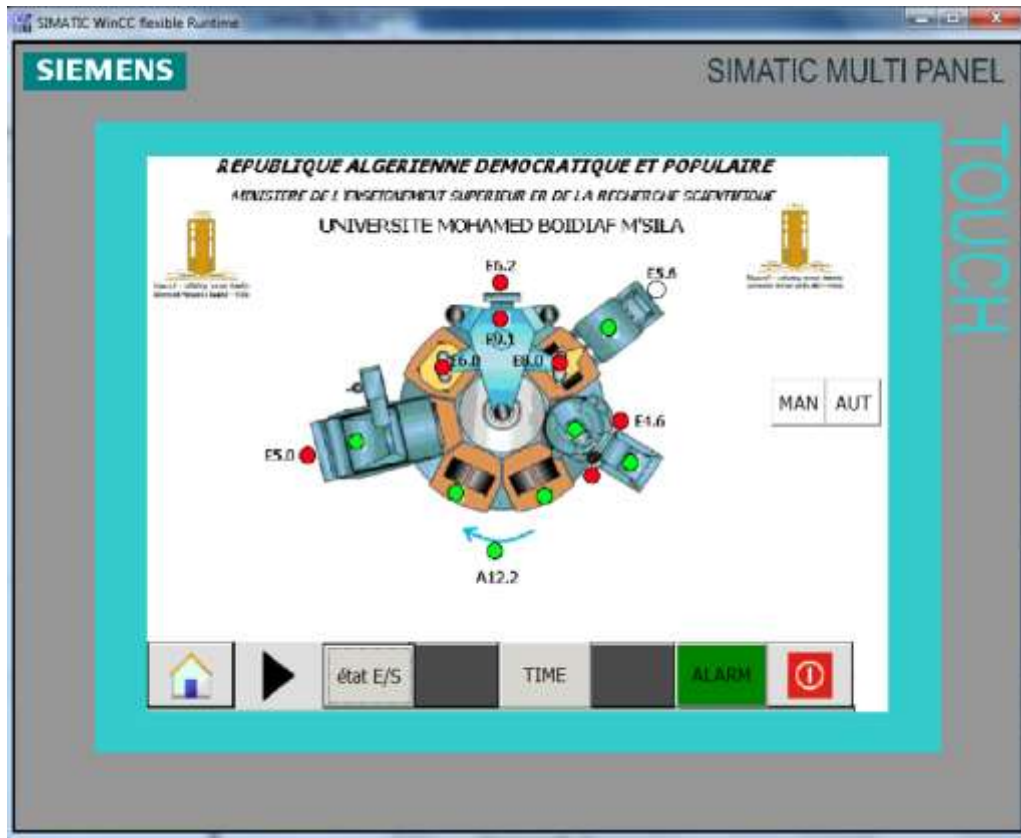


FIG (IV-7): vue d'accueil.

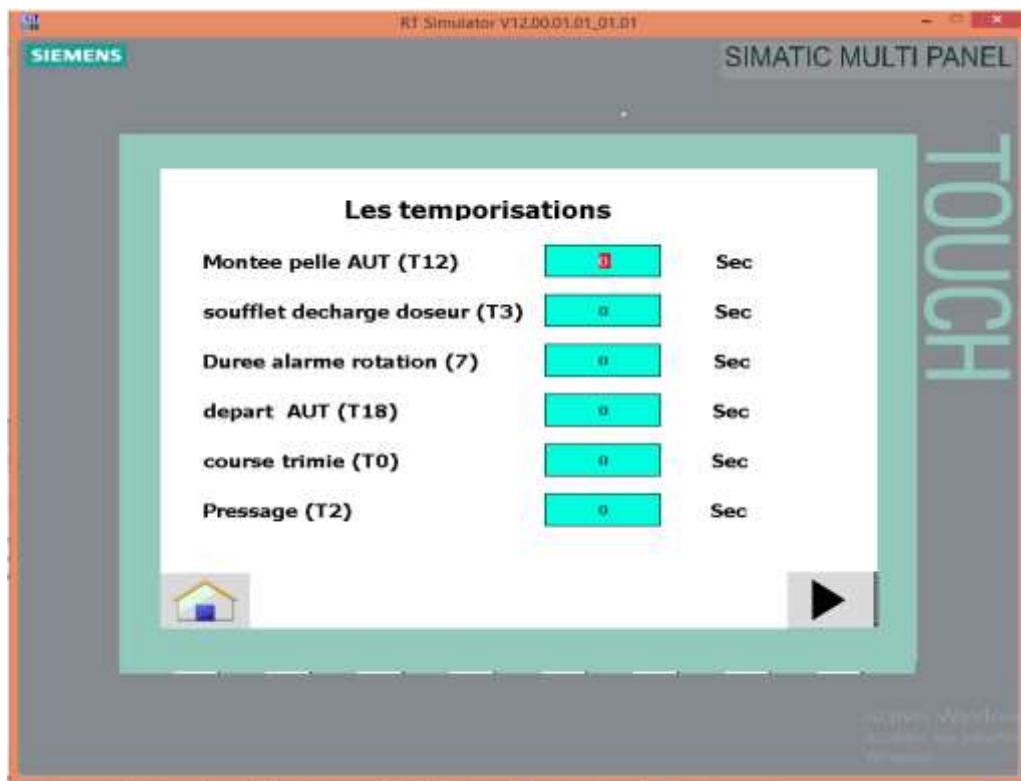


FIG (IV-8): vue de temporisation.

Sorties		Etat	Entrees		Etat
Rotation	A 12.0	●	PS montee demoleur	E 2.7	●
CNT vibr fosse tremie	A 12.0	●	PS repetition pressag	E 3.0	●
INT vibreur moules	A 12.2	●	INT consentement lina	E 3.1	●
CNT vibr fosse tremie	A 12.3	●	SL cycle automatique	E 3.2	●
CNT chariot B.M	A 12.4	●	PS Start Cycle AUT	E 3.3	●
CNT Tapis doseur	A 12.5	●	CNT malexeur doseur	E 3.4	●
CNT vibr Mob tremie	A 12.6	●	CNT pompe hydr	E 3.5	●
CNT Inverter tremie	A 12.7	●	PS desn vibreur	E 3.6	●
CNT Frein mot rot table	A 13.0	●	PS Montee vibreur	E 3.7	●
INT chariot pos AR	A 13.1	●	E 4.0	E 4.0	●
INT demoleur pos bas	A 13.2	●	E 4.1	E 4.1	●
lampe rotation auto t	A 13.3	●	SL Vibreur plangeant	E 4.2	●
Lampe sig vibration moule	A 13.4	●	SL dosage pos 1	E 4.3	●
sig acoustic rotation	A 13.5	●	SL dosage pos 2	E 4.4	●
EV premiere pressage	A 13.6	●	PS MAN doseur	E 4.5	●
EV pressage	A 13.7	●	FC doseur dech pos2/pos1	E 4.6	●

FIG (IV -9): vue d'état des entrées /sorties.

### Conclusion :

Dans ce dernier chapitre, nous avons commencé par donner quelques notions de la supervision et le rôle qu'elle occupe dans l'industrie. Ensuite, nous avons donné quelques notions de base du logiciel de supervision utilisé WinCC flexible.

Après la petite partie introductive, nous avons commencé le travail avec le logiciel WinCC flexible qui est très riche en options. Il suffit d'imaginer le design de l'installation et tous les effets d'animations qui seront nécessaires pour bien apporter l'état réel de l'installation à l'opérateur avec plus d'information à partir des messages configurés et l'attribution de couleurs différentes pour les différents états des objets.

# Conclusion générale

Ce travail de fin d'étude s'inscrit dans le cadre de l'automatisation et supervision pour but la commande à base d'un automate programmable industriel (API) S7-300 d'une presse rotative a carreaux granito,a cette fin, nous avons commencé par prendre connaissance de base sur le systeme automatise , leur structure et leur technogies ,puis nous avons passé à la description détaillée de l'architecture des automates programmables industriels (API) ,les différents langages de programmation et Les critères de choix essentiels d'un automate.

Nous avons procédé dans la deuxieme partie à l'étude technique détaillée de la presse rotative et son principe de fonctionnement, aussi les différentes parties de cette machine.

Nous avons passé en revue les automates programmables industriels de la gamme SIEMENS, leurs caractéristiques, critères de choix, avantages, ainsi que les langages de programmation utilisables

Après, nous avons élaboré la nouvelle solution qui est basée sur la programmation à l'aide du logiciel STEP 7 autour d'un automate programmable de la famille SIEMENS de la gamme SIMATIC de types S7-300. Le choix du CPU ainsi que les différents modules a été effectué selon les besoins de la commande.

Pour la simulation on a utilisé le logiciel optionnel PLCSIM de STEP7 pour visualiser les sorties après avoir forcé les entrées.

La dernière étape a été consacrée à la visualisation et la supervision la presse rotative à l'aide du logiciel WinCC flexible 2008.

Ce projet nous a permis de connaitre de près la démarche de résolution des problèmes, surtout dans un projet aussi complexe que la mise en oeuvre d'une unité industrielle, ainsi enrichir nos connaissances dans le domaine de l'automatisation industriel et de tirer profit de l'expérience des personnes du domaine.

En fin, nous souhaitons que ce modeste travail contribue à faciliter et illustrer les principes de l'automatisation et servir de documentation pour d'autres projets et pour une meilleure compréhension.

# Bibliographie

- [1] **.Mourad KCHAOU** « Introduction à l'Automatisme GRAFCET & GEMMA » Université de sousse, 2017.
- [2] **Jean-Yves Fabert** «Automatismes et automatique », 2003, Paris Ellipses.
- [3] **Mme EL HAMMOUMI** « Cours Automatismes Logiques & Industriels –GE1 » Universite sidi mohammed ben abdellah- fes,2014.
- [4] **Mr HADDOUR Ali** « Automatisation d'une station de pressurisation pour un système antiincendie avec API S7-300 » Mémoire de MASTER en Génie Electrique 2012.UNV Béjaia.
- [5] **SIEMENS** « SIMATIC HMI WinCC flexible 2008,Runtime »manul d'utilisation siemeens 2008.
- [6] **OCEM OFF COSTRUZIONI ELETTRICO MECCANICHE** « Manuel d'instructions et d'entretien de la machine OPA 650/300T » Année de fabrication 2002.
- [7] **.SIEMENS** « Mise en route STEP 7 V5.3 » Simatic Edition 01/2004.
- [8]. **SIEMENS** « programmer avec STEP7 5.1 » SIMATIC edition 08/ 2000.
- [9]. **SIEMENS** « Automate programmable S7-300 Configuration et programmation faciles » Cahier technique: Automate programmable S7-300 .
- [10] **SIEMENS** « Maintenance 1 », Cours ST-7SYH.
- [11] **ALBOUKAIRAT Mansour ,SALAH Toufik, SALEM Thameur**  
« AUTOMATISATION D'UN GROUPE ÉLECTROGÈNE PAR API S7-300 » Mémoire d'ingénieur, Université Mohamed Boudiaf M'sila 2012.
- [12] **ABDI Rabah- GUENDOUL Abdelghafour** « Automatisation et supervision d'une station de traitement des eaux à l'aide d'un automate Siemens S7-300 » Mémoire de MASTER en Génie Electrique 2017 UNV M'HAMED BOUGARA-BOUMERDES .

MEMOIRE DE FIN D'ETUDES EN VUE DE L'OBTENTION DU DIPLÔME

MASTER 2 EN GENIE ELECTRIQUE

**OPTION : ELECTROMECHANIQUE**

Dirigé par : **ZOGUAR ELOUALID**

Présenté par : **NADJI FOUAD & ZEMIH SAID**

**Thème : Etude d'automatisation de la presse rotative a carreau granito a l'aide d'un automate S7 300.**

**Résumé** : De nos jours, les procédés industriels évoluent de plus en plus avec l'amélioration de la technologie. Pour cela la plupart des sociétés industrielles ont pris pour but de frôler le sommet de la technologie en améliorant les anciennes machines et méthodes ; voir, créer des machines plus développées.

Ce projet concerne la conception et la commande par API SIEMENS d'une presse rotative a carreau. le travail présenté était principalement attaché à faire remplacer le système de commande de la presse rotative (logique câblée) par un automate programmable industriel S7-300.

**Mots Clés** : la technologie, presse rotative a carreau, le système de commande, logique câblée, un automate programmable industriel S7-300.

**ملخص:** اليوم ، تتطور العمليات الصناعية أكثر فأكثر مع تحسين التكنولوجيا. لهذا الغرض ، اتخذت معظم الشركات الصناعية هدفاً يتمثل في استخدام أعلى التقنيات من خلال تحسين الآلات والأساليب القديمة لإنشاء آلات أكثر تطوراً يتعلق هذا المشروع بالتصميم والتحكم بواسطة المتحكم سيمنز لمكبس البلاط الدوار. كان العمل المقدم مرتبطاً بشكل أساسي باستبدال نظام التحكم في المطبعة الدوارة (المنطق السلكي) بوحدة تحكم صناعية قابلة للبرمجة

**الكلمات الرئيسية:** التكنولوجيا ، مكبس البلاط الدوارة ، نظام التحكم ، المنطق الصلب ،