

الجمهورية الجزائرية الديمقراطية الشعبية  
République Algérienne démocratique et populaire

وزارة التعليم العالي والبحث العلمي  
Ministère de l'enseignement supérieur et de la recherche scientifique

جامعة سعد حطاب البليدة  
Université SAAD DAHLAB de BLIDA

كلية التكنولوجيا  
Faculté de Technologie

قسم الإلكترونيك  
Département d'Électronique



## Mémoire de Master

Filière :Automatique & Système

Spécialité :Automatique

Présenté par :

**Bouguerra Fouad & AchourChakib**

---

# Automatisation d'une ligne de mélangeur a base d'un API

---

Proposé par :

Mr MADDI Abdelkader

Mme BEDRANI Imène

Année Universitaire : 2020-2021

## Remerciements

---

Après avoir rendu grâce à dieu le tout puissant et le miséricordieux, qui nous a donné la force et la patience d'accomplir ce projet de fin d'étude, nous tenons à remercier en premier lieu nos parents de nous avoir encouragé, sans eux, on n'en serait pas là. On remercie également nos encadreurs Mr Maddi A. et Mme Bedrani I. pour ses précieux conseils et aides durant toute la période du travail.

---

ملخص: تشير الأتمتة الصناعية إلى استخدام وحدة تحكم منطقية صناعية قابلة للبرمجة تقوم بعمليات صناعية دون تدخل بشري. ومن هنا كان هدف عملنا، والذي يتمثل في التشغيل الآلي الكامل للخلاط الصناعي اليدوي من خلال ضمان سلامة الماكينة وإتباع نهج وصفي وتوضيحي وتحليلي منظم حول أربعة فصول. تركز بشكل أساسي على العموميات الأساسية للأتمتة، وأجهزة التحكم المنطقية القابلة للبرمجة والخلاط الصناعي لأداء محاكاة وبرمجة مما يسمح أخيرًا بالأتمتة الكاملة لعمل هذا الخلاط.

كلمات المفاتيح : أتمتة ، برمجة ، خلاط

---

**Résumé :** L'automatisme industriel désigne l'usage d'un automate programmable industriel effectuant des processus industriels sans intervention de l'Homme. D'où l'objectif de notre travail qui consiste à automatiser entièrement le fonctionnement d'un mélangeur industriel manuel en assurant la sécurité de la machine et en suivant une démarche descriptive, explicative et analytique articulée autour de quatre chapitres. Ces derniers portent essentiellement sur des généralités fondamentales correspondant à l'automatisation, l'automate programmable et au mélangeur industriel pour enfin réaliser une simulation et une programmation d'un automate permettant l'automatisation complète du travail de ce mélangeur.

**Mots Clés :** Automatisation, Programmation, Mélangeur

---

**Abstract :** Industrial automation refers to the use of an industrial programmable logic controller carrying out industrial processes without human intervention. Hence the objective of our work, which consists in fully automating the operation of a manual industrial mixer by ensuring the safety of the machine and by following a descriptive, explanatory and analytical approach, structured around four chapters. These mainly relate to fundamental generalities corresponding to automation, programmable logic controllers and industrial mixer to finally perform a simulation and programming of a PLC allowing the complete automation of the work of this mixer.

**Keywords :** Automation, Programming, Mixer

---

## Listes des acronymes et abréviations

<b>API</b>	Automate programmable industriel
<b>CPU</b>	Central processing unit
<b>LED</b>	Light emitting diode
<b>NF</b>	Normalement fermé
<b>NO</b>	Normalement ouvert
<b>NPN</b>	Negative positive negative
<b>OB</b>	Bloc d'organisation
<b>PG</b>	Programmiergeraten
<b>PLC</b>	Programmable logiccontroller
<b>PNP</b>	Positive negative positive
<b>RAM</b>	Randomaccess memory
<b>TOR</b>	Tout ou rien

# Table des matières

Remerciements

Listes des acronymes et abréviations

Liste des figures

Liste des tableaux

Introduction Générale ..... 1

## **Chapitre 1 : Généralités**

**1.1 Introduction** .....3

**1.2 Systèmes automatisés** .....4

**1.2.1 Définition d'un système automatisé** .....4

**1.2.2 Composants d'un système automatisé** .....4

1.2.2.1 La partie opérative .....4

1.2.2.2 La partie commande .....4

**1.2.3 Les capteurs** .....5

1.2.3.1 Définition d'un capteur .....5

1.2.3.2 Les différents types de capteurs .....6

1.2.3.3 Technologie.....7

**1.2.4 Actionneur** .....8

1.2.4.1 Définition d'un actionneur .....8

1.2.4.2 Types d'actionneurs .....9

**1.3 Conclusion** ..... 11

## **Chapitre 2 : Les automates programmables industriels**

<b>2.1</b>	<b><u>Introduction</u></b> .....	12
<b>2.2</b>	<b><u>Automatisation</u></b> .....	13
<b>2.2.1</b>	<b><u>Historique et objectif de l'automatisation</u></b> .....	13
<b>2.2.2</b>	<b><u>Les automates programmables industriels</u></b> .....	13
<b>2.2.2.1</b>	<i>Définition</i> .....	13
<b>2.2.2.2</b>	<i>Critère de choix de la station de commande</i> .....	17
<b>2.2.2.3</b>	<i>Nature des informations traitées par l'automate</i> .....	17
<b>2.2.2.4</b>	<i>Présentation de l'automate utilisé</i> .....	16
<b>2.2.3</b>	<b><u>Architecture des Automates</u></b> .....	17
<b>2.2.3.1</b>	<i>Structure Externe</i> .....	17
<b>2.2.3.2</b>	<i>Architecture interne d'un API</i> .....	18
<b>2.2.4</b>	<b><u>Processeur de communication (CP)</u></b> .....	26
<b>2.2.4.1</b>	<i>Signification des états de l'automate</i> .....	26
<b>2.2.4.2</b>	<i>Commutateur de mode de l'automate</i> .....	27
<b>2.2.4.3</b>	<i>Interface MPI (interface multipoint)</i> .....	27
<b>2.3</b>	<b><u>Conclusion</u></b> .....	28

## **Chapitre 3 : Présentation de la machine et Step 7**

<b>3.1</b>	<b><u>Introduction</u></b> .....	29
<b>3.2</b>	<b><u>La machine KARL SCHNELL GMBH &amp; CO</u></b> .....	30
<b>3.2.1</b>	<b><u>Présentation de la machine</u></b> .....	30
<b>3.2.2</b>	<b><u>Description de chaque partie</u></b> .....	31
<b>3.2.2.1</b>	<i>Une partie de sortie de mélangeur :</i> .....	31
<b>3.2.2.2</b>	<i>Les composants utilisés</i> .....	
<b>3.2.3</b>	<b><u>L'automate SIEMENS S7-300</u></b> .....	41
<b>3.2.3.1</b>	<i>Présentation de l'Automate</i> .....	41
<b>3.2.3.2</b>	<i>Logiciel de programmation STEP 7</i> .....	42
<b>3.2.3.3</b>	<i>Création d'un projet STEP 7</i> .....	42
<b>3.2.3.4</b>	<i>Configuration matérielle</i> .....	43
<b>3.2.3.5</b>	<i>Configurer les liaisons de communication.</i> .....	43
<b>3.2.3.6</b>	<i>Structure du programme STEP 7</i> .....	44

3.2.3.7	Les opérandes et types de données autorisés dans la table des mnémoniques .....	45
3.2.4	<b><u>Les langages de programmation d'un API</u></b> .....	46
3.3	<b><u>Conclusion</u></b> .....	48

## **Chapitre 4 : Simulation et programmation**

4.1	<b><u>Introduction</u></b> .....	49
4.2	<b><u>Simulation sous step7</u></b> .....	50
4.2.1	<b><u>Présentation</u></b> .....	50
4.2.2	<b><u>Création de la table des mnémoniques</u></b> .....	50
4.2.3	<b><u>Programme de mélangeur automatique</u></b> .....	51
4.2.3.1	Activation de la simulation la simulation .....	52
4.2.3.2	Chargement du programme .....	52
4.2.3.3	Le lancement de la simulation .....	53
4.2.4	<b><u>Visualisation</u></b> .....	54
4.2.4.1	Visualiser les variables .....	54
4.2.4.2	Visualisation de tapis .....	55
4.2.4.3	Visualisation des conditions de démarrage de mélangeur .....	55
4.2.4.4	Visualisation de démarrage vis mélangeurs 2 sens automatique ...	57
4.3	<b><u>Conclusion</u></b> .....	58
	<b><u>Conclusion Générale</u></b> .....	59

## **Bibliographies**

## Liste des figures

Figure 1.1	Structure détaillée d'un automatisme .....	4
Figure 1.2	Système Automatisé .....	5
Figure 1.3	Capteurs TOR .....	6
Figure 1.4	Capteur à levier et à tige .....	6
Figure 1.5	Branchement des capteurs de différents fils .....	7
Figure 1.6	Des actionneurs TOR .....	8
Figure 1.7	Vérin pneumatique .....	9
Figure 1.8	Vérin double effet .....	9
Figure 1.9	Vérin simple effet .....	10
Figure 2.1	Automates programmable industriels .....	14
Figure 2.2	Architecture d'un PLC .....	15
Figure 2.3	Information Toute ou Rien .....	15
Figure 2.4	Information analogique .....	16
Figure 2.5	Information numérique .....	16
Figure 2.6	Automate SIEMENS S7-300. ....	17
Figure 2.7	Des API compacts. ....	17
Figure 2.8	Des automates modulaires.....	18
Figure 2.9	Architecture d'un API .....	29
Figure 2.10	Représentation de la CPU S7-300.....	20
Figure 2.11	Châssis d'automate .....	22
Figure 2.12	Alimentation d'automate .....	22



Figure 2.13	Module entrée TOR.....	23
Figure 2.14	Module sortie TOR.....	23
Figure 2.15	Module entrée analogique.....	24
Figure 2.16	Module sortie analogique.....	24
Figure 2.17	Module de coupleur .....	25
Figure 2.18	Module FM .....	26
Figure 3.1	Les parties de mélangeur.....	30
Figure 3.2	Sortie de mélangeur .....	31
Figure 3.3	Vis de mélangeur .....	32
Figure 3.4	Pinions de mélangeur.....	33
Figure 3.5	Trappe de couvercle .....	34
Figure 3.6	Disjoncteur.....	34
Figure 3.7	Sectionneur.....	34
Figure 3.8	Fusibles.....	35
Figure 3.9	Sonde de température .....	35
Figure 3.10	Variateur de vitesse.....	36
Figure 3.11	Vanne.....	36
Figure 3.12	Moteur .....	37
Figure 3.13	Débitmètre.....	37
Figure 3.14	Contacteur .....	39
Figure 3.15	Relais thermique .....	39
Figure 3.16	Régulateur de pression .....	40
Figure 3.17	Choix de la CPU .....	41
Figure 3.18	Nom et création de projet.....	42
Figure 3.19	Configuration matériels .....	43
Figure 4.1	Déclaration des variables dans le tableau mnémorique .....	44

Figure 4.2	Déclaration des variables dans le tableau mnémorique .....	50
Figure 4.3	Appelle de bloc de programme dans OB1.....	51
Figure 4.4	Activation de la simulation .....	52
Figure 4.5	Chargement de programme.....	53
Figure 4.6	Lancement le simulateur .....	54
Figure 4.7	Table de variable et simulateur .....	55
Figure 4.8	Conditions de démarrage .....	55
Figure 4.9	Conditions des capteurs.....	56
Figure 4.10	Démarrage vis mélangeur.....	56
Figure 4.11	Ouverture des trappes .....	57
Figure 4.12	Démarrage de pompe .....	57

## Liste des tableaux

Tableau 3.1	Les différents types de variable .....	46
-------------	--	----

# Introduction générale

---

L'automatisme industriel est l'ensemble des technologies utilisant l'électronique, l'électrotechnique, la mécanique, la télécommunication, qui correspond aux automatismes séquentiels et couvrent l'ensemble des systèmes de contrôle-commande permettant de superviser ou de piloter une chaîne de production.

Ceci dans le but de concevoir des machines ou des processus automatisés qui peuvent fonctionner sans intervention humaine tout en assurant la sécurité de ses derniers.

L'automatisme industriel est présent dans plusieurs secteurs comme l'agroalimentaire avec les lignes de conditionnement par exemple, la chimie, l'industrie pétrolière, les usines de production d'électricité, l'industrie pharmaceutique etc...

Les industries qui élaborent un produit par synthèse ou mélange (l'industrie chimique, le traitement de l'eau, la pharmaceutique, etc.) ont recours à l'utilisation des agitateurs ou mélangeurs industriels.

Le mélangeur industriel, aussi appelé pétrin ou malaxeur, situé dans une cuve de mélange industriel en acier inoxydable, permet de travailler et pétrir une grande capacité de matières premières. [1]

Dans l'industrie agroalimentaire qui a fait objet de notre étude, les cuves sont essentielles à la fabrication de jus de fruit, de produits laitiers comme notre cas. L'étape du mélange doit se dérouler en évitant toute altération de la texture ou de la qualité des ingrédients.

L'objectif de notre étude consiste essentiellement à automatiser entièrement le fonctionnement d'un mélangeur industriel manuel tout en assurant la sécurité de la machine en suivant une démarche descriptive, explicative et analytique centrée sur une simulation et une programmation.

Notre question principale : **Comment automatiser le fonctionnement d'un mélangeur industriel manuel avec un automate programmable ?**

La réponse à cette question va nous permettre de mieux comprendre l'automatisation industrielle, les automates programmables et le fonctionnement d'un mélangeur industriel afin de pouvoir automatiser son fonctionnement.

Ce qui nous mène à nous interroger d'avantage, sur :

- Quel est le rôle de l'automatisation industrielle ?
- Qu'est un automate programmable et quelles sont ses principales composantes ?
- Qu'est un mélangeur industriel ?
- Comment est-il possible d'automatiser le fonctionnement d'un mélangeur industriel manuel avec un automate programmable?

Notre travail s'articule autour de quatre chapitres dont les deux premiers reprennent des généralités fondamentales relatives à l'automatisation et son rôle, les automates programmables industriels et leur constitution. Le troisième chapitre introduit le mélangeur industriel et ses principales parties ainsi que le logiciel STEP 7 utilisé dans le quatrième chapitre dans la simulation et la programmation de la machine.

# Chapitre 1 Généralités

---

## **1.1 Introduction**

Les automaticiens ont tendance à avoir une vue systémique sur les machine et les processus.

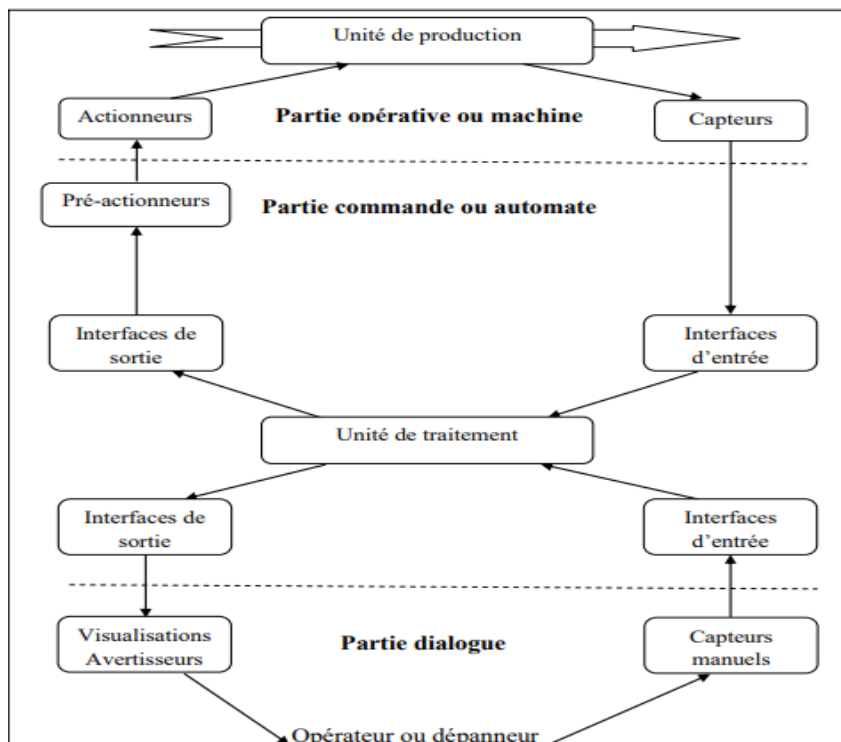
L'objectif principal, consiste à automatiser et à asservir ces systèmes afin d'améliorer la qualité et la sécurité de la machine et l'homme au même temps, aujourd'hui les composants et les capteurs sont les plus utilisé dans le monde industriel

Dans ce chapitre, nous présenterons des généralités sur les systèmes automatisés et les capteurs les plus utilisé, assurant le bon fonctionnement et la sécurité des machines.

## 1.2 Systemes automatisés

### 1.2.1 Définition d'un système automatisé

Un système est dit « automatisé » s'il exécute toujours le même cycle de travail après avoir reçu les consignes d'un opérateur. Ce système est composé d'une partie opérative « capteur » qui transmet l'information détectée à la partie de commande « API » programmée au préalable pour donner un ordre aux actionneurs. [1]



**FIGURE 1.1** : Structure détaillée d'un automatisme.

### 1.2.2 Composants d'un système automatisé

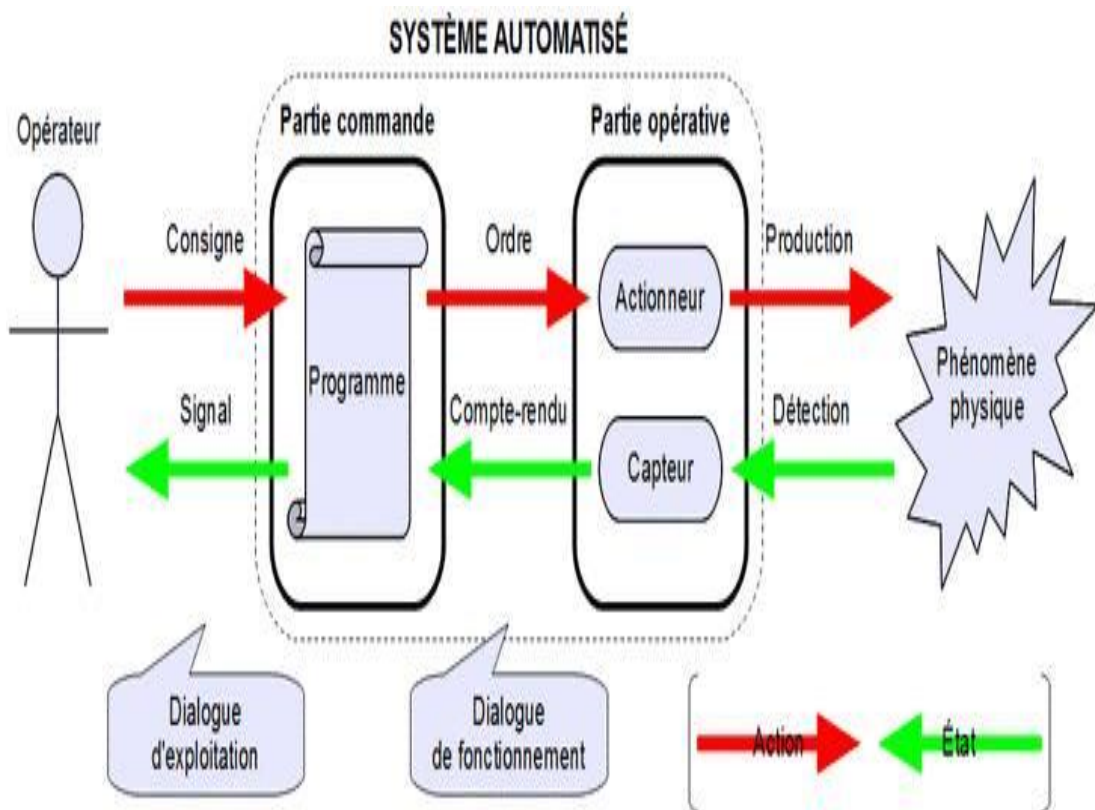
#### 1.2.2.1 La partie opérative :

La partie opérative est aussi dite « partie puissance d'un système automatisé ». Elle est composée de capteurs et d'actionneurs et est chargée d'effectuer et exécuter les ordres reçus de la partie commande.

#### 1.2.2.1 La partie commande :

C'est la partie qui gère le fonctionnement du système automatisé. Elle est en général, composée d'un ordinateur qui contient dans ses mémoires un programme. Elle transmet les ordres aux actionneurs de la partie opérative à partir :

- D'un programme qu'elle contient
- Des informations reçues par les capteurs
- Des consignes données par l'utilisateur ou par l'opérateur [8]



**FIGURE 1.2** : Système Automatisé

### 1.2.3 Les capteurs

#### 1.2.3.1 Définition d'un capteur

Un capteur est un élément de la partie opérative qui permet de recueillir des informations et de les transmettre à la partie commande, il est capable de détecter avec ou sans contact un phénomène physique dans son environnement « présence ou déplacement d'un objet, chaleur, lumière » [1]



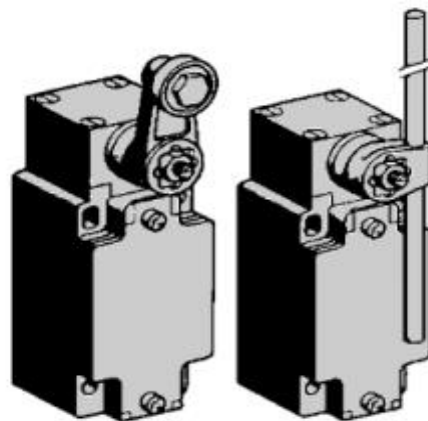


**FIGURE 1.3** : capteurs TOR

### 1.2.3.2 Les différents types de capteurs :

Il existe plusieurs types de capteurs utilisables selon les exigences des systèmes réalisés, et sont comme suit :

**Capteur avec contact « fin de course »** : Les capteurs de position sont des capteurs de contact. Ils peuvent être équipés d'un galet, d'une tige souple, d'une bille. L'information donnée par ce type de capteur est de type **TOR** et peut être électrique ou pneumatique.



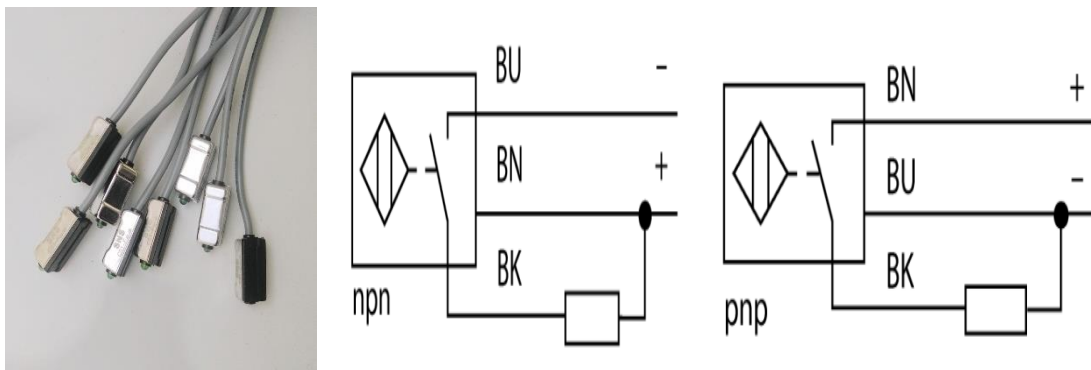
**FIGURE 1.4** : Capteur à levier et à tige

**Capteur de proximité** : Nous observons aussi cinq technologies:

- Les détecteurs capacitifs.
- Les détecteurs inductifs.
- Les détecteurs photo électriques.
- Les détecteurs magnétiques.
- Les détecteurs ultrasons.

Dans ce chapitre nous allons présenter le capteur magnétique car c'est le capteur utilisé dans notre projet.

**Capteurs magnétiques** : Cette technologie est basée sur un principe mécanique détectant un champ magnétique. Ces capteurs intègrent une lame (constituée d'un mélange de métal et de verre) qui présente la caractéristique d'être magnétisée très rapidement en présence d'aimant, et démagnétisée très rapidement en son absence, Servent à détecter la position du piston des vérins pneumatiques. Ils sont fixés directement sur le vérin. L'aimant torique du piston est détecté à travers la paroi non magnétisée du vérin. [7]



**FIGURE 1.5** : Branchement des capteurs de différents fils

### **1.2.3.3 Technologie**

#### **1.2.3.3.1 Types de branchement**

Deux types de technologie (branchement) sont retenus:

- La technique **3 fils** pour les détecteurs alimentés en courant continu, deux fils servent l'alimentation (généralement le bleu la masse et le marron 24V), le troisième (Noire) à la transmission du signal de sortie.
- La technique **4 fils** pour les détecteurs alimentés en courant continu, généralement deux fils pour l'alimentation (Marron + et Bleu -), le troisième fil (Noir) pour le signal NO et le quatrième fil (Blanc) pour le signal NF.

### 1.2.3.3.2 Le détecteur PNP ou NPN

Pour comprendre le branchement, on assimilera ce dernier à un contact électrique.

**Pour le détecteur PNP:** Lorsque qu'il y a détection, le transistor est passant (contact fermé). Il va donc imposer le potentiel « + » sur la sortie **S**. La charge est branchée entre la sortie **S** et le potentiel « - ». Ce type de détecteur est adapté aux unités de traitement qui fonctionnent en logique positive.

**Pour le détecteur NPN:** Lorsque qu'il y a détection, le transistor est passant (contact fermé). Il va donc imposer le potentiel « - » sur la sortie **S**.

La charge est branchée entre la sortie **S** et le potentiel « + ». Ce type de détecteur est adapté aux unités de traitement qui fonctionnent en logique négative.

**Remarque:** Un capteur de type PNP donne un signal plus « + » en sortie par contre un capteur de type NPN donne un signal moins « - ».

## 1.2.4 Actionneur :

### 1.2.4.1 Définition d'un actionneur

Les actionneurs transforment l'énergie qu'ils reçoivent en un autre phénomène physique « un déplacement, un dégagement de chaleur, une émission de lumière ... ». Donc le système automatisé doit être capable de s'exprimer, c'est-à-dire d'émettre des sons ou encore d'effectuer un mouvement. Il existe donc, pour chacune de ces actions, un type d'actionneur [22].

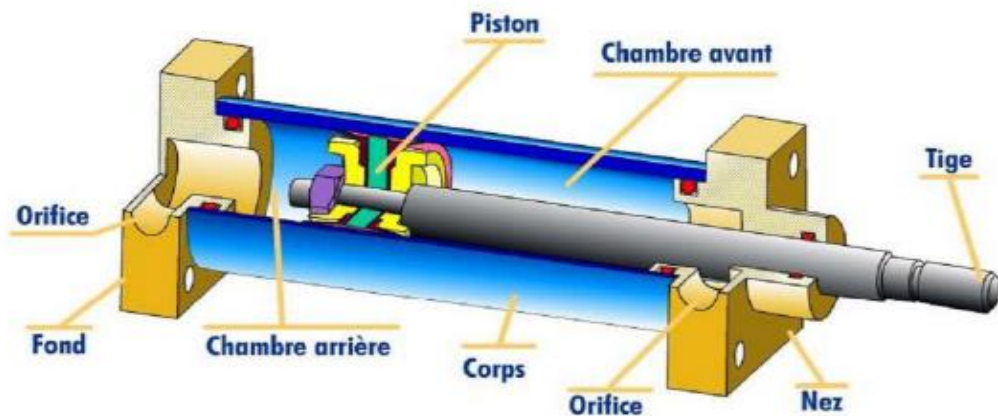


**FIGURE 1.6 :** Des actionneurs TOR

### 1.2.4.2 Types d'actionneurs :

#### 1.2.4.2.1 Les vérins pneumatiques

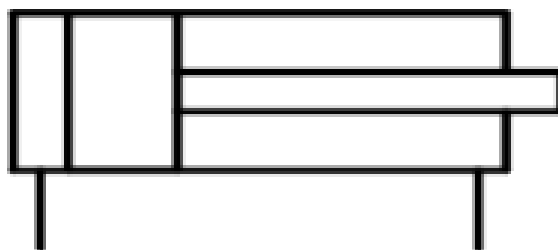
Un vérin pneumatique est un actionneur linéaire dans lequel l'énergie de l'air comprimé est transformée en travail mécanique. Cet actionneur de conception robuste et de simplicité est utilisé dans toutes les industries manufacturières. Il permet de reproduire les actions manuelles d'un opérateur telles que pousser, tirer, plier, serrer, soulever, positionner, etc... [23]



**FIGURE 1.7 :** Vérin pneumatique

#### 1.2.4.2.2 Les Vérins double effet

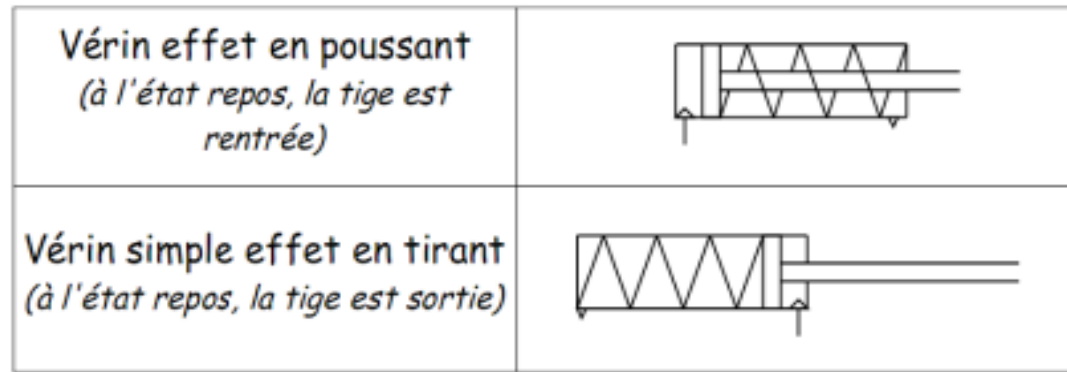
Un vérin double effet a deux directions de travail. Il comporte deux orifices d'alimentation et la pression est appliquée alternativement de chaque côté du piston ce qui entraîne son déplacement dans un sens puis dans l'autre.



**FIGURE 1.8 :** Vérin double effet

#### 1.2.4.2.3 Les Vérins simple effet

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



**FIGURE 1.9 :** Vérin simple effet

#### **1.2.4.2.4 Electrovanes**

Une électrovanne est composée de deux parties :

- Une tête magnétique constituée principalement d'une bobine, tube, culasse, bague de déphasage, ressort(s).
- Un corps, comprenant des orifices de raccordement, obturés par clapet, membrane, piston, etc. selon le type de technologie employée. L'ouverture et la fermeture de l'électrovanne est liée à la position du noyau mobile qui est déplacé sous l'effet du champ magnétique engendré par la mise sous tension de la bobine.

Nous avons deux types d'électrovannes :

**Vanne « TOR » :** La vanne tout ou rien, est soit complètement fermée ou ouverte selon l'information envoyée par l'API.

**Vanne régulatrice :** La commande de la vanne contrôle le débit du fluide qui la traverse.

### **1.3 Conclusion**

Ce chapitre a permis de dresser une description globale de l'ensemble des appareils électriques et des appareils de mesure.

Ce qui nous orienté vers le choix des composants appropriés relatives à notre projet. En effet, le choix du matériel se repose sur un ensemble de critères que nous devrions respecter.

Ceci dans le but d'assurer le bon fonctionnement de la machine.

# Chapitre 2 Les automates programmables

## industriels

---

### **2.1 Introduction**

Un automate programmable industriel (API) et en anglais programmable Logic Controller (PLC) est une carte électronique qui gère un ensemble de composants actionneurs et capteurs grâce à une programmation dans l'automate étant le plus sûr en terme de sécurité humaine en remplaçant ce dernier dans ces actions pénibles, délicates et répétitives.

En effet, l'automate est un produit, facile à programmer, à connecter, adapté aux conditions industrielles.

Dans ce chapitre, nous présenterons les automates programmables industriels, leur architecture, structure et l'ensemble des modules qui le formule tout en abordant l'automate programmable S7-300 étant des machines dotée d'une grande flexibilité.

## **2.2 Automatisation**

### **2.2.1 Historique et objectif de l'automatisation**

Les Automates Programmables Industriels (API) sont apparus aux Etats-Unis vers 1969 où ils répondaient aux désirs des industries de l'automobile de développer des chaînes de fabrication automatisées qui pourraient suivre l'évolution des techniques et des modèles fabriqués. [2]

L'automatisation a pour objectif l'amélioration des conditions de travail, et surtout la sécurité Parmi les buts de l'automatisation il y a :

- La supervision des installations et des machines.
- La simplification du travail de l'humain.
- Elimination les tâches dangereuses et pénibles, en faisant exécuter par la machine les tâches humaines complexes ou indésirables.
- Augmentation de la sécurité
- Économisassions des matières premières et d'énergies.
- Accroissement de la productivité. Le système automatisé permet d'obtenir de façon reproductible la valeur ajoutée.

### **2.2.2 Les automates programmables industriels**

#### **2.2.2.1 *Définition***

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 utilisateurs 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. »

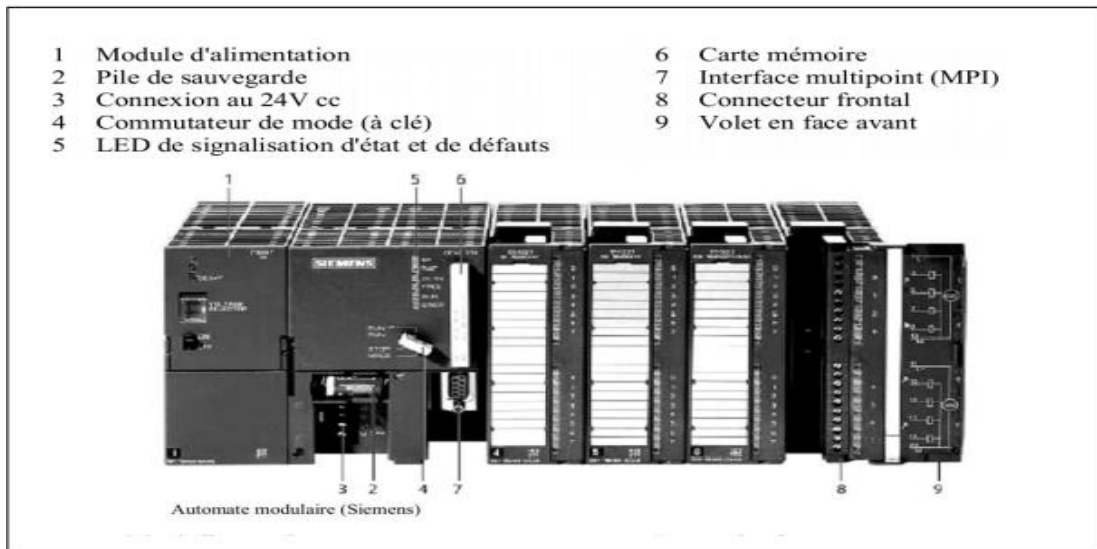


**FIGURE 2.1:** Automates programmable industriels

### **2.2.2.2 Critère de choix de la station de commande**

Le choix de la station de commande repose sur plusieurs critères, on parle sur les principaux de ses critères dans notre cas :[13]

- Disponibilité d'équipement sur le marché avec un cout moins cher
- La méthode la plus simple de diagnostic er la maintenance
- amélioration de la productivité
- augmentation de la qualité du produit
- le nombre d'entrer er de sorties de l'automate
- la nature des entrées/sortie (TOR, analogique, numérique)
- facilité de la programmation
- la nature de traitement (compteur, temporisateur,...)
- type de communication avec d'autres automates
- la disponibilité de l'automate a les magasins de notre pays
- l'habitude de la personne a tous les types d'automates

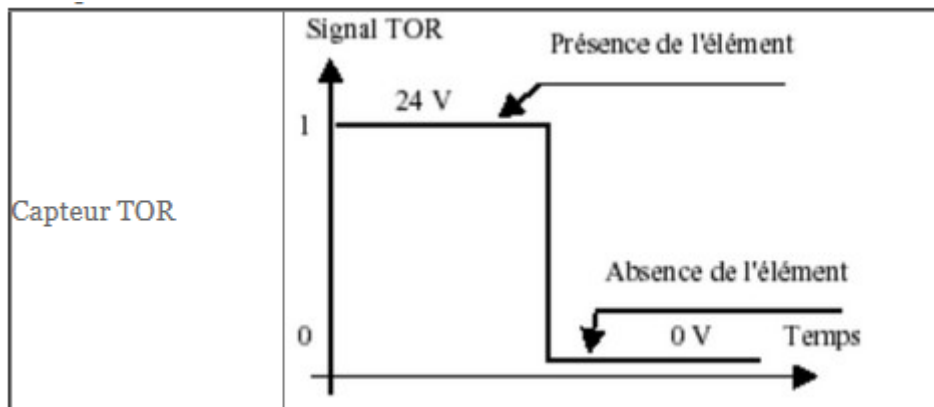


**FIGURE 2.2:** Architecture d'un PLC

### 2.2.2.3 Nature des informations traitées par l'automate :

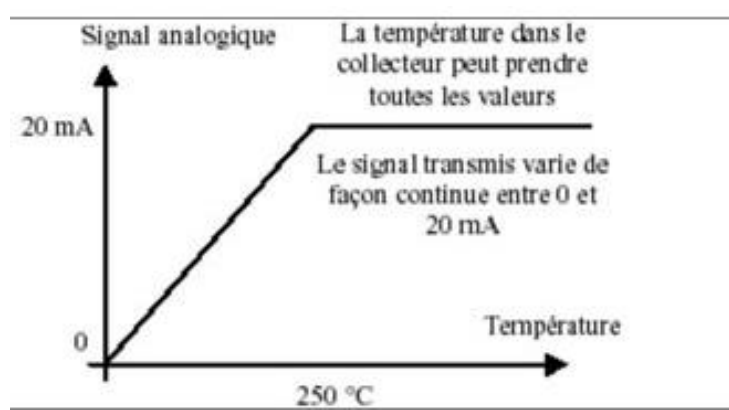
Les informations peuvent être de type [11]

**Tout Ou Rien (TOR) :** L'information ne peut prendre que deux états (vrai/faux, 0 ou 1). C'est le type d'informations délivrées par un détecteur, un bouton poussoir...etc.



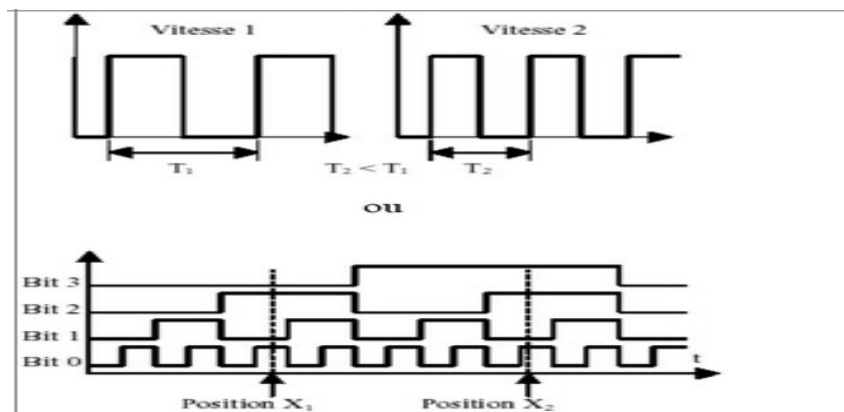
**FIGURE 2.3:** information Toute ou Rien

**Analogique :** L'information est continue et peut prendre une valeur comprise dans une plage bien déterminée. C'est le type d'information délivrée par un capteur (pression, température...etc.)



**FIGURE 2.4:** information analogique

**Numérique :** L'information est contenue dans des mots codés sous forme binaire ou bien hexadécimale. C'est le type d'information délivrée par un ordinateur ou un module intelligent.



**FIGURE 2.5:** information numérique

#### 2.2.2.4 Présentation de l'automate utilisé

L'automate S7-300 est un mini automate modulaire de milieu de gamme de la série S7, fabriqué par la firme SIEMENS. On peut le composer en fonction de nos besoins à partir d'un vaste éventail de module tel que

- Module alimentation PS.
- Module unité centrale CPU.
- Module de signaux d'entrées/sorties SM.
- Module de fonction FM.

- Module Processeurs de communication pour liaison réseau CP Tous les modules du S7-300 sont montrés sous un boîtier. La constitution d'un S7-300 est représentée par la figure suivante [4]



**FIGURE 2.6:** Automate SIEMENS S7-300.

## 2.2.3 Architecture des Automates

### 2.2.3.1 *Structure Externe*

Les automates peuvent être de type compact ou modulaire [3]

#### 2.2.3.1.1 Type Compacte

On distingue les modules de programmation (LOGO de siemens, ZELIO de Schneider, MILENIUM ...) sont des micros automates. Il intègre le processeur, l'alimentation, les entrées et les sorties. Selon les modèles et les fabricants, il pourra réaliser certaines fonctions supplémentaires (comptage rapide, E/S analogique...etc.) et recevoir des extensions en nombre limité. Ces automates, de fonctionnement simple sont généralement destinés à la commande de petits automatismes.

SIEMENS LOGO	CROUZET MILLENIUM	SCHNEIDER ZELIO	SCHNEIDER TWIDO	MOELLER PS4
				

**FIGURE 2.7 :** Des API compacts.

### **2.2.3.1.2 Type modulaire**

Le processeur, l'alimentation et les interfaces d'entrées / sorties résident dans des unités séparées (modules) et sont fixées sur un ou plusieurs racks contenant le "fond de panier" (bus plus connecteurs). Ces automates sont intégrés dans les automatismes complexes où puissance, capacité de traitement et flexibilité sont nécessaires.



**FIGURE 2.8:** Des automates modulaires.

### ***2.2.3.2 Architecture interne d'un API :***

De manière générale, un API est structuré autour de plusieurs élément de base, que sont l'unité de traitement, la mémoire, l'unité d'alimentation, interface d'entrées-sorties, l'interface de communication et le périphérique de programmation (voir figure 1.5).[3] Il est constitué d'une unité centrale de traitement (CPU), qui comprend le micro processeur, la mémoire, les entrées-sorties du système, elle contrôle et exécute toute les opérations de l'API, il est muni d'une horloge dont la fréquence détermine la rapidité de fonctionnement de l'API. Au sein de l'API toutes les informations sont transmises au moyen de signaux numérique, les chemins par lesquels passent ces signaux sont appelé bus.

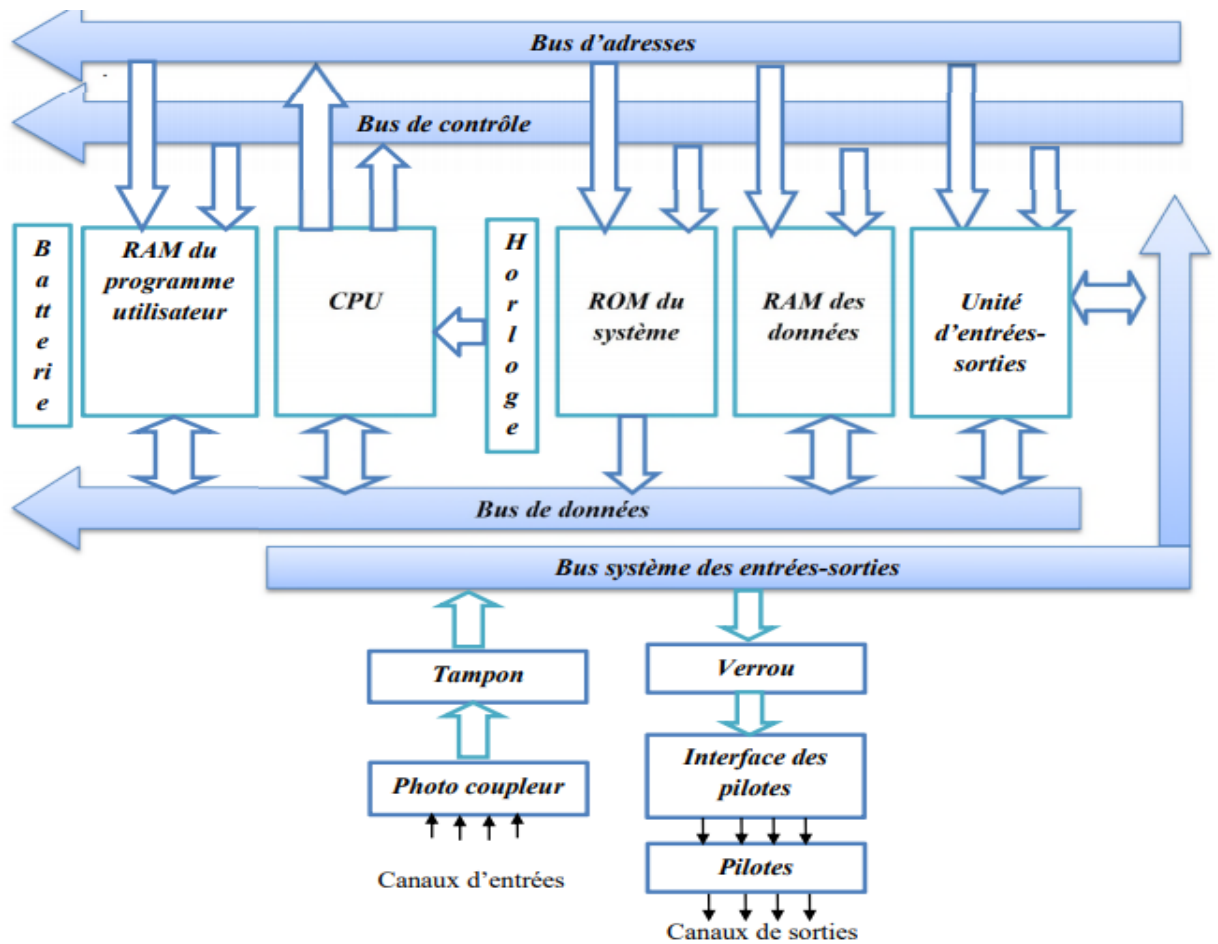


FIGURE 2.9: Architecture d'un API

### 2.2.3.2.1 CPU : (CPU Central Processing Unit)

Elle contient le microprocesseur la CPU interprète les signaux d'entrées et effectues les actions de commande conformément au programme stocker en mémoire en communiquant aux sorties les décisions sous forme de signaux d'actions.



**FIGURE 2.10:** Représentation de la CPU S7-300

### 2.2.3.2.2 Bus :

Les bus représentent les chemins au sein de l'API, les informations sont transmises en binaire sous forme de groupe de bit, et un bit est un chiffre binaire qui vaut 1 ou 0, un mot est un groupe de bits qui constitue une information.

Le système comprend quatre bus :

**Bus de données :** transporte les données utilisées dans le traitement effectué par la CPU, il est bidirectionnel, le nombre de fils de ce bus varie suivant les microprocesseurs.

**Bus d'adresse :** transporte les adresses des emplacements mémoire pour que chaque mot puisse être localisé en mémoire chaque emplacement possède une adresse unique que la CPU utilise pour accéder aux données enregistrées à cet emplacement, que ce soit pour lire ou pour y écrire, c'est le bus d'adresse qui fournit les informations stipulant l'adresse à laquelle la CPU doit accéder. Si le bus est constitué de 8 lignes le nombre de mots de 8 bits et par conséquent le nombre d'adresses distinctes est égale à  $2^8$  c'est-à-dire 256, il est possible d'accéder à 65536 emplacements.

**Bus de contrôle :** transporte les signaux utilisés par la CPU pour le contrôle, il sert à informer les dispositifs mémoires s'ils vont recevoir des données à partir d'une entrée ou s'ils vont envoyer des données et à transmettre les signaux de minutage qui permettent de synchroniser les opérations.

**Bus système** : sert aux communications entre les ports d'entrées-sorties et l'unité d'entrées sorties.

#### 2.2.3.2.3 Les mémoires

Un système à processeur est toujours accompagné d'un ou de plusieurs types de mémoires. Les automates programmables industriels possèdent pour la plupart les mémoires suivantes :

**Mémoire de travail** : La mémoire de travail contient les parties du programme significatif pour son exécution. Le traitement du programme a lieu exclusivement dans la mémoire de travail et dans la mémoire système.

**Mémoire système** : La mémoire système contient les éléments de mémoire que chaque CPU met à la disposition du programme utilisateur comme, par exemple, mémoire image des entrées, mémoire image des sorties, mementos, temporisation et compteurs. La mémoire système contient, en outre, la pile des blocs et la pile des interruptions. Elle fournit aussi la mémoire temporaire allouée au programme (pile des données locales).

**Mémoire de chargement** : La mémoire de chargement sert à l'enregistrement du programme utilisateurs sans affectation de mnémoniques ni de commentaires (ces derniers restent dans la mémoire de la console de programmation). La mémoire de chargement peut être soit une mémoire vive (RAM), soit une mémoire EPROM.

**Mémoire RAM non volatile** : Zone de mémoire configurable pour sauvegarder des données en cas de défaut d'alimentation.

**Mémoire ROM** : sont des mémoires mortes dont l'utilisateur ne peut que lire le contenu, dans le cas des automates programmables, elle contient le système d'exploitation qui gère la CPU.

#### 2.2.3.2.4 Châssis d'extension :

Les châssis sont utilisés pour fixé le CPU et l'alimentations et les autres matériels .





### 2.2.3.2.5.1 Modules d'entrées et sorties TOR (Tout Ou Rien) :

**Modules d'entrées TOR (Tout Ou Rien) :** L'automate reçoit ses informations sur le processus via les capteurs de signaux reliés aux entrées. Les modules d'entrée TOR permettent de recevoir les signaux des différents capteurs logiques qui peuvent être des détecteurs qui reconnaîtront si la pièce d'usinage se trouve à une position donnée (détecteurs des niveaux -haut et bas-, Cellules photoélectrique . . . ) ou de simples commutateur ou interrupteur qui peuvent être fermés ou ouverts. Ce qui fait que l'information délivrée par ces capteurs et qui sera traitée par la CPU ne peut prendre que deux valeurs 0 ou 1.



**FIGURE 2.13 :** Module entrée TOR

**Modules de sorties TOR (Tout Ou Rien) :** Ces modules permettent de délivrer des signaux qui permettent à l'automate d'agir sur les pré-actionneurs du système à commander tels que (Vanne Electromagnétique, Electrovanes, Contacteurs, Pompes, Voyants. . . ).



**FIGURE 2.14:** Module sortie TOR

### 2.2.3.2.5.2 Modules d'entrées et de sorties Analogiques :

**Modules d'entrées Analogiques :** L'information traitée est continue et prend une valeur qui évolue dans une plage bien déterminée. C'est le type d'information délivrée par un capteur (débit, niveau, pression, interface, température. . . ).



**FIGURE 2.15 :** Module entrée analogique

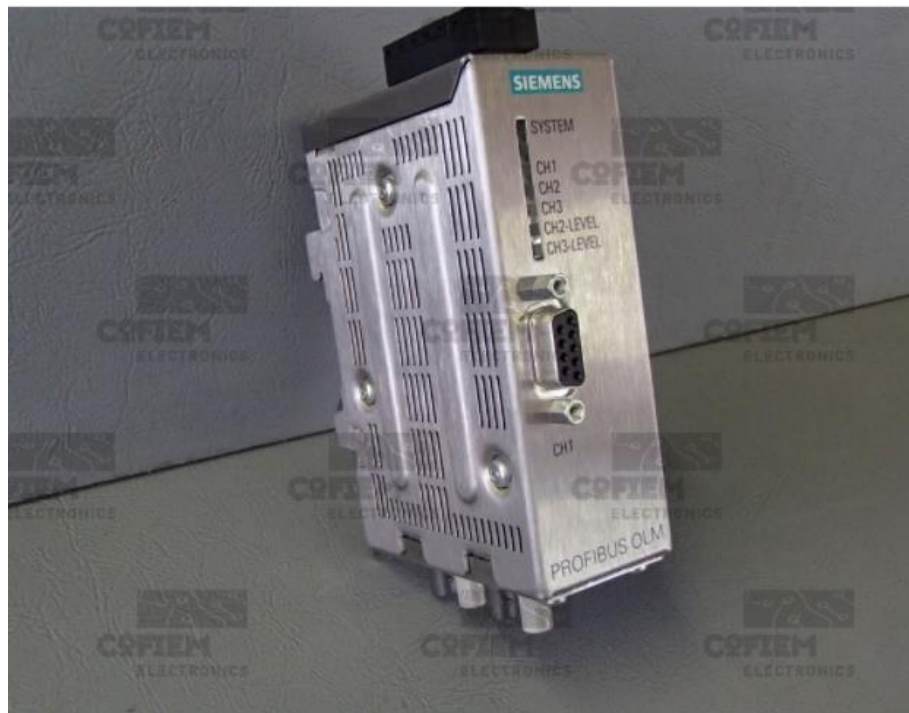
**Modules de sorties Analogiques :** Ils émettent un signal analogique qui représente l'état que doit prendre un actionneur entre deux limites. Ce module est muni d'un convertisseur analogique- numérique [11].



**FIGURE 2.16 :** Module sortie analogique

### 2.2.3.2.5.3 Module de coupleur :

Les modules de couplages sont des cartes électroniques utilisées pour assurer la communication entre l'unité centrale CPU et les périphériques de l'automate (entrées/sorties), console de programmation, etc. Les coupleurs (IM360, IM361, ou IM365) permettent de réaliser la configuration à plusieurs châssis. Ils occupent l'emplacement N°3 dans l'API et ce dernier reste vide au cas on n'utilise pas les coupleurs (IM).



**FIGURE 2.17:** Module de coupleur

### 2.2.3.2.5.4 Module (FM) :

Ce module réduit la charge de traitement de la CPU, en trouve dedans des fonctions spéciales de la régulation PID, des compteurs, et temporisateurs.



**FIGURE 2.18:** Module FM

## **2.2.4 Processeur de communication (CP):**

L'automate S7-300 dispose de différentes interfaces de communication [24] :

- Modules de communication pour la connexion aux systèmes de bus AS-Interface, PROFIBUS et PROFINET /Industriel Ethernet.
- Module de communication pour les liaisons point -à -point.
- Interface multipoint MPI, intégrée à la CPU : la solution économique pour le branchement des systèmes d'automatisation SIMATIC S7/C7.

### **2.2.4.1 Signification des états de l'automate**

La CPU comporte des LED de signalisation suivante [10]

**SF (rouge)** signalisation groupée de défauts, elle s'allume si on a défaut matériels et en cas d'erreurs de programmation, de paramétrage, de calcul...etc.

**BATF (rouge)** Défaut de la pile, elle s'allume si elle est défectueuse, absente ou déchargée. DC5V(vert) Alimentation 5Vcc pour la CPU et le bus S7-300, elle s'allume si les 5V sont présentes et elle clignote s'il ya surcharge de courant.

**FRCE (jaune)** elle s'allume en cas de forçage permanent.

**RUN (vert)** Etat de fonctionnement RUN, elle clignote en cas de démarrage de la CPU.

**STOP (jaune)** : Etat de fonctionnement STOP, elle s'allume si la CPU ne traite aucun programme utilisateur et elle clignote en cas ou la CPU demande un effacement général. Les LED de signalisation de défaut SF-DP et BUSF ne se rencontrent que dans le cas de la CPU31 relative à la configuration maitre-esclave du S7-300. [17]

#### **2.2.4.2 Commutateur de mode de l'automate**

**RUN** : La CPU traite le programme de l'utilisateur. Dans cette position, la clé peut être retirée pour éviter qu'une personne non habilitée change le mode de fonctionnement.

**RUN-P** La CPU traite le programme de l'utilisateur et la clé ne peut être retirée, il est possible de lire le programme de la CPU avec une PG (CPU vers PG) et de transférer des programmes dans la CPU (PG vers CPU).

**STOP** La CPU ne traite aucun programme utilisateur, la clé peut être retirée pour éviter le changement de mode inattendue mais on peut lire et écrire dan la CPU.

**MRES** Cette position instable du commutateur de mode de fonctionnement, en vue de l'effacement générale de la CPU. Le contenu de la mémoire de chargement rémanente intégrée reste inchangé après un effacement général.

#### **2.2.4.3 Interface MPI (interface multipoint)**

L'interface MPI est l'interface de la CPU utilisée pour la console de programmation(PG), le pupitre operateur(OP) ou par la communication au sein d'un réseau MPI. La vitesse de transmission typique est de 187,5 k Bauds

## **2.3 Conclusion**

Le développement scientifique a laissé sa trace sur les systèmes de production donnant naissance aux systèmes automatisés de production et aux automates programmables.

Chaque processus industriel de fabrication ou de transformation se compose d'un ensemble de machines destinées à réaliser la fabrication ou la transformation considérée et aujourd'hui l'application des automates programmable est palpable dans l'industrie vue sa grande flexibilité et son aptitude à s'adapter.

Ce chapitre nous a permis de conclure que chaque machine a une partie opérative qui contient des vannes, des moteurs et des capteurs. Ces dispositifs sont gérés essentiellement, par la partie de commande. Autrement dit, Il s'agit de l'automate ou de l'homme machine interface HMI.

## 3 Présentation de la machine et Step7

---

### 3.1 Introduction

Un mélangeur industriel dans le milieu industriel est un appareil qui permet mélanger ou homogénéiser un ou plusieurs composants, dont au moins un composant solide ou liquide. Cet appareil est utilisé essentiellement dans les industries cosmétique, chimique, pharmaceutique et agroalimentaire.

Dans ce chapitre, nous présenterons toutes parties composant la machine du mélangeur, lié un automate programmable que nous utiliserons dans le contrôle et l'automatisation cyclique.



## 3.2 La machine KARL SCHNELL GMBH & CO

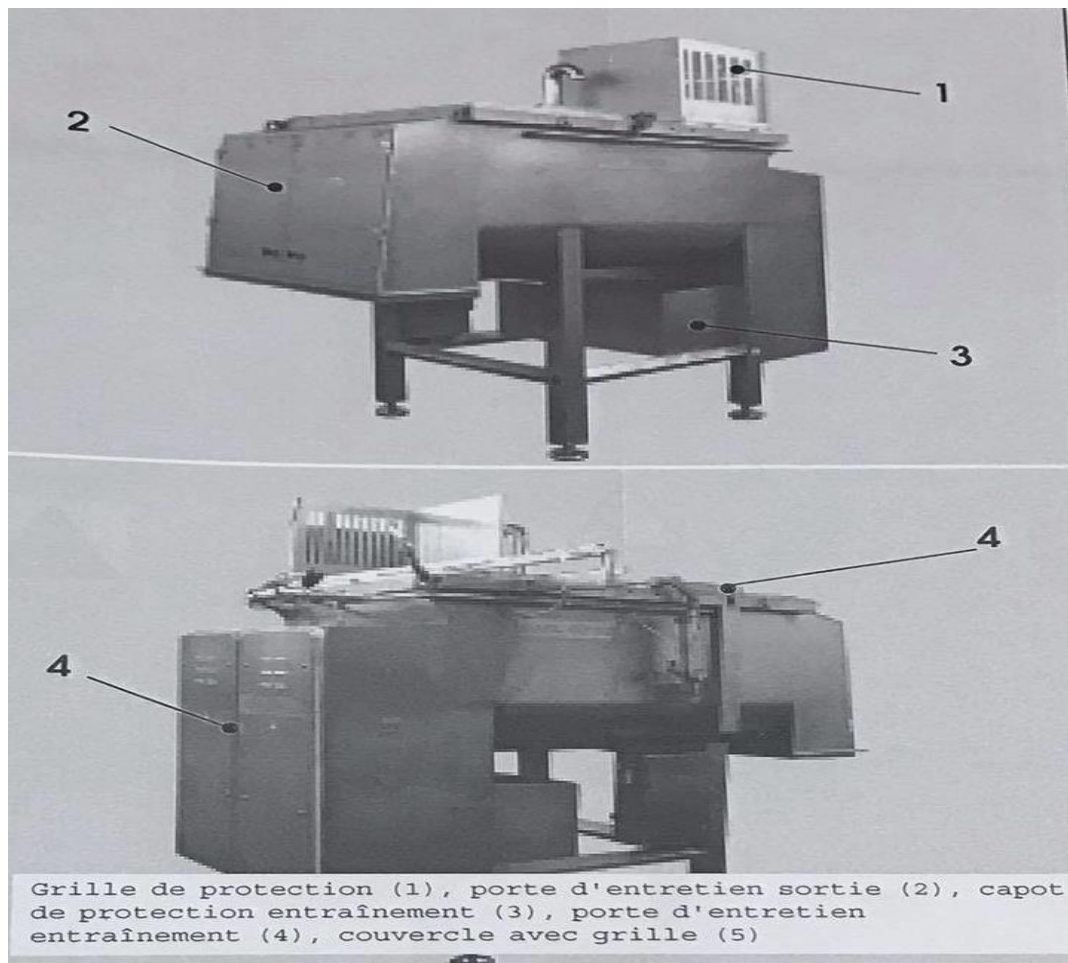
### 3.2.1 Présentation de la machine

Un mélangeur permet d'obtenir un produit final à partir du mélangeur de plusieurs ingrédients.

Un mélangeur est principalement utilisé par les industries notamment dans le domaine agroalimentaire, pharmaceutique, nucléaire, cosmétique ou encore chimique.

Le mélangeur contient plusieurs parties nécessaires qu'on peut définir comme ceci :

- Grille de protection
- Porte d'entretien sortie
- Capot de protection entraînement
- Porte d'entretien entraînement

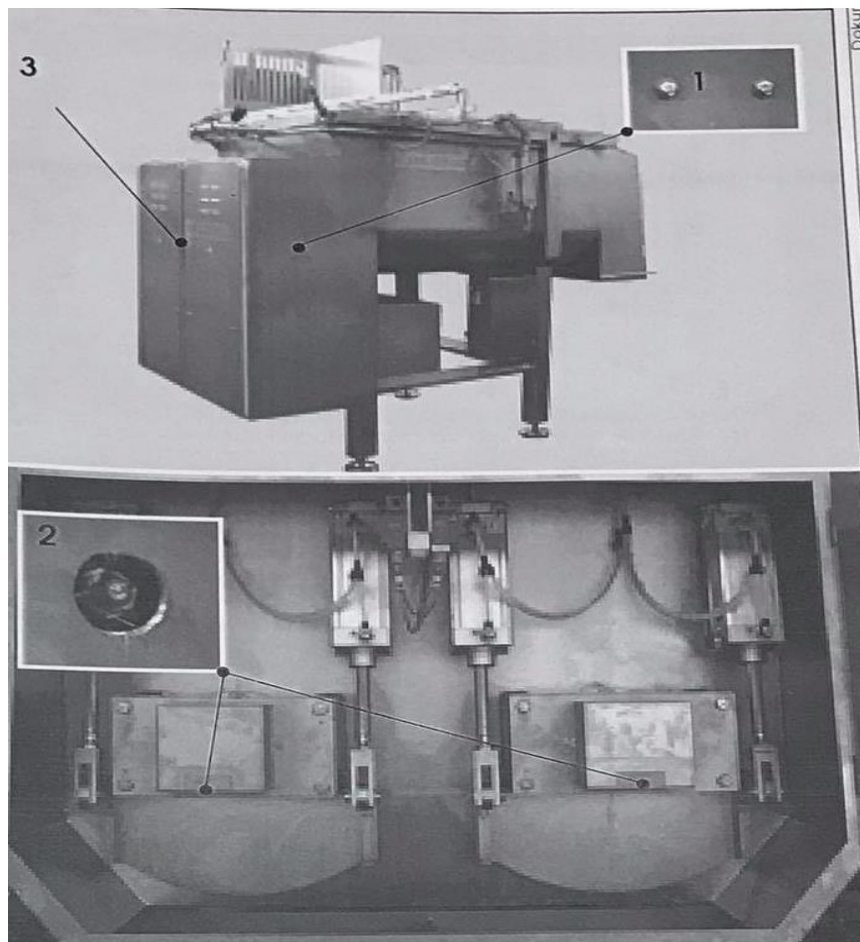


**FIGURE 3.1:** Les parties de mélangeur

### 3.2.2 Description de chaque partie

#### 3.2.2.1 Une partie de sortie de mélangeur :

cette partie contient 4 vérin qu'il sont collé avec deux trappes, quand les vérins soient leur l'axe est sortie donc les trappe sont fermé alors le produit ne peut pas sortir a la deuxième cuve de stockage et quand les vérins soient ouvert le produit qui est dans la cuve de mélangeur ou ce trouve les vices il peut sortir a la deuxième cuve de stockage.



**FIGURE 3.2:**Sortie de mélangeur

#### 3.2.2.1.1 Les vices de mélangeur :

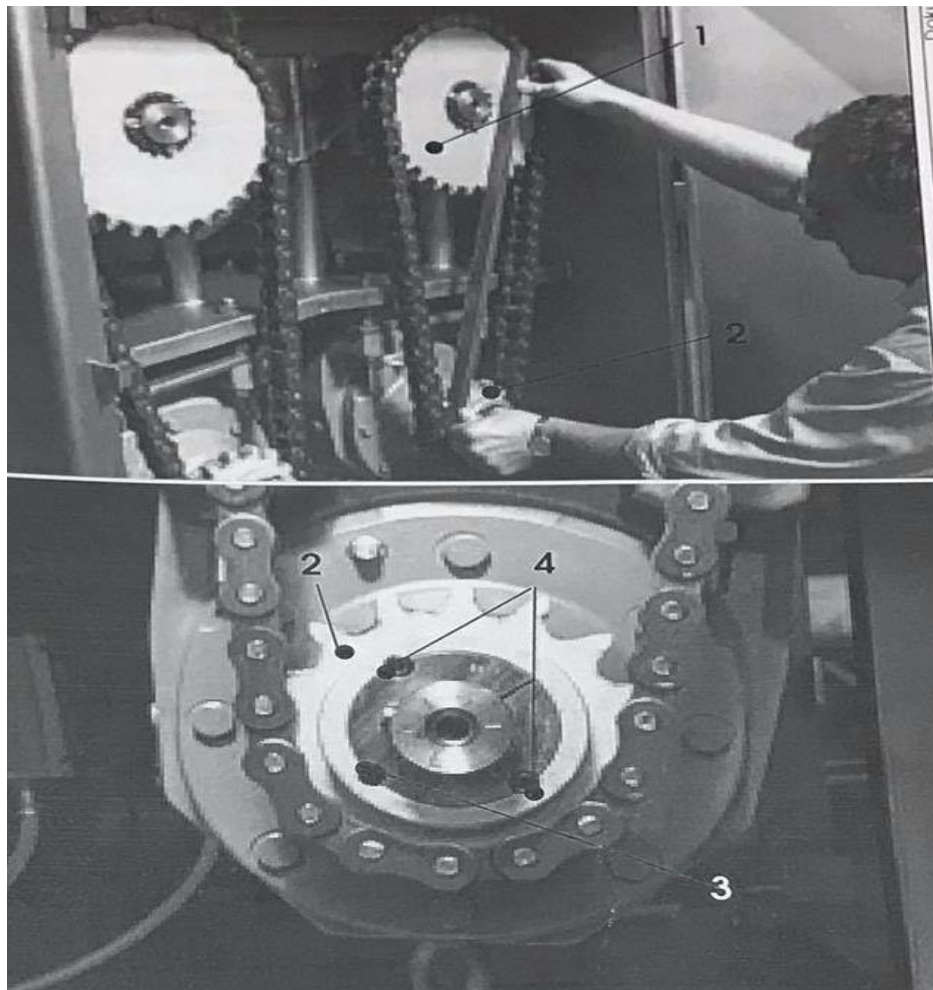
Les mélangeuses sont équipées d'une trémie qui est le lieu du mélange de produit grâce à une ou plusieurs vis verticales ou horizontales. Il existe plusieurs types de mélangeuses distributrices . A notre cas on a deux vices qui tournent chacun leur sens de rotation.



**FIGURE 3.3:** Vis de mélangeur

#### **3.2.2.1.2 Les pinions :**

Pinions roue dentée faisant partie d'un système à chaîne (inventée en 1880) ou d'un couple d'engrenage dans un système de transmission ou un réducteur. Le pinions est la roue dentée qui a le plus petit nombre de dents. A notre cas on a un moteur qui est collé avec un petit pignon il s'appelle un coupleur quand il marche il tourne avec lui le grand pignon de mélangeur grâce a une chaine.



**FIGURE 3.4** : Pinions de mélangeur

### **3.2.2.1.3 Trappe de couvercle :**

C'est une porte qui s'ouvre et fermer avec un vérin pour qu'il passe le produit dans la cuve de mélangeur.



**FIGURE 3.5 :** Trappe de couvercle

### **3.2.2.2 Les composants utilisés :**

#### **3.2.2.2.1 Le disjoncteur**

Un disjoncteur est un appareil mécanique de connexion capable d'établir, de supporter et d'interrompre un courant dans un circuit électrique. Un disjoncteur protège l'installation contre :

- Les surcharges (action du déclencheur thermique).
- Contre les courts-circuits (action de déclencheur magnétique)
- Pour remettre l'installation sous tension, il suffit de remettre l'interrupteur en position " on "



**FIGURE 3.6 :** Disjoncteur

### 3.2.2.2 Le sectionneur

C'est un appareil mécanique de connexion ; capable d'ouvrir et fermer un circuit lorsque le courant est nul afin d'isoler la partie de l'installation en aval du sectionneur.

- Il n'a pas de pouvoir de coupure ou de fermeture.
- Ce n'est pas un appareil de protection.



FIGURE 3.7 : sectionneur

### 3.2.2.3 Les fusibles

Le principe de la protection par fusible repose sur la fusion contrôlée d'un élément fusible, fusion qui intervient après un temps donné pour un courant donné.



FIGURE 3.8: Fusibles

#### **3.2.2.2.4 Sonde de température**

La sonde Pt 100 est un capteur de température qui est utilisé dans le domaine industriel (agroalimentaire, chimie, raffinerie...). Ce capteur est constitué d'une résistance en Platine. La valeur initiale du Pt100 est de 100 ohms correspondant à une température de 0°C.[41]



**FIGURE 3.9** :Sonde de température

#### **3.2.2.2.5 Les variateurs de vitesse**

Un variateur de vitesse est un équipement permettant de faire varier (commander) la vitesse d'un moteur, une nécessité pour de nombreux procédés industriels.

Durant ces dernières années, les entraînements à vitesse variable trouvent de plus en plus de larges domaines d'application dans les équipements industriels, comme les papeteries, les laminoirs, les pompes ou les machines-outils. Parmi toutes les technologies existantes pour varier la vitesse des moteurs électriques, les variateurs de vitesse, dont le rôle consiste à moduler l'énergie électrique fournie au moteur. En effet, grâce à ces variateurs, il est possible de contrôler parfaitement les phases de mise en rotation et d'arrêt de l'application, mais aussi d'effectuer un contrôle indépendant de la vitesse et du couple. De plus, les variateurs de vitesse assurent toutes les fonctionnalités de protection du variateur et du moteur. Un variateur de vitesse est donc un dispositif permettant de réaliser l'alimentation et la commande d'un moteur. [45]



**FIGURE 3.10 :**Variateur de vitesse

### 3.2.2.2.6 Vanne

Pour ce type de vanne, la section de passage du fluide est égale à 0% ou 100% de la section de passage à pleine ouverture, selon la forme de siège de la vanne et la disposition du clapet.

La vanne automatique comprend trois parties bien distinctes :

**Le servomoteur** : c'est l'organe qui, recevant le signal du régulateur ou de la commande à distance manuelle, le convertit en une force de valeur qui peut déplacer des charges d'une position définie à une autre position.

**Le corps** : c'est l'élément de la vanne automatique à l'intérieur duquel le fluide peut être arrêté ou, au contraire, circuler plus ou moins librement.

**Le positionneur** : c'est un relais intercalé dans le circuit d'air modulé du régulateur. Son rôle est de réaliser un positionnement du clapet correspondant exactement au signal émis par le régulateur, un positionnement rapide et fidèle. [14] [15]



**FIGURE 3.11:**Vanne



### 3.2.2.2.7 Capteur analogique :

La sortie du capteur analogique peut prendre n'importe quelle valeur entre deux extrêmes. A chaque valeur de la grandeur physique doit correspondre une valeur en sortie. Une variation de la grandeur physique d'entrée du capteur produit une variation de la caractéristique électrique du capteur (courant, tension, fréquence, valeur moyenne, rapport cyclique,).

Comme beaucoup de composants ou ensembles électroniques, il existe un certain nombre de paramètres électriques communs à tous les capteurs, ils caractérisent les performances et donc leurs utilisations dans un environnement donné [9].

**Précision:** La précision d'un capteur est caractérisée par l'incertitude absolue obtenue sur la grandeur électrique obtenue en sortie du capteur. Elle s'exprime en fraction de la grandeur physique mesurée.

**Sensibilité :** Ce paramètre caractérise l'aptitude du capteur à détecter la plus petite variation de la grandeur à mesurer.

**Étendue de mesure:** cette caractéristique donne la plage de fonctionnement du capteur pour la grandeur à mesurer. Elle est souvent notée E.M.

**Linéarité :** Un capteur est dit linéaire s'il présente la même sensibilité sur toute l'étendue de sa plage d'emploi.

**Fidélité :** Un capteur est dit fidèle si le signal qu'il délivre ne varie pas pour une série de mesures concernant la même valeur de la grandeur d'entrée

### 3.2.2.2.8 Convoyeur

Un convoyeur est un mécanisme ou machine qui permet le transport d'une charge isolée (cartons, bacs, sacs, ...) ou de produit en vrac (terre, poudre, aliments, sucre...) d'un point A à un point B.

Dans notre cas, le convoyeur alimenté par un moteur asynchrone, qui sert à transporter le produit pour qu'il soit mis dans les trémies de mélangeur.

### 3.2.2.2.9 Les moteurs asynchrones

Ces moteurs sont robustes, ils sont intéressants, lorsque la vitesse du dispositif à entraîner n'a pas à être rigoureusement constante. La vitesse de la rotation du moteur n'est pas la même que la vitesse du champ électrique créé (champ statorique) les deux vitesses sont asynchrone.



**FIGURE 3.12 :** Moteur

### 3.2.2.2.10 Débitmètre

Le débit d'un fluide est la quantité du fluide traversant la section transversale d'une conduite par unité de temps. Cette quantité s'exprime, soit en débit masse, soit en débit volume. Le débit est l'une des grandeurs de système les plus fréquemment mesurées dans l'industrie. L'eau, le gaz naturel, la vapeur, le pétrole, les produits chimiques, les eaux usées ne sont que quelques-unes des produits mesurés quotidiennement. Un débitmètre est un appareil destiné à mesurer le débit massique ou volumique d'un fluide liquide ou gazeux. Sécurité de l'installation, qualité constante des produits, optimisation des procédés, protection de l'environnement, voilà quelques aspects qui confirment que la mesure de débit occupe une place prépondérante dans l'industrie. [14]



**FIGURE 3.13:** Débitmètre

### 3.2.2.2.11 Contacteurs

Le contacteur permet "d'établir ou interrompre" un courant nominal important avec une faible énergie de commande (commutation). Il possède un fort "pouvoir de coupure", il peut donc être actionné en charge. C'est à dire lorsqu'un courant le traverse. [16]

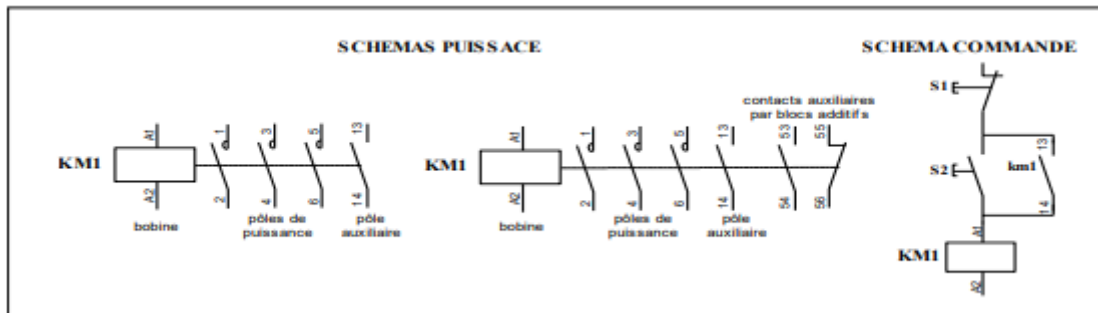


FIGURE 3.14 : Contacteur

### 3.2.2.2.12 Relais thermique [16]

Le relais thermique est un appareil qui protège le récepteur contre les surcharges et les coupures de phase. Pour cela, il surveille en permanence le courant dans le récepteur. En cas de surcharge, le relais thermique n'agit pas directement sur le circuit de puissance. Un contact du relais thermique ouvre le circuit de commande d'un contacteur et par conséquent ce dernier coupe le courant dans le récepteur



FIGURE 3.15: Relais thermique

### 3.2.2.2.13 Régulateur de pression

Le régulateur a pour fonction de ramener la pression primaire, qui est variable, à une pression secondaire pratiquement constante. Par ailleurs, une pression trop élevée augmente également la consommation d'air et donc le coût d'énergie. Le régulateur de pression est donc essentiel sur la majorité des systèmes, car il permet de réaliser

des économies sur les coûts d'exploitation et d'accroître la durées de vie de l'équipement [25].



**FIGURE 3.16:** Régulateur de pression

### **3.2.3 L'automate SIEMENS S7-300 :**

#### **3.2.3.1 Présentation de l'Automate**

L'API S7-300 de gamme SIEMENS, est un Mini-automate modulaire pour les applications de milieu de gamme, avec possibilité d'extension jusqu'à 32 modules d'entrées/sorties, et une mise en réseau par l'interface multipoint (MPI), PROFIBUS et industriel Ethernet. [11]

Les caractéristiques principales des automates programmables industriels (API) Siemens S7- 300 sont les suivantes : Il s'agit d'un matériel multiprocesseur :

- un processeur logique (bit processor).
- un processeur pour les opérations arithmétiques (Word processor).
- un processeur dédié à la régulation de type PID.
- un processeur dédié à la gestion des communications. Le logiciel Siemens S7 permet une programmation multi langage, c'est-à-dire qu'il peut être programmé dans plusieurs langages différents, qui peuvent être même mélangés dans un même programme (mais pas dans une même sous-routine) :
- Liste d'instructions ou Instruction List (IL).
- Langage à contacts ou Ladder diagramme (CONT).

- Logigramme ou Fonctionnel Block (LOG). Le mode séquentiel est accessible :
- soit en utilisation une programmation en GRAFCET directement
- soit en créant une séquence d'exécution. Blocs d'organisation

### 3.2.3.2 Logiciel de programmation STEP 7 [17]

STEP 7 est le logiciel de base pour la configuration et la programmation des systèmes SIMATIC par exemple (S7-300).

Il fait partie de l'industrie logicielle SIMATIC. Les tâches de base qu'il offre à son utilisateur lors de la création d'une solution d'automatisation sont :

- La création et gestion de projet ;
- La configuration et le paramétrage du matériel et de la communication ;
- La gestion des mnémoniques ;
- La création des programmes ;
- Le test de l'installation d'automatisation. Il s'exécute sous les systèmes d'exploitation de Microsoft à partir de la version Windows 95. Par conséquent, il s'adapte à l'organisation graphique orientée objet qu'offrent ces systèmes d'exploitation.

### 3.2.3.3 Création d'un projet STEP 7

Le logiciel SIMATIC Manager étant maintenant ouvert puis assistant nouveau projet, après la sélection du type de la CPU et l'insertion du bloc d'organisation et le langage à liste,

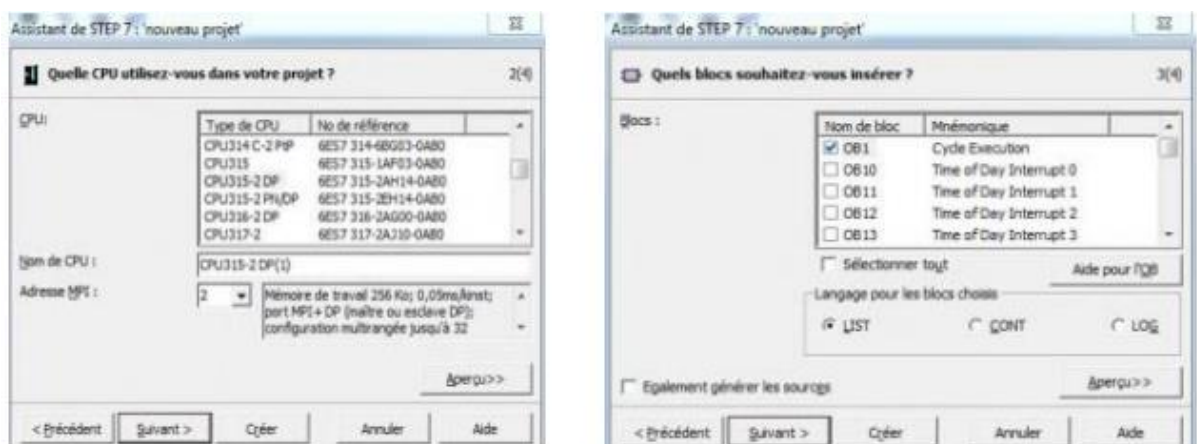
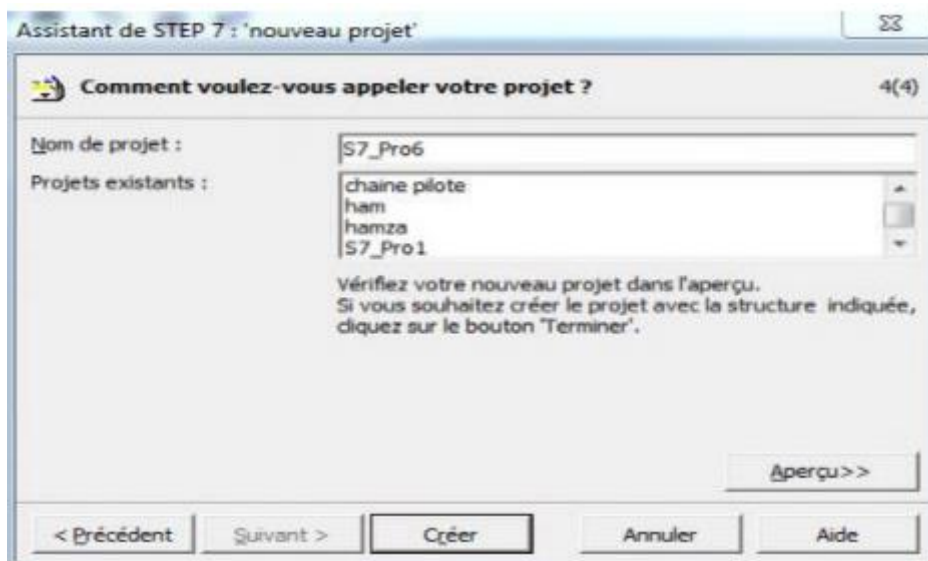


FIGURE 3.17: Choix de la CPU

Une fenêtre s'ouvre pour donner un nom de projet et on clique sur « Créer » pour finaliser la création de notre projet.



**FIGURE 3.18 :** Nom et création de projet

#### **3.2.3.4 Configuration matérielle**

C'est une étape importante, qui correspond à l'agencement des châssis, et des modules. Ces derniers sont fournis avec des paramètres définis par défaut en usine. Une configuration matérielle est nécessaire pour :

Les paramètres ou les adresses préregler d'un module ;

#### **3.2.3.5 Configurer les liaisons de communication.**

Le choix du matériel SIMATIC S7-300 avec une CPU315-nous conduit à introduire la hiérarchie suivante :

On commence par le choix du châssis selon la station choisie auparavant, pour la station SIMATIC S7-315, on aura le châssis « RACK-300 » qui comprend un rail profilé.

Sur ce profil, l'alimentation préalablement sélectionnée se trouve dans l'emplacement n°1. Parmi celles proposées notre choix s'est porté sur la «ps-307 5A».

- La «CPU315-2PN/DP» est impérativement mise à l'emplacement n°2.
- L'emplacement n°3 est réservé comme adresse logique pour un coupleur dans une configuration multi-châssis.

- Les emplacements n°4, 5, et 6 sont réservés pour les modules d’entrées TOR à 16 bits chacun.
- Les emplacements n°7, 8, et 9 sont réservés pour les modules de sorties TOR à 16 bits chacun.
- La CPU315-2PN/DP comporte un module des entrées analogiques intégré. L’entrée analogique d’adresse DW128 est réservée pour la consigne de vitesse

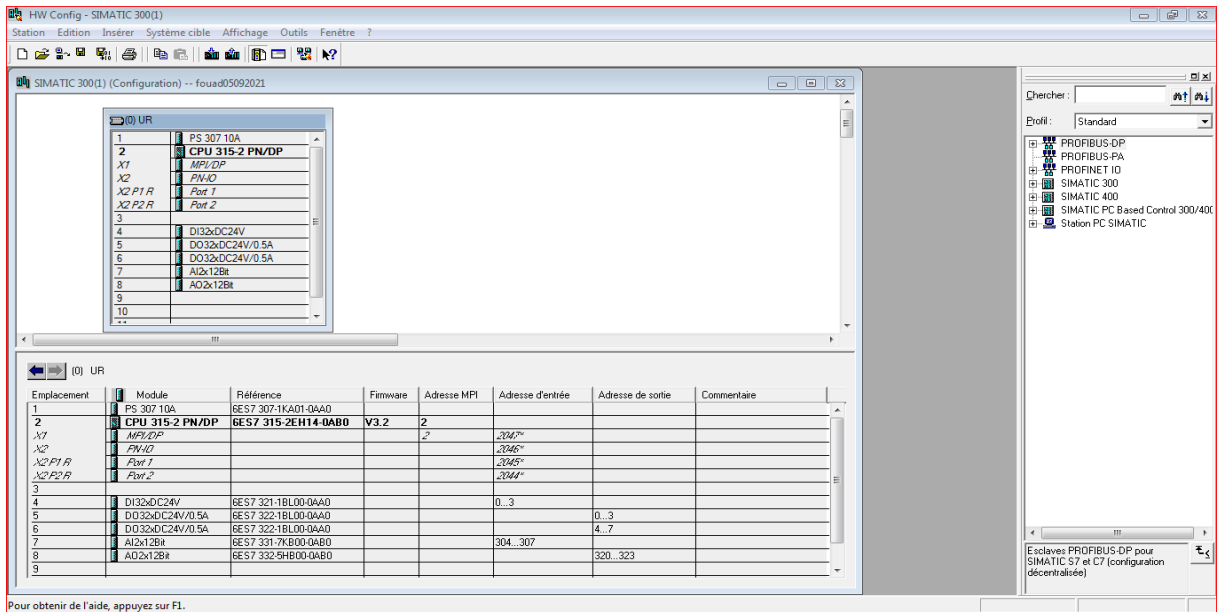


FIGURE 3.19: Configuration matériels

### 3.2.3.6 Structure du programme STEP 7

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

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

Bloc d'organisation : Les blocs d'organisation (OB) constituent l'interface entre le système OB d'exploitation et le programme utilisateur. L'ensemble du programme peut être concaténé dans un seul bloc OB1 (programme linéaire) appelé de manière cyclique par le système d'exploitation ou être structuré dans plusieurs blocs (programme structuré).

**Fonction : Une fonction (FC)** assure une fonctionnalité spécifique du programme. FC, SFC Les fonctions peuvent être paramétrables. Dans ce cas, des paramètres sont transmis à la fonction lorsqu'elle est appelée. Les fonctions conviennent donc pour la programmation de fonctionnalités récurrentes et complexes, par exemple pour effectuer des calculs. Les fonctions système (SFC) sont des fonctions paramétrables, intégrées au système d'exploitation de la CPU, dont le numéro et la fonctionnalité sont définis de manière fixe. Pour de plus amples informations, se reporter à l'aide en ligne.

**Bloc fonctionnel** : Du point de vue du programme, les blocs fonctionnels s'apparentent aux FB, SFB fonctions, mais ils disposent en plus de zones mémoire spécifiques, sous forme de blocs de données d'instance. Les blocs fonctionnels conviennent donc pour la programmation de fonctionnalités récurrentes encore plus complexes, par exemple pour assurer des tâches de régulation.

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

**Blocs de données** : Les blocs de données (DB) sont des zones de données du programme DB utilisateur.

### **3.2.3.7 Les opérandes et types de données autorisés dans la table des mnémoniques : [12]**

Saisir les mnémoniques est très utile, il vaut mieux saisir un programme entièrement en symbole qu'en adressage absolu, c'est beaucoup plus lisible et compréhensible. Il suffit d'aller dans la table des mnémoniques et y entrer les différents éléments, le nom du symbole, son adresse réel, son type et son commentaire. Les paramètres peuvent être des types simples tel que : BOOL, WORD, INT, REAL, ...etc., ou de types complexes tel que : TIMER, COUNTER, etc.



Les différents types de variables sont donnés dans le tableau qui suit, dans le cadre de cette première approche, on fera plus particulièrement attention aux variables suivantes :

<i>Opérandes</i>	<i>Désignation</i>	<i>Type de données</i>	<i>Plage d'adresse</i>
<i>E</i>	<i>Bit d'entrée</i>	<i>BOOL</i>	<i>0.0 → 127.7</i>
<i>A</i>	<i>Bit de sortie</i>	<i>BOOL</i>	<i>0.0 → 127.7</i>
<i>M</i>	<i>Bit de memento</i>	<i>BOOL</i>	<i>0.0 → 255.7</i>
<i>MB</i>	<i>Octet de memento</i>	<i>BYTE, CHAR (8bit)</i>	<i>0 → 255</i>
<i>MW</i>	<i>Mot de memento</i>	<i>WORD, INTEGER (16bit)</i>	<i>0 → 254</i>
<i>MD</i>	<i>Double mot de memento</i>	<i>WORD, REAL (32bit)</i>	<i>0 → 254</i>
<i>T</i>	<i>Temporisation</i>	<i>TIMER</i>	<i>0 → 127</i>
<i>Z</i>	<i>Compteur</i>	<i>COUNTER</i>	<i>0 → 127</i>
<i>VAT</i>	<i>Table de variable</i>	<i>---</i>	<i>0 → 127</i>

**TABLEAU 3.1** : Les différents types de variable

### 3.2.4 Les langages de programmation d'un API

Les langages destinés à la programmation des automates programmables industriels ont pour objectifs d'être facilement mis en œuvre par tout technicien après une courte formation. L'écriture d'un programme consiste à créer une liste d'instructions permettant l'exécution des opérations nécessaires au fonctionnement du système. [50] Actuellement les API disposent en tout ou partie des langages de programmation suivants :

**Langage à Contact LD (Ladder Diagram)** Le langage LD est le plus utilisé, il ressemble aux schémas électriques à contacts, c'est un langage graphique avec une approche aisée et visuelle du problème dédié à la programmation d'équations booléennes (true/false). [18][19]

**Langage FBD (Function Bloc Diagram)** C'est un langage qui permet de programmer graphiquement à l'aide de blocs programmés ou programmables, reliés entre eux et réalisant tout type de fonctions. [18][19]

**Langage List IL (Instruction List)** Le langage List est très proche de l'assembleur, il utilise la totalité des fonctions de l'API. Tous les réseaux de programmation, définis dans un autre langage, sont compilés dans ce langage. [18][19]

**Langage littéral structuré ST (Structure Texte)** Appelé aussi SCF (Structure Control Language), ressemble beaucoup au langage C. C'est un langage structuré qui convient bien pour les applications faisant appel à des calculs compliqués et au traitement des chaînes de caractère, Il permet la programmation de tout type d'algorithme plus ou moins complexe. [81][19]

**Langage GRAFCET** Graphe Fonctionnel de Commande des Etapes et Transitions ou Séquentiel Fonction Charte (SFC), est un outil graphique qui décrit les différents comportements de l'évolution d'un automatisme. C'est un mode de représentation et d'analyse d'un automatisme, particulièrement bien adapté aux systèmes à évolution séquentielle, c'est à dire décomposable en étapes. [19][14]

### **3.3 Conclusion**

Ce chapitre nous a permis de dresser une description complète de la machine du mélangeur ainsi que ses parties composantes, sachant que cette machine fonctionne de manière automatique.

Nous avons également défini le logiciel STEP 7 que nous allons utiliser dans la programmation de la machine, tout en expliquant les différents les différents langages de programmation.

## 4 Simulation et programmation

---

### 4.1 Introduction

Dans le cadre de la programmation automatique, rappelons que notre projet consiste à transformer le fonctionnement manuel de la machine du mélangeur en fonctionnement automatique.

Dans ce chapitre nous allons simuler notre programme. Nous allons étudier le fonctionnement du mélangeur dans le cas d'une automatisation tout au long du travail, du début jusqu'à la fin tout en respectant les conditions de marche. Nous visualisons également l'état de moteurs et des vannes fermé et ouvert d'après la simulation ainsi que les tableaux des variables.

## 4.2 Simulation sous step7

### 4.2.1 Présentation

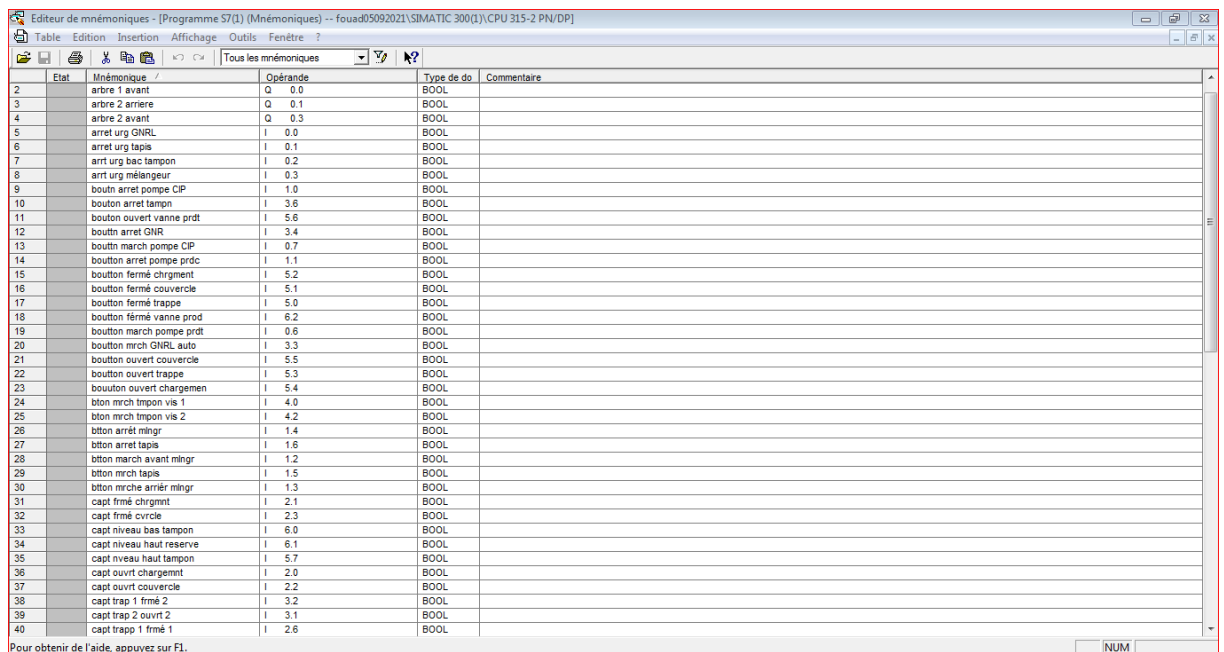
La simulation permet de détecter des erreurs logiques de configuration, par ex. des Valeurs limites incorrectes.

Avant tout, on doit faire appeler toutes les fonctions dans FB1 par la commande « CALL » Dans le bloc OB1

### 4.2.2 Création de la table des mnémoniques

Dans tout programme il faut définir la liste des variables qui vont être utilisées lors de la programmation, pour cela la table des mnémoniques doit être créée. L'utilisation des noms appropriés rend le programme plus compréhensible et plus facile à manipuler.

On édite la table des mnémoniques en respectant notre cahier des charges, pour les entrées et les sorties. La **FIGURE 4.1** suivante présente une partie de la table des mnémoniques.



Éditeur de mnémoniques - [Programme S7(1) (Mnémoniques) -- fouad05092021\SIMATIC 300(1)\CPU 315-2 PN/DP]

Table Edition Insertion Affichage Outils Fenêtre ?

Tous les mnémoniques

Etat	Mnémonique	Opérande	Type de do	Commentaire
2	arbre 1 avant	Q 0.0	BOOL	
3	arbre 2 arriere	Q 0.1	BOOL	
4	arbre 2 avant	Q 0.3	BOOL	
5	arret urg GNRL	I 0.0	BOOL	
6	arret urg tapis	I 0.1	BOOL	
7	arrrt urg bac tampon	I 0.2	BOOL	
8	arrrt urg mélangeur	I 0.3	BOOL	
9	boutn arret pompe CIP	I 1.0	BOOL	
10	bouton arret tampon	I 3.6	BOOL	
11	bouton ouvert vanne prdt	I 5.6	BOOL	
12	boutn arret GHR	I 3.4	BOOL	
13	boutn march pompe CIP	I 0.7	BOOL	
14	bouton arret pompe prdc	I 1.1	BOOL	
15	bouton fermé chargemnt	I 5.2	BOOL	
16	bouton fermé couvercle	I 5.1	BOOL	
17	bouton fermé trappe	I 5.0	BOOL	
18	bouton fermé vanne prod	I 6.2	BOOL	
19	bouton march pompe prdt	I 0.6	BOOL	
20	bouton mrch GNRL auto	I 3.3	BOOL	
21	bouton ouvert couvercle	I 5.5	BOOL	
22	bouton ouvert trappe	I 5.3	BOOL	
23	bouton ouvert chargemen	I 5.4	BOOL	
24	btou mrch tmon vis 1	I 4.0	BOOL	
25	btou mrch tmon vis 2	I 4.2	BOOL	
26	btton arret mingr	I 1.4	BOOL	
27	btton arret tapis	I 1.6	BOOL	
28	btton march avant mingr	I 1.2	BOOL	
29	btton mrch tapis	I 1.5	BOOL	
30	btton mrche arriér mingr	I 1.3	BOOL	
31	capt frmé chargemnt	I 2.1	BOOL	
32	capt frmé cvrcle	I 2.3	BOOL	
33	capt niveau bas tampon	I 6.0	BOOL	
34	capt niveau haut reserve	I 6.1	BOOL	
35	capt niveau haut tampon	I 5.7	BOOL	
36	capt ouvert chargemnt	I 2.0	BOOL	
37	capt ouvert couvercle	I 2.2	BOOL	
38	capt trap 1 frmé 2	I 3.2	BOOL	
39	capt trap 2 ouvert 2	I 3.1	BOOL	
40	capt trapp 1 frmé 1	I 2.6	BOOL	

Pour obtenir de l'aide, appuyez sur F1.

NUM

**FIGURE 4.1** : Déclaration des variables dans le tableau mnémoniques

Etat	Mnémonique	Opérande	Type de do	Commentaire
41	captrapp 1 ouvert 1	I 2.4	BOOL	
42	captrapp 1 ouvert 2	I 2.5	BOOL	
43	captrapp 2 ouvert 1	I 3.0	BOOL	
44	captrapp 2 ferm 1	I 2.7	BOOL	
45	CIP pompe mche	Q 2.0	BOOL	
46	contact relais thrm3	I 4.4	BOOL	
47	contact relais thrmq 1	I 3.5	BOOL	
48	contact relais thrmq4	I 4.5	BOOL	
49	contactv relais thrmq2	I 3.7	BOOL	
50	couvercle fermé	Q 1.1	BOOL	
51	couvercle ouvert	Q 1.0	BOOL	
52	debit	PW 304	WORD	
53	fin crs porte trappe	I 0.5	BOOL	
54	fin crse bac tampon	I 1.7	BOOL	
55	fin de crs couvercle mngr	I 0.4	BOOL	
56	mode auto	I 4.7	BOOL	
57	mode manuel	I 4.6	BOOL	
58	pompe eau	Q 3.4	BOOL	
59	pompe march produit	Q 2.1	BOOL	
60	porte chrgm frmé dstrbr	Q 1.3	BOOL	
61	porte chrgm ouvrt dstrbr	Q 1.2	BOOL	
62	Read Analog Value 464-2	FC 105	FC 105	Read Analog Value 464-2
63	Read Analog Value 466-1	FC 106	FC 106	Read Analog Value 466-1
64	SCALE	FC 8	FC 8	Scaling Values
65	tapis marche	Q 2.2	BOOL	
66	trappe 1 fermé dstrbr	Q 0.5	BOOL	
67	trappe 1 ouvert dstrbr	Q 0.4	BOOL	
68	trappe 2 frmé dstrbr	Q 0.7	BOOL	
69	trappe 2 ouvert dstrbr	Q 0.6	BOOL	
70	UNSCALE	FC 9	FC 9	Unscaling Values
71	vanne CIP 1	Q 2.3	BOOL	
72	vanne CIP 2	Q 2.4	BOOL	
73	vanne produit ouvert	Q 2.5	BOOL	
74	vis tampon 1 arrière	Q 1.5	BOOL	
75	vis tampon 1 avant	Q 1.4	BOOL	
76	vis tampon 2 arrière	Q 1.7	BOOL	
77	vis tampon 2 avant	Q 1.6	BOOL	
78				

FIGURE 4.2 : Déclaration des variables dans le tableau mnémoniques

### 4.2.3 Programme de mélangeur automatique

Nous avons, choisi le langage de programmation à contact (CONT) et programmé le bloc FB1 et on le fait l'appelle dans le bloc d'organisation (OB) .comme elle présente la figure

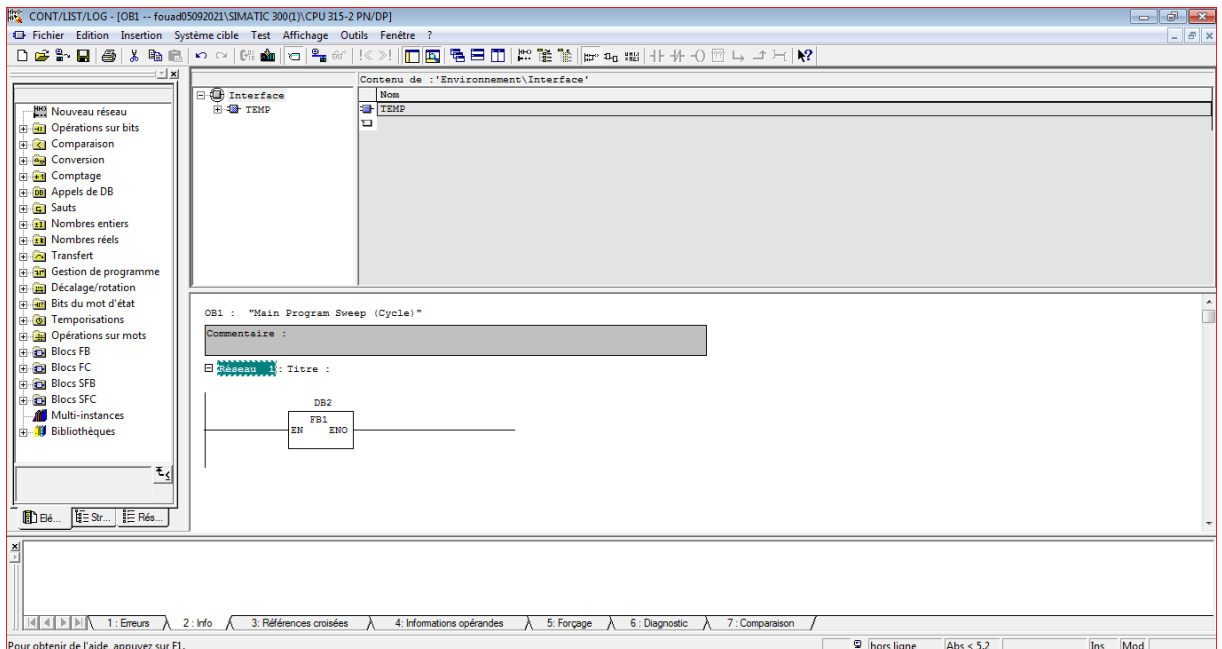



FIGURE 4.3: Appelle de bloc de programme dans OB1

### 4.2.3.1 Activation de la simulation

La figure ci-après explique comment simuler notre programme pour qu'on puisse le suivre en ligne, s'il fonction bien ou pas.

Pour activé la simulation, on aura qu'a appuyé sur l'icône 

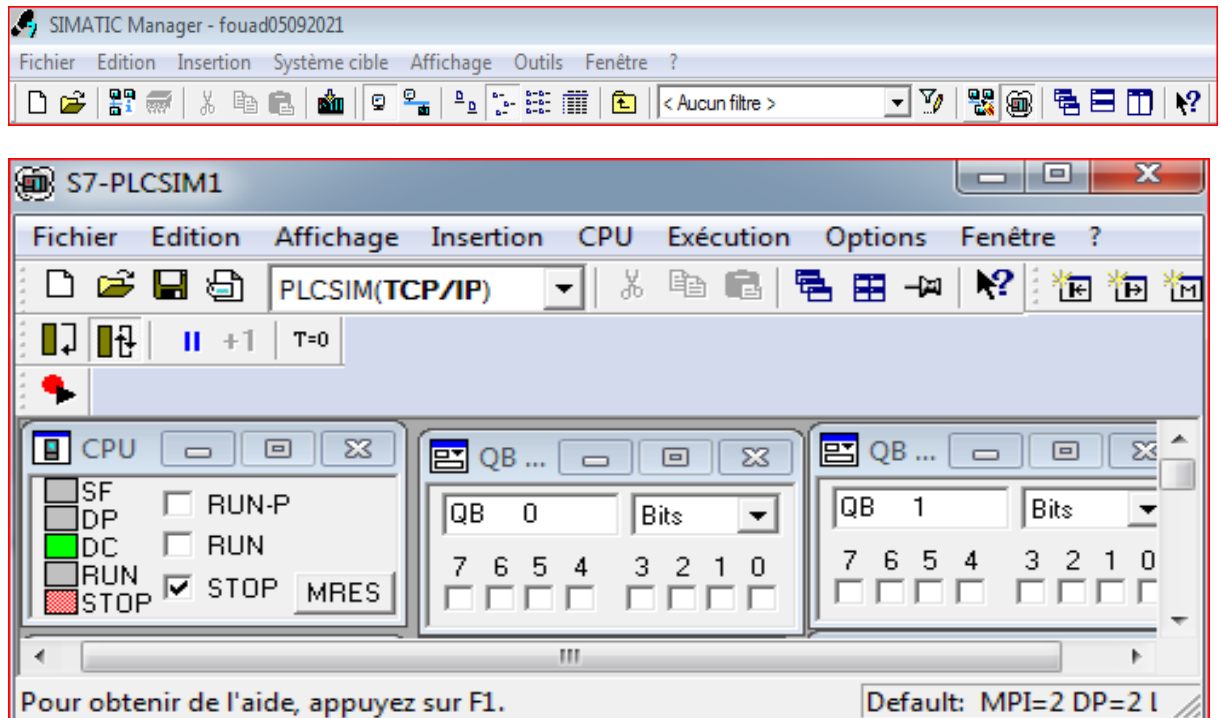


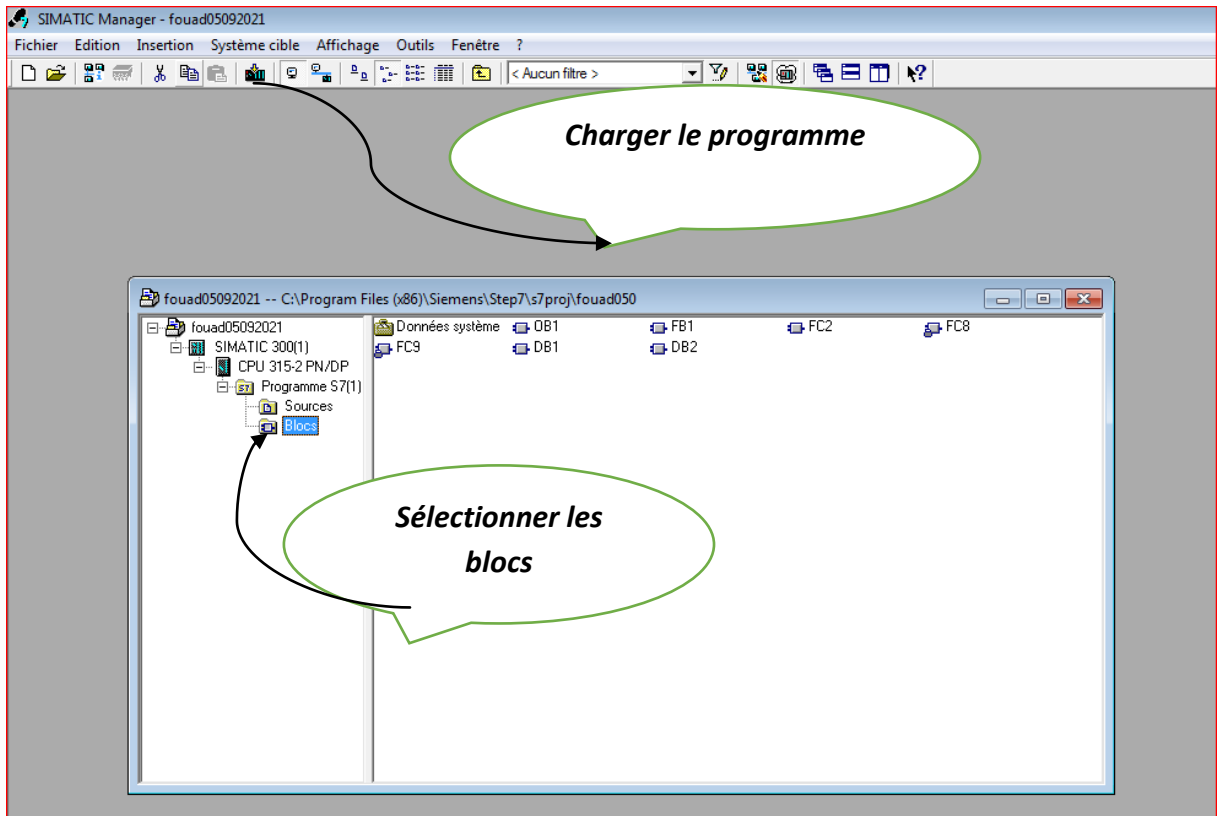


FIGURE 4.4 : Activation de la simulation

### 4.2.3.2 Chargement du programme

Il suffit d'abord sélectionner les blocs de programme en appuyant sur , et pour le chargement du programme se fait en appuyant sur , cette action permet de charger le programme dans la CPU pour son exécution.




**FIGURE 4.5:**Chargement de programme

**Remarque :**

Il faut d'abord sélectionner « BLOCS » avant de charger le programme et il faut le simulateur doit être ouvert.

**4.2.3.3 Le lancement de la simulation :**

Cela se fait en appuyant sur « RUN-P », et on appuyant sur visualisation  dedans le bloc de programme comme le montre la figure ci-dessous :



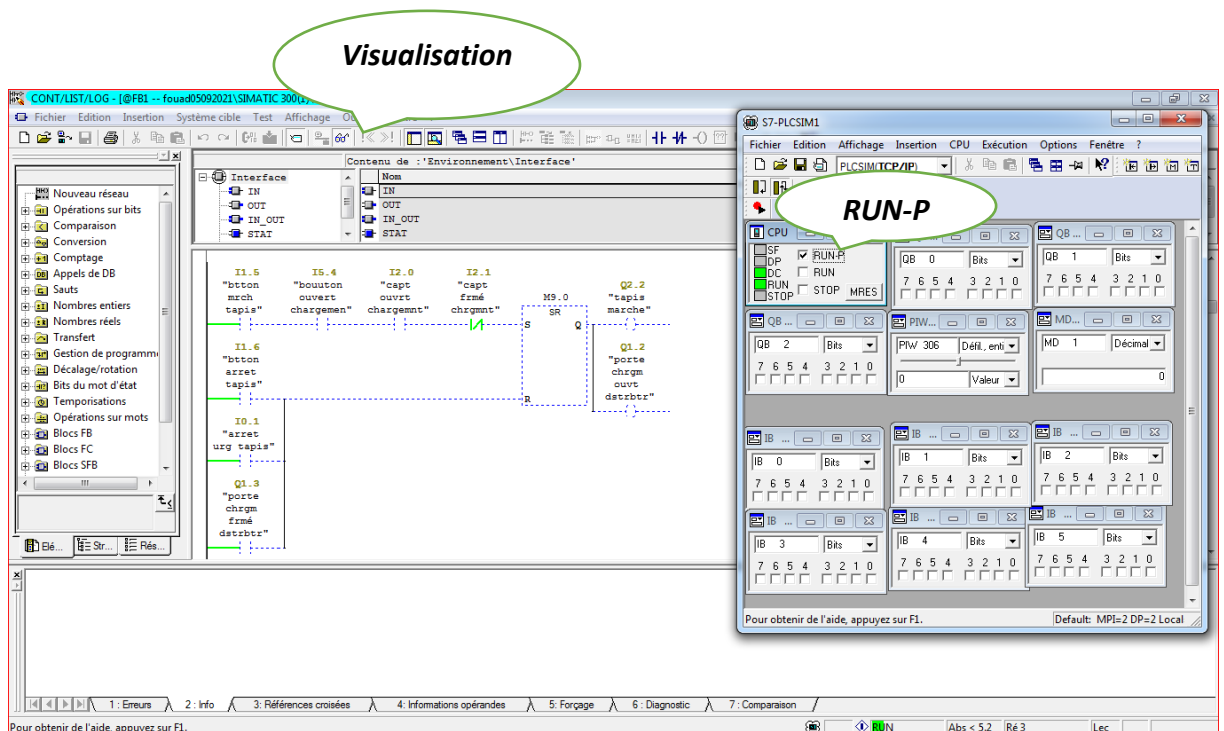


FIGURE 4.6 : Lancement le simulateur

## 4.2.4 Visualisation

### 4.2.4.1 Visualiser les variables :

Dans l'outil configuration matérielle, on peut visualiser ou forcer directement les opérants des modules configurés, la fonction visualiser des variables permet de contrôler le câblage des modules d'entrée et la fonction Forcer des variables, celui des modules de sortie.

Il existe deux possibilités pour visualiser les variables :

- actualisation unique des valeurs d'état à l'aide de la commande **Variable -> Actualiser valeurs d'état** ou en cliquant sur
- actualisation cyclique des valeurs d'état à l'aide de la commande **Variable -> Visualiser** ou en cliquant sur

#### Remarque :

Avant de pouvoir activer les fonctions "Visualiser" et "Forcer", il faut établir une liaison ligne avec la CPU.

### 4.2.4.2 Visualisation de tapis :

Après avoir chargé le programme dans notre CPU, on double clic sur la fonction « VAT 1 » qu'on a créé au pare avant et on voit aussi la simulation de les bobines de tapis et la porte de chargement dans les schémas de bloc qui se trouve dans FB1. On peut voir que le tapis est en mode marche avec son bit (A 2.2) qu'il est à un (1) et true dans (VAT 1), On aura la figure suivante :

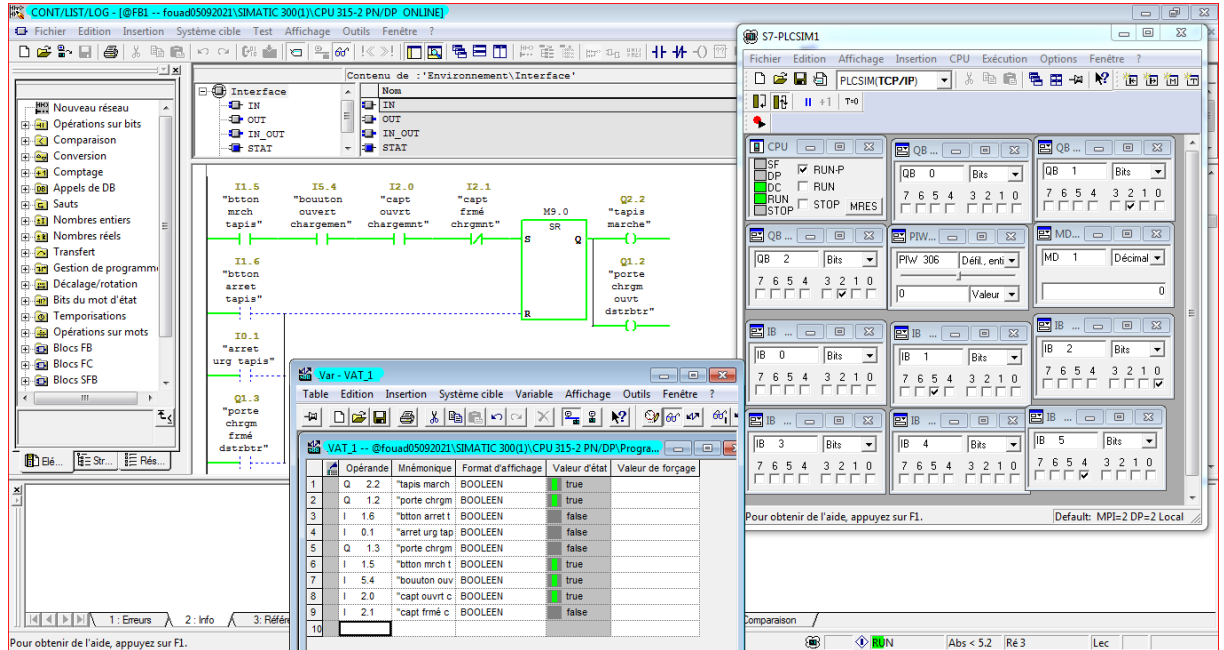


FIGURE 4.7:Table de variable et simulateur

#### 4.2.4.3 Visualisation des conditions de démarrage de mélangeur :

On peut voir condition de démarrage de mélangeur d'après cette figure :

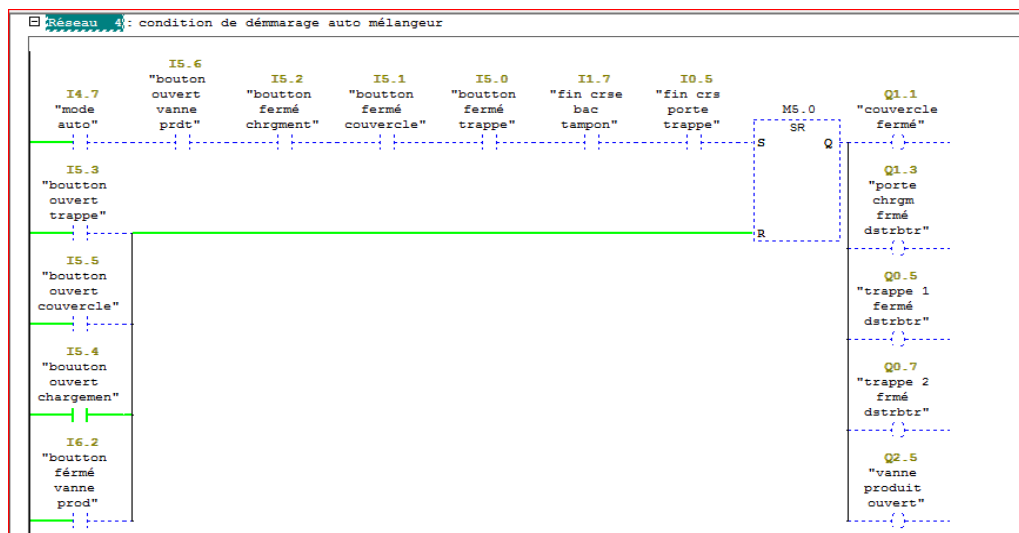
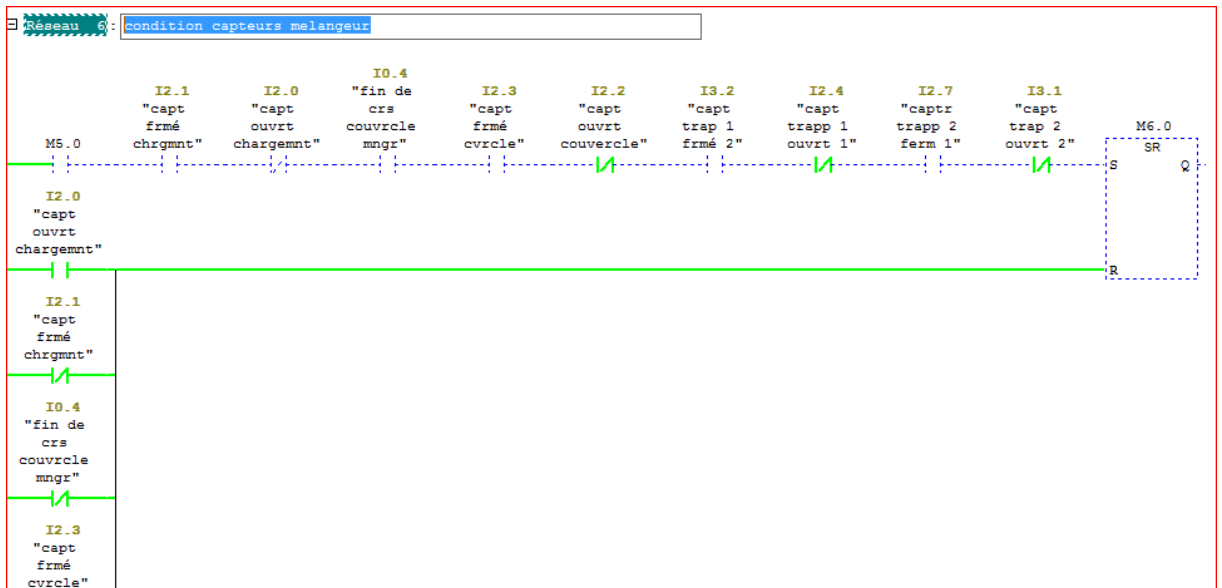


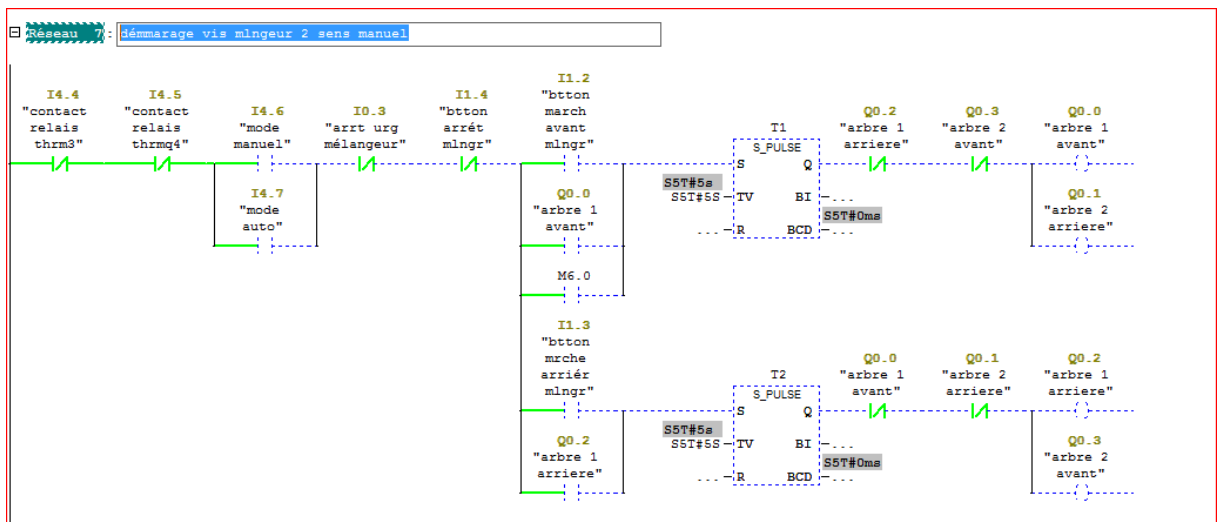
FIGURE 4.8:Conditions de démarrage



**FIGURE 4.9:** Conditions des capteurs

#### 4.2.4.4 Visualisation de démarrage vis mélangeurs 2 sens automatique :

On peut voir de démarrage vis mélangeurs 2 sens automatique d'après cette figure :



**FIGURE 4.10 :** Démarrage vis mélangeur

Pour la suite après 20 minutes de mélange de produit dans la cuve de mélangeur, ils vont ouvrir les trappes automatiquement pour aller à la 2ème cure de stockage et l'envoyer directement par la pompe de production à la troisième cuve.

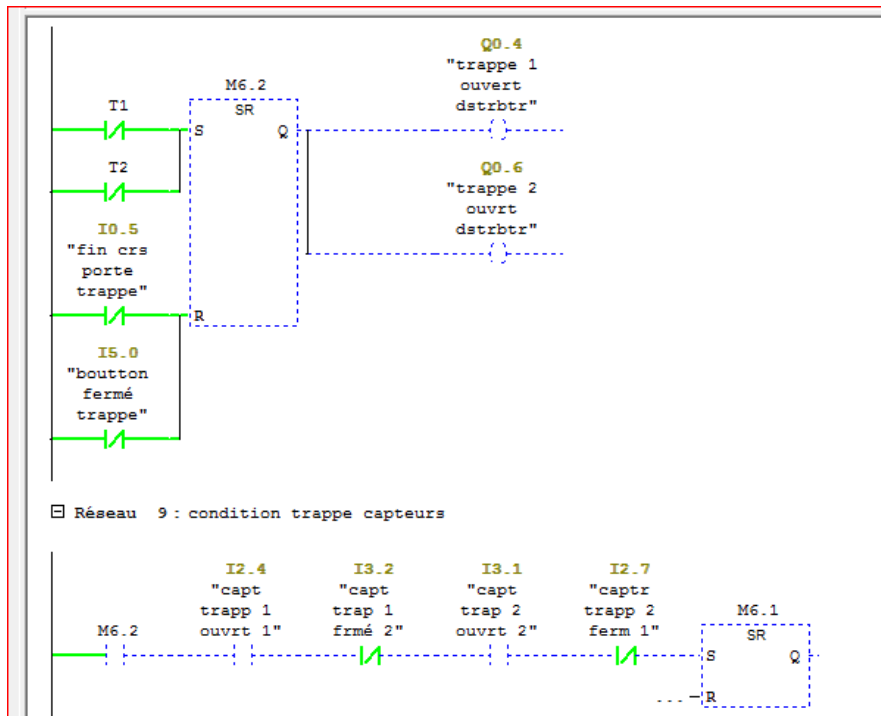


FIGURE 4.11: Ouverture des trappes

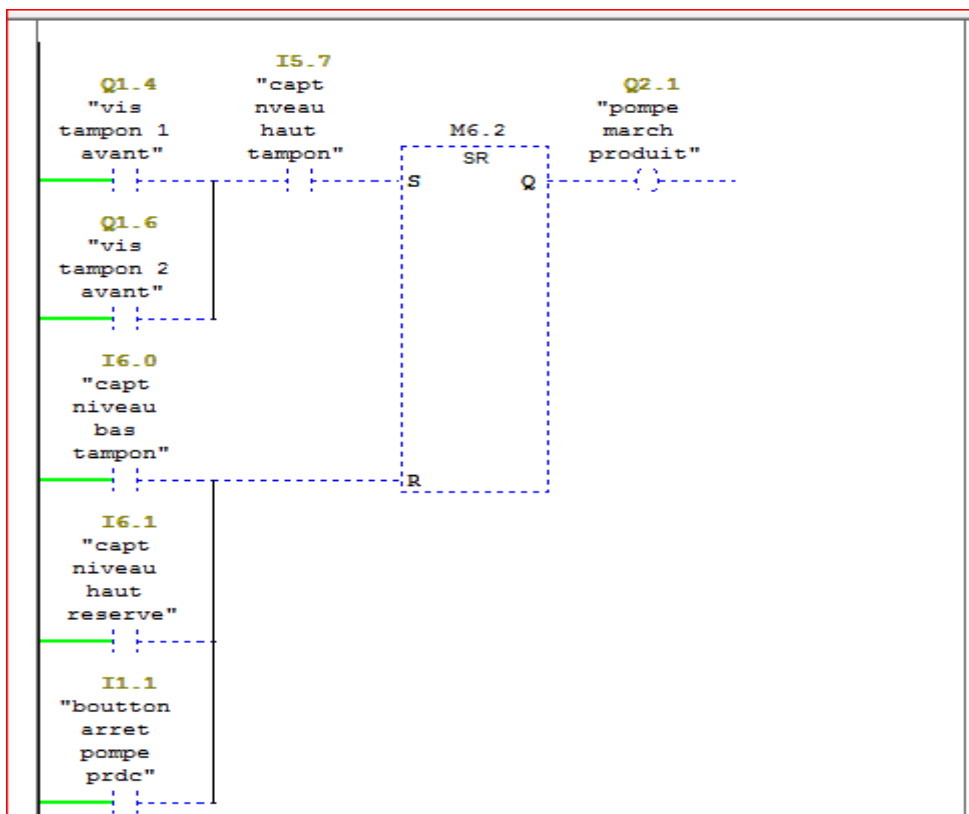


FIGURE 4.12: Démarrage de pompe

### **4.3 Conclusion**

Dans ce chapitre, nous avons expliqué le fonctionnement du mélangeur suite à une programmation et une simulation. Nous avons pour ce fait, réalisé toutes les étapes assurant sa marche selon notre programme dans l'automate.

Rappelons que ceci, n'était faisable que grâce à la configuration matériel, chargement du programme et la fenêtre de simulation ainsi que la visualisation nous permettant de savoir l'état des entrées et des sorties de l'automate.

Nous avons également et en dernier réalisé le tableau de variable qui nous permis de réaliser le forçage des variables.

## Conclusion générale

---

L'automatisme industriel offre plusieurs débouchés dans les domaines de la maintenance comme la conception. Les automaticiens peuvent travailler dans plusieurs secteurs comme l'agroalimentaire, la cimenterie ou l'énergie comme technicien ou ingénieur de maintenance ou aussi comme ingénieur de conception en bureau d'étude. Ils programment et configurent les automates programmables, réalisent les schémas électriques et le câblage d'armoires électriques.

Dans le cas de notre étude, l'objectif consistait à automatiser entièrement le fonctionnement d'un mélangeur industriel manuel dans un secteur agroalimentaire, tout en assurant la sécurité de la machine.

Pour ce faire, nous avons dans premier lieu, en deux chapitres, présenté des généralités sur les systèmes automatisés et les automates programmables ainsi que leur architecture et fonctionnement. En troisième chapitre nous avons introduit et décrit le mélangeur industriel en expliquant son fonctionnement et en mettant en valeur le logiciel STEP 7 utilisée dans la programmation en quatrième chapitre.

L'étude en question a fait relever l'important rôle de l'automatisation dans une industrie et a accentué la nécessité d'y avoir recours. Les trois premiers chapitres ont fait relever des généralités élémentaires et fondamentales relatives à l'automatisation industrielle et capturent les plus utilisés dont l'enjeu et d'assurer le bon fonctionnement et la sécurité des machines. Ils ont également mis en avant, les automates programmables industriels, leur architecture, structure et l'ensemble des modules, tout en introduisant l'automate programmable S7-300 comme étant une machine dotée d'une grande flexibilité, tout en s'intéressant à la composition technique du mélangeur et son fonctionnement.

Le quatrième chapitre nous a permis de réaliser une simulation de la programmation de l'automate correspondant au système du mélangeur, d'étudier le fonctionnement de ce dernier tout au long son travail, du début jusqu'à la fin tout en respectant les conditions de marche tout en visualisant l'état de moteurs et des vannes fermé et ouvert d'après la simulation ainsi que les tableaux des variables.

Par le biais du présent travail, nous avons répondu à nos questions posées au paravent et nous avons automatisé le travail d'un mélangeur industriel manuel grâce à la programmation d'un automate programmable industriel.

# Bibliographies

---

- [1] P.Grare et I. Kacem, « AUTOMATISME, CE QU'IL FAUT SAVOIR SUR LES AUTOMATISMES », Livre, Paris : Ellipse, 2008.
- [2] J. Perrain F. Binet, J.J. Doumer, C. Merlaud, J.P. Trichard, «Automatique et informatiqueindustrielle», Edition Nathan, 2006.
- [3] A. Simon, " Automates programmables industriels Niveau, l'Édition-Liège, 1991
- [4] P.Brard, G.Colombari, « Le GRAFCET, outil de description des automatismes séquentiels », édition Techniques de l'ingénieur, Paris, 1999.
- [5]A.Foudil-bey, «Etude et simulation d'une chaine de remplissage et d'emballage à base de siemens S7-300», Mémoire de Master en Automatique, Département d'Electronique, Université de Blida 1, Juin 2019
- [6]M.Boudah, « Automatisation d'une ligne de préparationde jus à COJEK CEVITAL », Mémoire de Master en Électronique, Département d'Automatique, de Télécommunication & d'Électronique, Université Abderrahmane Mira De Béjaïa, 2016-2017
- [7]G. Alain, « Les automates programmables industriels » Novembre, 2004
- [8] Document siemens, " information et formation, automatisation et entrainements, programmation niveau A ", Edition siemens AG, 2003.
- [10] SIMATIC S7-300 Système d'Automatisation S7-300 Caractéristiques des modules, Version Juin 2017.
- [11] L. Bergougnoux, «automate programmable industriel», Polytech Marseille, cours, 2016.



[12] Manuel Siemens Step7, « Installation programmation», édition Siemens AG, 2010.

[13]M. BOUROUIS, B. CHIBOUTI, « Automatisation d'alimentation de trémies d'une ligne de production de sucre avec supervision IHM », Mémoire de fin d'études Master en Automatismes, Département Génie Électrique, Université Abderrahmane Mira Bejaia, 2020.

[14]M.Grout et P.Salaün, « INSTRUMENTATION INDUSTRIELLE, SPECIFICATION ET INSTALLATION DES CAPTEURS ET VANNES DE REGULATIONS », Livre, Paris : Dunod, 2012.

[15] Y.Hamour et A. Sofiane, « ETUDE COMPARATIVE DE VARIATEUR DE VITESSE ELECTRONIQUE ET HYDRODYNAMIQUE POUR STATION DE POMPAGE DE PETROLE (SONATRACH) », Mémoire de fin d'études Master, Département de Génie Electrique, Université de Bejaia, 2010.

[16] A. Laifaoui, « Cours schémas et appareillages électriques (3ieme LAGE) ». UNIVERSITE de BEJAIA. (2012 / 2013).

[17]S.Kaci, Y.Khodja, « Implémentation d'un contrôleurPID sur un dsPIC 33F », Mémoire de fin d'études Master en Automatismes, Département Automatique Télécommunication et Électronique, Université Abderrahmane Mira de Bejaia, 2014-2015

[18] A. Simon, « Automate programmable industriel », édition l'ELAN, liège 1991.

[19]A.Mzoughi, A.Zitouni, H.Labidi, M. Ben Henda et F.Ayari, « GENIE ELECTRIQUE » Manuel de cours 4ème année de l'enseignement secondaire tunisien, OMEGA EDITIONS, 2013.

[20]<https://www.automation-sense.com/blog/automatisme/automatisme-industriel.html>(Consulté le 31/08/2021)

[21]<https://www.legarrec.com/entreprise/melangeur-industriel-principes/>(Consulté le 31/08/2021)

[22]<https://junior.universalis.fr/encyclopedie/actionneur/> (consulté en juin 2020).

[23] <https://geea.org.pagesperso-orange.fr/PNEUM/verin.htm> (consulté en juillet 2020).

[24] [https://cache.industry.siemens.com/dl/files/591/8860591/att\\_105768/v1/HB\\_CPU312IF\\_M\\_bis\\_318-2DP\\_F.pdf](https://cache.industry.siemens.com/dl/files/591/8860591/att_105768/v1/HB_CPU312IF_M_bis_318-2DP_F.pdf)(Consulté le 14/08/2021)

[25] <https://www.maxicours.com/se/cours/regulateurs-de-pression/>(consulté en août 2020).