

*République Algérienne Démocratique et Populaire*  
*Ministère de l'Enseignement Supérieur Et de La Recherche Scientifique*



*Université de Ghardaïa*

N° d'ordre :

N° de série :

Faculté des Sciences et Technologies  
Département des Sciences et Technologie

**Mémoire présenté en vue de l'obtention du diplôme de**

**MASTER**

**Domaine : *Sciences et Technologies***

**Filière : Automatique**

**Spécialité : *Automatique***

Par : - BOULENACHE ABDALLAH  
- HOUACHE DAOUED

**Thème**

**Automatisation des processus industriels par l'automate programmable industriel (API)**

**Soutenu publiquement le : ..../05/2017**

**Devant le jury :**

Laadjal Boumedien	<b>M.A.A</b>	Univ. Ghardaïa	<b>Président</b>
Bennouna Med salah	<b>M.A.A</b>	Univ. Ghardaïa	<b>Examineur</b>
Fiha khire Mahdi	<b>M.A.A</b>	Univ. Ghardaïa	<b>Examineur</b>
BEKKAR Belgacem	<b>M.A.A</b>	Univ. Ghardaïa	<b>Encadreur</b>

**Année universitaire : 2016/2017**



## Remerciements

*En premier lieu, nous remercions Dieu, notre créateur, qui  
Nous a donné la force et la persévérance pour réaliser ce travail.*

*Nous tenons à exprimer notre profonde gratitude et nos sincères  
remerciements à tous ceux qui ont contribué, de près ou de loin à  
l'élaboration de ce rapport de fin d'étude.*

*Aussi, nous tenons à remercier infiniment :*

- ❖ *Nos chers parents pour leurs soutiens au long de nos études.*
- ❖ *Mr. Belgacem bekkar notre encadreur qui nous a accordé son soutien, son aide indéfectible et surtout sa patience et sa gentillesse.*

*Nos remerciements vont aussi à :*

- ❖ *Tous nos amis qui nous ont donné la force et l'ambiance du travail collectif.*

*On a l'honneur et le plaisir de remercier également tous les enseignants De la spécialité d'automatique et de nos Group d'étudiants (2<sup>eme</sup> année master automatique).*



## Dédicace

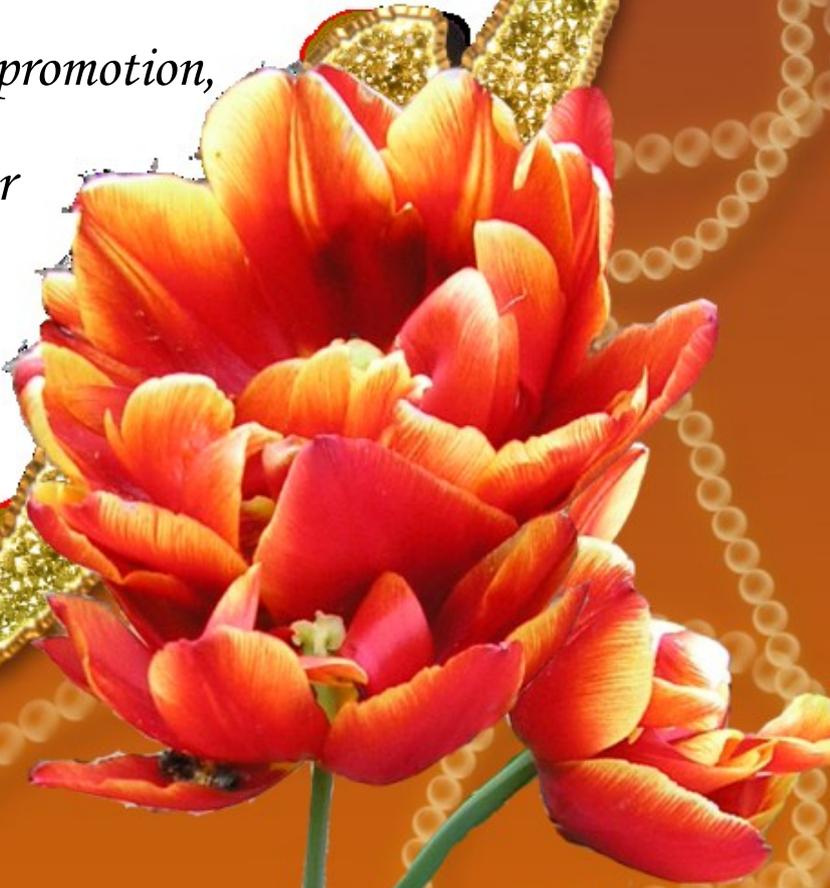
*Je dédie ce mémoire à tous ceux qui me sont chers :*

✓ *À mes parents, pour leurs sacrifices, aides, conseils et leur générosité qui grâce à eux j'ai pu réaliser ce modeste travail.*

✓ *À mes frère rostom et ali et mes sœurs*

✓ *À mes amis de promotion,  
2eme année master  
automatique*

*a. boulenache*



## Dédicace

*Je dédie ce mémoire à tous ceux qui me sont chers :*

✓ *A mes parents, pour leurs sacrifices, aides, conseils et leur générosité qui grâce à eux j'ai pu réaliser ce modeste travail.*

✓ *A mes frères et mes sœurs*

✓ *A mes amis de promotion,*

*2eme année master  
automatique*

*d.houache*



## Résumé :

Le sujet traité dans ce mémoire de fin d'étude porte sur la modélisation des systèmes automatisés industriels par un automate programmable industriel (Schneider), ce qui nous permis de creuser nos connaissance dans ce domaine. Le travail présenté constitue une description des systèmes automatisés, et de leurs différents outils de modélisation graphique.

Dans ce mémoire On a pu modéliser un cahier des charges d'un ascenseur et de parking en plus, un autre système industriel, à travers un outil de modélisation graphique "Grafcet", ce dernier est un outil puissant pour faire la modalisation d'un système complexe. Ce travail constitue aussi la simulation de ce système par le programme " LADDER".

**MOTS CLES :** Systèmes automatisés, automate programmable industriel, système, programmation.

## ملخص:

الموضوع المعالج في مذكرة التخرج هذه يتمحور حول نمذجة الانظمة الصناعية الاوتوماتيكية باستعمال جهاز تحكم الي قابل للبرمجة (Schneider), مما يسمح لنا بتقوية معارفنا في هذا المجال. العمل المقدم يحوي وصف للانظمة الاوتوماتيكية ومختلف ادوات النمذجة بالرسوم البيانية.

تمكنا في هذه المذكرة من نمذجة دفتر شروط لمصعد الي و موقف للسيارات اضافة الى نظام صناعي اخر باستعمال اداة البرنامج (GRAFCET), هذه الاخيرة تعتبر اداة قوية لنمذجة نظام معقد. هذا العمل يحتوي ايضا محاكات لهذا النظام باستخدام لغة البرمجة (LADDER)

**كلمات مفتاحية :** الانظمة الاوتوماتيكية , الالية المبرمجة الصناعية, نظام, البرمجة

# Table des matières

Remerciements .....	i
<i>Dédicace</i> .....	i
<i>Dédicace</i> .....	ii
Résumé : .....	iii
Table des matières .....	iv
Liste des figures.....	viii
Liste des tableaux .....	xi
Liste des abréviations .....	xii
Introduction générale.....	1
Chapitre 1 : .....	2
Les systèmes automatisés industriels .....	2
1.1 Introduction.....	3
1.2 Définitions de l'automatisme.....	3
1.3 Structure générale d'un automatisme .....	4
1.3.1 Partie opérative .....	4
1.3.2 Partie commande .....	6
✓ Logique programmée .....	6
1.3.3 Poste de contrôle (interface homme-machine (IHM)).....	7
1.4 Communication dans un système automatisé .....	7
1.5 Connaissance de matériel.....	9
1.5.1 Pré-actionneur.....	9
1.5.2 Actionneur .....	10
1.5.3 Transmetteur .....	10
1.5.4 Effecteur .....	11

1.5.5	Capteur.....	11
1.5.6	Interface H / M.....	12
1.5.7	Partie commande .....	12
1.5.8	Interface H / M.....	12
1.6	Objectifs de l'automatisation .....	13
1.7	Conséquences de l'automatisation .....	13
1.7.1	Avantages de l'automatisation.....	13
1.7.2	Inconvénients de l'automatisation .....	14
1.8	Conclusion .....	15
Chapitre 2 : .....		16
Les Automates Programmables Industriels .....		16
2.1	Introduction.....	17
2.2	Définition .....	17
2.3	Architecture general d'un automate programmable industriel .....	18
2.3.1	Unité centrale.....	18
2.3.2	Configuration.....	18
2.3.3	Fonctions Internes.....	18
2.3.4	Programme Utilisateur.....	19
2.3.5	Supervision .....	19
2.3.6	Interfaces.....	19
2.4	La structure interne d'un API.....	19
2.4.1	L'unité centrale .....	20
2.4.2	Le module d'entrées .....	21
2.4.3	Le module de sorties .....	22
2.4.4	Le module d'alimentation.....	22
2.4.5	Le module de communication .....	22

2.5	Domaines d'utilisation des API .....	23
2.5.1	Exemples de domaines d'application .....	24
2.6	Langage de programmation .....	26
2.6.1	Le GRAFCET .....	27
2.6.2	Langage Ladder .....	28
2.7	Logiciel de programmation .....	30
2.8	Différents types d'API .....	38
2.9	Critères de choix d'un automate .....	39
2.10	Sécurité .....	40
2.11	Conclusion .....	41
Chapitre 3 : .....		42
Automatisation des processus industriels .....		42
3.1	Introduction .....	43
3.2	Exemple 1 : Contrôle continu de bouteilles .....	43
3.2.1	Description .....	43
3.2.2	Cahier des charges fonctionnels .....	43
3.2.3	GRAFCET .....	45
3.2.4	Programmation .....	45
3.3	Exemple 2 : Barrière parking .....	47
3.3.1	Description .....	47
3.3.2	Cahier des charges fonctionnel de Barrière Parking .....	47
3.3.3	Schémas pneumatique .....	49
3.3.4	GRAFCET .....	50
3.3.5	Programmation .....	51
3.4	Exemple 3 : l'ascenseur .....	52
3.4.1	Description .....	52

3.4.2	Cahier des charges d'ascenseur relié à l'automate .....	54
3.4.3	GRAFCET .....	57
3.4.4	Programmation .....	58
3.5	conclusion .....	59
	Conclusion générale .....	60
	Bibliographies.....	61
	Annexe.....	63

# Liste des figures

Figure 0-1 : Structure générale d'un automatisme. ....	4
Figure 0-2 : l'échange d'information dans les différents niveaux.....	8
Figure 0-3 : Niveaux de commande. ....	9
Figure 0-4 : différents types de pré-actionneurs.....	9
Figure 0-5: différents types des actionneurs.....	10
Figure 0-6: Transmetteurs Sans transformation de mouvement.....	10
Figure 0-7: Transmetteurs avec transformation de mouvement.....	10
Figure 0-8: Effecteurs. [9] .....	11
Figure 0-9: capteurs.....	11
Figure 0-10: les interfaces H / M.....	12
Figure 0-11: partie commande.....	12
Figure 0-12: interface M / H.....	13
Figure 0-1: Structure général d'un API. ....	18
Figure 0-2 : Structure interne d'un A P I.....	19
Figure 0-3 : L'industrie d'automobile. ....	24
Figure 0-4: appareil d'emballage. ....	25
Figure 0-5: l'usinage du bois. ....	25
Figure 0-6 : l'unité de perçage.....	25
Figure 0-7: machine de détection des défauts. ....	26
Figure 0-8: portail automatisé.....	26
Figure 0-9: Exemple d'un programme en grafcet. ....	27
Figure 0-10: Exemple d'un programme en Ladder. ....	29
Figure 0-11: création d'un fichier de programmation. ....	32
Figure 0-12: choix de la catégorie d'API disponible.....	32

Figure 0-13: type et référence d'automate sélectionné.....	33
Figure 0-14: facultative d'autre module. ....	33
Figure 0-15: langage de programmation choisi.....	34
Figure 0-16: page d'édition de programme Ladder.....	34
Figure 0-17: Exemple de marche et arrêt d'un moteur.....	35
Figure 0-18: détermination l'état du contact. ....	35
Figure 0-19: insertion d'autres symboles. ....	36
Figure 0-20: réalisation d'un schéma en Ladder. ....	36
Figure 0-21: simulation d'un programme en Ladder.....	37
Figure 0-22: Transfert du programme dans l'API.....	37
Figure 0-23: mise en marche d'API. ....	38
Figure 0-24: Automate programmable industriel.....	39
Figure 0-1: Chaine de remplissage bouteille. ....	44
Figure 0-2: grafcet de la chaine de remplissage des bouteilles. ....	45
Figure 0-3: La Barrière parking.....	47
Figure 0-4: Schémas descriptif de gestion de parking.....	48
Figure 0-5: Les places sont disponibles.....	49
Figure 0-6: Les Places sont occupées.....	49
La Figure 0-7 représente le schéma pneumatique :.....	49
Figure 0-8: Schémas pneumatique dessinée.....	50
Figure 0-9: Le GRAFCET principal de parking.....	51
Figure 0-10: Ascenseur électrique à adhérence.....	53
Figure 0-11: Ascenseur électrique à Tambour. ....	54
Figure 0-12: L'ascenseur « Hydraulique ». ....	54
Figure 0-13: LE GRAFCET principal d'ascenseur. ....	57



## Liste des tableaux

Tableau 0-1: les principaux éléments d'un réseau LD. ....	30
Tableau 0-1: Entrée sortie de la chaine de remplissage.....	44
Tableau 0-2: Les entrées/sorties de Barrière parking. ....	50
Tableau 0-3: Entrée/sortie d'ascenseur. ....	55
Tableau 0-4: les ordres de commandes d'ascenseur. ....	55

# Liste des abréviations

**T.O.R** : tout ou rien.

**API** : Automate programmable industriel.

**I.H.M** : interface homme / machine.

**PLC** : Programmable Logic Controller.

**PO** : partie opérative.

**CPU** : en anglais (*central processing unit*), Un processeur.

**UL** : Unité Logique.

**UAL** : Unité Arithmétique et Logique.

**RAM** : (*Random Access Memory*).

**EAROM** : (*Electrically Alterable Read Only Memory*).

**PROM** : (*Programmable Read Only Memory*).

**EEPROM** : (*Electric Erasable PROM*).

**Langage ST** : (*Structured Text*).

**Langage IL** : (*Instruction List*).

**Langage FBD** : en anglais (*function block diagram*), Boîte fonctionnelles.

**LD** : (*Ladder Diagramme*) ou le langage Ladder.

**E/S** : entrée/sortie.

**CP** : le capteur.

# Introduction générale

L'automatique est devenu indispensable dans l'industrie. Il a pour objectif de concevoir et d'étudier les divers automatismes en mettant en œuvre les actionneurs électriques et pneumatiques. Chaque système automatisé possède une partie commande et une partie opérative. Dans la partie commande, l'automate programmable représente l'élément principal de la machine ou de l'installation, car c'est celui qui renferme le programme et doit procéder à son exécution en fonction de l'état des entrées et des sorties, mais la partie opérative représente en général les actionneurs ou bien les paramètres gérés.

Dans le domaine de l'automatisation, comme dans d'autres techniques, l'informatique a révolutionné beaucoup de choses. La connexion d'automates à l'ordinateur a permis de franchir une étape de plus dans la voie du progrès technologique.

De nos jours, on trouve les systèmes automatisés un peu partout, on peut citer comme exemple bien spécifié (les ascenseurs, la gestion de parking, ...) qui font l'objet du modeste travail de notre projet de fin d'étude.

Ce mémoire sera composé de trois chapitres qui sont exposés de manière séquentielle :

- Dans le premier chapitre on donne une description théorique sur les systèmes automatisés, et la présentation de ses différentes parties.
- Le deuxième chapitre est destiné à la définition et la présentation de différentes utilisations d'automate. Ensuite, nous présentons l'architecture d'API, les différents types et les critères de choix d'automate,
- En fin le troisième chapitre sera consacré à la simulation et la réalisation de nos systèmes choisie (parking), en d'autre terme la création du projet dans le logiciel (*Zelio Software*).

# **Chapitre 1 :**

# **Les systèmes automatisés in-**

# **dustriels**

## 1.1 Introduction

Depuis toujours l'homme est en quête de bien-être. Cette réflexion (qui rejoint la notion de besoin) peut paraître bien éloignée d'un cours de sciences industrielles, pourtant c'est la base de l'évolution des sciences en général, et de l'automatisation en particulier. L'homme a commencé par penser, concevoir et réaliser. Lorsqu'il a fallu multiplier le nombre d'objets fabriqués, produire en plus grande quantité, l'automatisation des tâches est alors apparue : remplacer l'homme dans des actions pénibles, délicates ou répétitives.

Dans ce chapitre on pénètre dans le monde des systèmes automatisés, c'est-à-dire une machine, un ensemble de machines ou plus généralement un équipement industriel, consiste à en assurer la conduite par un dispositif technologique. Le système ainsi conçu sait prendre en compte les situations pour lesquelles sa commande a été réalisée. L'intervention d'un opérateur est souvent nécessaire pour assurer un pilotage global du procédé pour surveiller les installations et prendre en commande manuelle (non automatique) tout ou partie du système.

## 1.2 Définitions de l'automatisme

L'automatisation consiste à rendre automatique les opérations qui exigeaient auparavant l'intervention humaine.

Une autre définition [1] :

L'automatisation est considérée comme l'étape d'un progrès technique où apparaissent des dispositifs techniques susceptibles de seconder l'homme, non seulement dans ses efforts musculaires, mais également dans son travail intellectuel de surveillance et de contrôle.

Un automatisme est un sous-ensemble d'une machine, destinée à remplacer l'action de l'être humain dans des tâches en générales simples et répétitives, réclamant précision et rigueur. On est passé d'un système dit manuel, à un système mécanisé, puis au système automatisé.

Dans l'industrie, les automatismes sont devenus indispensables : ils permettent d'effectuer quotidiennement les tâches les plus ingrates, répétitives et, dangereux. Parfois, ces automatismes sont d'une telle rapidité et d'une telle précision, qu'ils réalisent des actions impossibles pour un être humain. L'automatisme est donc synonyme de productivité et de sécurité

Le savoir-faire de l'opérateur est transposé dans le système automatisé, il devient le PROCES-SUS.

Un processus peut être considéré comme un système organisé d'activités qui utilise des ressources (personnel, équipement, matériels et machines, matière première et informations) pour transformer des éléments entrants (les intrants) en éléments de sortie (les extrants) dont le résultat final attendu est un produit.

Le système automatisé de production industrielle doit également répondre à des contraintes économiques et donc à une certaine flexibilité.

Il existe deux concepts technologiques :

- Les automatismes séquentiels,
- Les asservissements (*feedback control*), ou régulation.

Cependant, dans la plupart des systèmes complexes modernes, ces deux types d'automatismes sont trouvés imbriqués.

### 1.3 Structure générale d'un automatisme

Tout système automatisé peut se décomposer selon la Figure 0-1[8] :

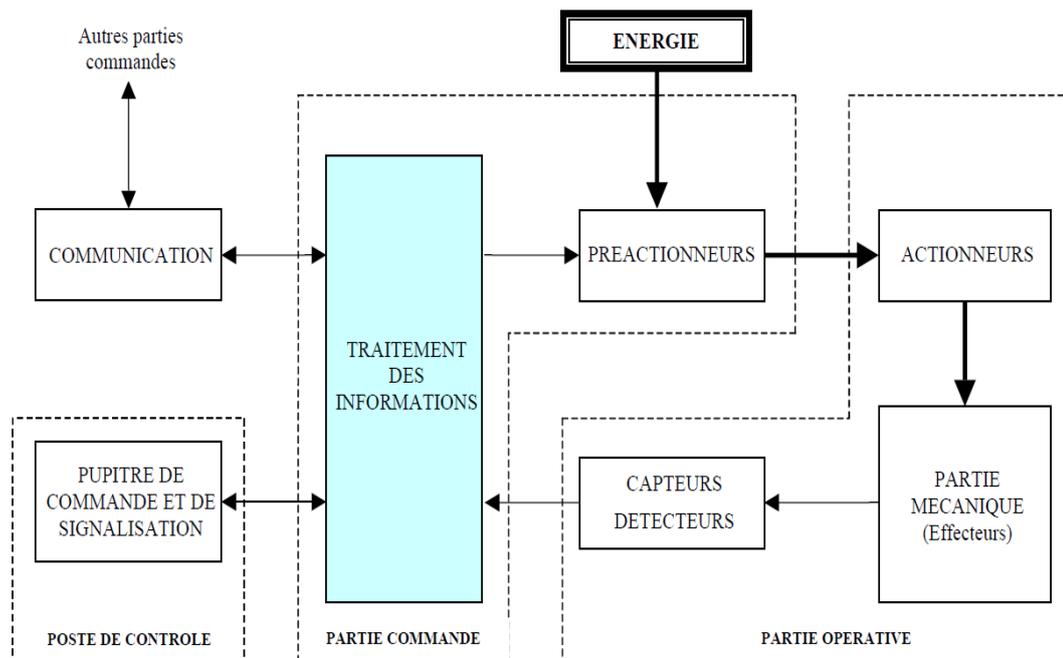


Figure 0-1 : Structure générale d'un automatisme.

#### 1.3.1 Partie opérative

Appelée parfois partie puissance, la partie opérative d'un automatisme assure la transformation ajoutée de la matière d'œuvre.

La partie opérative d'un automatisme est le sous-ensemble qui effectue les actions physiques (déplacement, émission de lumière...), mesure des grandeurs physiques (température, humidité, luminosité...) et rend compte à la partie commande.

Elle est généralement composée d'actionneurs, de capteurs, d'effecteurs et d'un bâti.

Plus simple : la partie opérative reçoit les ordres de la partie commande et les exécute.

- Exemple : l'ensemble des pompes électriques, des électrovannes, des capteurs de niveau d'eau et de température est la partie opérative d'un lave-linge

- **La partie mécanique**

Chariots, glissières, engrenages, poulies, broches... [3].

- **Les actionneurs**

Convertissent l'énergie d'entrée disponible sous une certaine forme (électrique, pneumatique, hydraulique) en une énergie utilisable sous une autre forme, par exemple :

- Energie thermique destinée à chauffer un four (l'actionneur étant alors une résistance électrique).
- Energie mécanique destinée à provoquer une translation de chariot (l'actionneur pouvant être un vérin hydraulique ou pneumatique).
- Energie mécanique destinée à provoquer une rotation de broche (l'actionneur pouvant être alors un moteur électrique).

- **Les préactionneurs**

Reçoivent les signaux de commande et réalisent la commutation de puissance avec les actionneurs. Les pré-actionneurs des moteurs électriques sont appelés contacteurs. Les pré-actionneurs des vérins et des moteurs hydrauliques et pneumatiques sont appelés distributeurs (à commande électrique ou pneumatique).

- **Les capteurs**

Qui communiquent à la partie commande des informations sur la position d'un mobile, une vitesse, la présence d'une pièce, une pression...

- **Les capteurs T.O.R** : (Tout ou rien), qui délivrent un signal de sortie logique, c'est-à-dire 0 ou 1.

**Exemple** : détecteur de fin de course.

- **Les capteurs numériques** : ou « incrémentaux », qui associés à un compteur, délivrent des signaux de sortie numérique.

**Exemple** : capteur ou codeur incrémental utilisé pour la mesure des déplacements des chariots de machine à commande numérique.

- **Les capteurs analogiques** : ou proportionnels » qui permettent de prendre en compte la valeur réelle d'une grandeur physique. **Exemple** : Sonde de température.

### 1.3.2 Partie commande

- Elle donne les ordres de fonctionnement à la partie opérative [3].

Les pré-actionneurs permettent de commander les actionneurs ; ils assurent le transfert d'énergie entre la source de puissance (réseau électrique, pneumatique ...) et les actionneurs. Exemple : contacteur, distributeur.

Ces pré-actionneurs sont commandés à leur tour par le bloc traitement des informations. Celui-ci reçoit les consignes du pupitre de commande (opérateur) et les informations de la partie opérative transmises par les capteurs / détecteurs.

En fonction de ces consignes et de son programme de gestion des tâches (implanté dans un automate programmable ou réalisé par des relais (on parle de logique câblée)), elle va commander les pré-actionneurs et renvoyer des informations au pupitre de signalisation ou à d'autres systèmes de commande et/ou de supervision en utilisant un réseau et un protocole de communication.

Trois technologies sont actuellement utilisées :

- ❖ électromécanique,
- ❖ pneumatique,
- ❖ électronique.

### ✓ Logique programmée

L'enchaînement des mouvements du système automatisé est programmé sous forme d'instructions (programme), traitées et gérées par l'unité centrale de la partie commande. Plusieurs types de systèmes permettent de réaliser cette fonction comme :

- **Les automates programmables industriels (A.P.I.)** : Ils possèdent presque tous un langage adapté au GRAFCET. Ils sont munis de bornes d'entrées et sorties.

- **Les micros et mini-ordinateurs** : Leur utilisation demande des connaissances en informatique. Le **GRAFNET** doit être traduit dans un langage informatique. Ils ne possèdent pas en général de bornes **d'entrées et sorties**.
- **Les micros systèmes** : comme ci-dessus mais possèdent des bornes d'entrées et sorties.

### 1.3.3 Poste de contrôle (interface homme-machine (IHM))

Composé des pupitres de commande et de signalisation, il permet à l'opérateur de commander le système (marche, arrêt, ...). Il permet également de visualiser les différents états du système à l'aide de voyants, de terminal de dialogue ou d'interface homme-machine (IHM).

## 1.4 Communication dans un système automatisé

Pour s'ajuster à l'évolution continue d'un processus réparti entre plusieurs unités de production, un système automatisé doit être mis en liaison avec ses composants afin de permettre des échanges d'information de différents niveaux. Un système de production automatisé utilise à la Figure 0-3 niveaux de commande assurant des fonctions définies ; ce sont les niveaux de commande 0, 1 et 2.

- Le **niveau 0** permet l'acquisition d'information et la commande des actions de la partie opérative. Les capteurs et les pré-actionneurs tout ou rien sont reliés à l'automate programmable par une liaison fil-à-fil à fil.
- Le **niveau 1** permet la **coordination** au moyen d'échanges d'information entre les éléments de traitement. Un câble d'interconnexion raccorde les automates les uns aux autres par 'intermédiaire de connecteurs prévus à cette fin. Lorsque l'on branche un terminal de programmation sur le réseau, on peut avoir un accès direct aux programmes résidant dans les différents automates. Dans les applications industrielles, on distingue 2 types de réseaux. Les **réseaux homogènes** sont constitués par des automates programmables de même nature et compatibles.

Les **réseaux hétérogènes** utilisent des éléments interconnectés de nature différente (automates programmables, machines à commande numérique, robots). La figure 1-2 montre des liaisons par réseaux utilisant un réseau homogène et un réseau hétérogène [2].

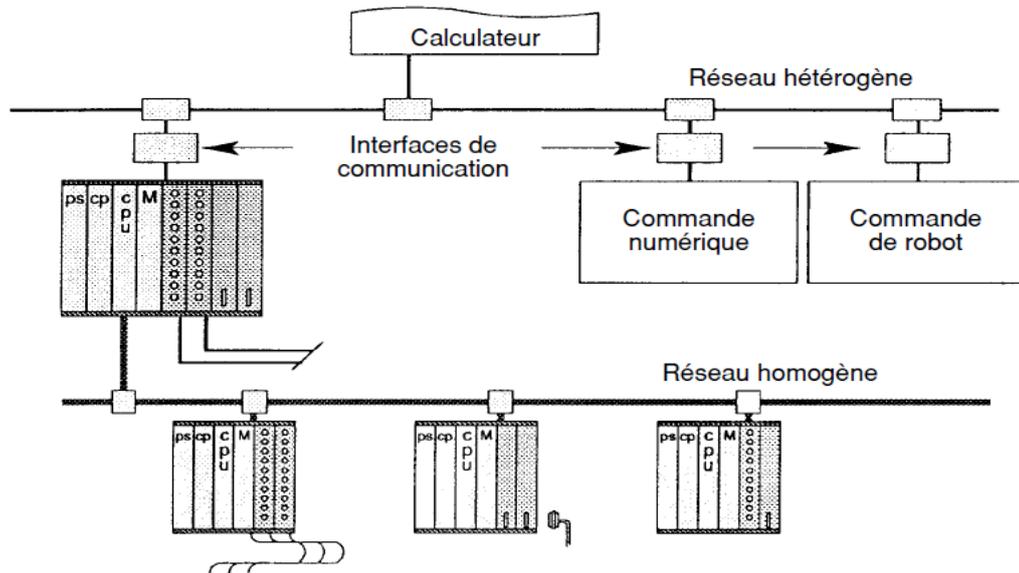


Figure 0-2 : l'échange d'information dans les différents niveaux.

- Le niveau 2 permet la supervision des différentes unités de production. On y assure la centralisation de l'information et le dialogue avec d'autres organes de communication. Le dialogue de supervision assure la conduite du système de production. Il permet :
  - De lancer des campagnes de production ;
  - d'effectuer des réglages ;
  - d'afficher des messages ;
  - de gérer les défauts ;
  - de produire des statistiques sur la production ;
  - de gérer l'entretien préventif.

La figure 1.4 résume les différents niveaux de commande en identifiant les liaisons pour chacun de ces niveaux [2] :

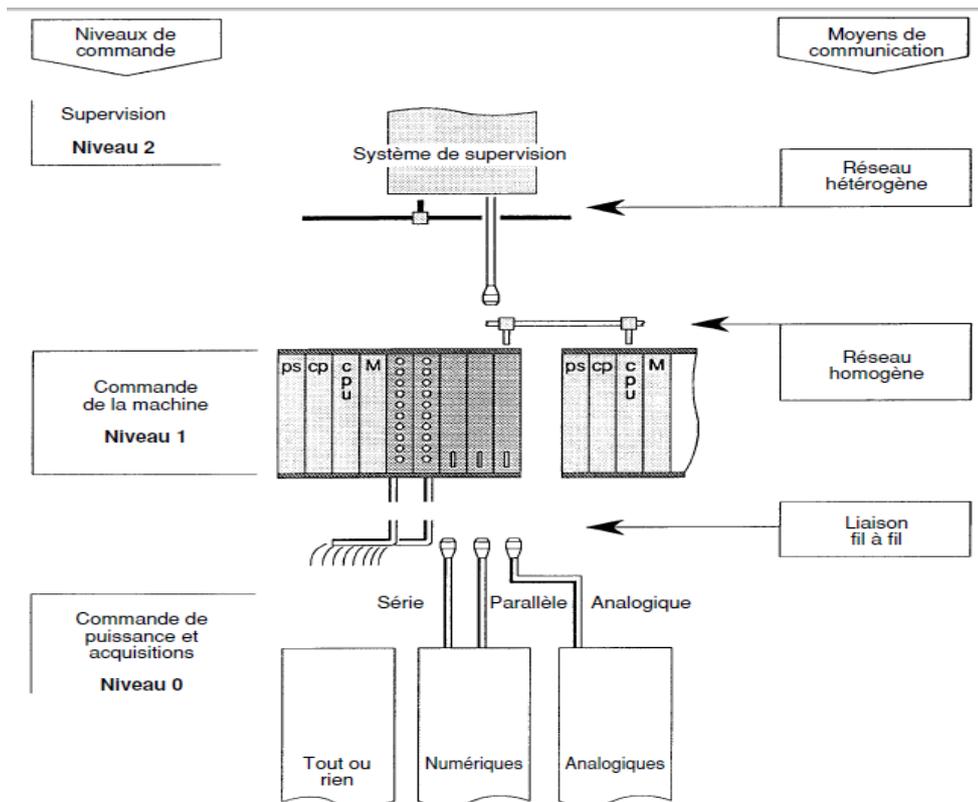


Figure 0-3 : Niveaux de commande.

## 1.5 Connaissance de matériel

### 1.5.1 Pré-actionneur

Son rôle est de distribuer, sur ordre de la partie commande, l'énergie utile aux actionneurs [4].



Figure 0-4 : différents types de pré-actionneurs.

### 1.5.2 Actionneur

Dans une machine, un actionneur est une chose qui transforme l'énergie qui lui est fournie en un phénomène physique qui fournit un travail, modifie le comportement ou l'état d'un système.



Figure 0-5: différents types des actionneurs.

### 1.5.3 Transmetteur

Son rôle est d'adapter et de transmettre l'énergie mécanique délivrée par l'actionneur pour la rendre utilisable par l'effecteur. [4]

➤ **Sans transformation de mouvement :**



Figure 0-6: Transmetteurs Sans transformation de mouvement

Et aussi : pignon-chaine, roue et vis sans fin ....

➤ **avec transformation de mouvement :**



Figure 0-7: Transmetteurs avec transformation de mouvement

Et aussi : came, bielle-manivelle, croix de malte...

### 1.5.4 Effecteur

Son rôle est d'effectuer la transformation de la matière d'œuvre afin de lui apporter sa valeur ajoutée.

Exemple : doigts d'une pince, tapis roulant, outil d'un centre d'usinage, ventouse ou électro-aimant d'un système de préhension ...

Systeme	Effecteur
 <p>ventilateur</p>	 <p>pale de ventilateur</p>

Figure 0-8: Effecteurs. [9]

### 1.5.5 Capteur

Son rôle est de prélever une grandeur physique et d'en produire une image exploitable par la partie commande. [4]



Figure 0-9: capteurs.

Et aussi : capteur d'accélération, capteur de température....

### 1.5.6 Interface H / M

Son rôle est de traduire la consigne d'un utilisateur en une image exploitable par la partie commande.



Figure 0-10: les interfaces H / M.

### 1.5.7 Partie commande

Son rôle est, à l'aide du programme implanté, de traiter les informations en provenance des capteurs et de l'interface H / M afin d'émettre les ordres destinés aux pré-actionneurs des différentes chaînes d'énergie.

Elle envoie aussi des signalisations à l'interface M / H qui seront traduit en signaux lumineux et / ou sonores à destination de l'opérateur. [4]



Figure 0-11: partie commande.

### 1.5.8 Interface H / M

Son rôle est de permettre à l'opérateur d'être informé sur l'état du système.



Figure 0-12: interface M / H.

## 1.6 Objectifs de l'automatisation

La compétitivité de l'entreprise et des produits. Cette compétitivité passe par la qualité, la maîtrise des coûts et l'innovation. Cela induit une disponibilité à tous les niveaux. On cherche donc à améliorer la productivité. L'amélioration des conditions de travail, et surtout la sécurité, font partie des objectifs de l'automatisation [1].

Les buts (ou objectifs) de l'automatisation sont donc :

- ✓ Simplifier le travail de l'homme,
- ✓ Augmenter la sécurité (responsabilité),
- ✓ Accroître la productivité.

C'est également :

- ✓ Economiser les matières premières et l'énergie,
- ✓ S'adapter à des contextes particuliers : flexibilité,
- ✓ Améliorer la qualité...

Le système automatisé permet d'obtenir de façon reproductible la valeur ajoutée.

Moyens : Les moyens permettant d'atteindre ces objectifs sont d'ordres financiers, matériels, organisationnels, et humains. On distingue notamment : informatiser, automatiser, robotiser, rationaliser, organiser, Etc.,

## 1.7 Conséquences de l'automatisation

### 1.7.1 Avantages de l'automatisation

- ✓ **La réduction des coûts de production** : un retour sur investissement rapide permet de compenser les coûts initiaux d'installation.

- ✓ **L'optimisation des temps de cycle des pièces** : une ligne de production allégée est cruciale dans tout effort visant à accroître l'efficacité. Les robots peuvent fonctionner plus longtemps et plus rapidement, ce qui augmente les taux de production.
- ✓ **L'amélioration de la qualité et de la fiabilité** : les tâches effectuées dans une ligne de production automatisée sont précises et reproductibles à l'identique, ce qui assure que tous les produits sont fabriqués avec les mêmes spécifications et qu'ils auront la même qualité. Les réparations sont espacées et peu nombreuses.
- ✓ **Une meilleure utilisation de l'espace au sein de l'unité de production** : l'automatisant des équipements et des machines intégrés dans votre ligne de production pourront vous faire gagner beaucoup d'espaces et vous permettent de rendre le flux du processus plus efficace.
- ✓ **La réduction des déchets** : les robots sont si précis qu'ils peuvent économiser de la matière première, ce qui diminue les coûts de traitement des déchets.
- ✓ **La préservation des emplois locaux** : au lieu de délocaliser votre production vers des pays « *low cost* » en termes de main-d'œuvre, vous pouvez intégrer l'automatisation dans quelques étapes clés du processus de la production où, par exemple, la précision d'un robot est essentielle. Cela vous permettra d'accroître la qualité de vos produits et d'augmenter vos marges de profit afin que vous puissiez garder votre entreprise dans son emplacement actuel.
- ✓ **Demeurer compétitif** : la réduction des délais de livraison et les coûts de production sont des éléments qui attirent les clients. L'automatisation permet, justement, d'attendre un rapport qualité-prix/prix très avantageux.
- ✓ **Augmentation de la productivité et maintien de la qualité.**

### 1.7.2 Inconvénients de l'automatisation

- ✓ 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.
- ✓ Pannes.
- ✓ Consommation d'énergie.
- ✓ Formation d'un personnel plus qualifié (technicien de maintenance, de contrôle...). [10]

## **1.8 Conclusion**

Les systèmes automatisés ont été utilisés pour remplacer l'homme et sur tout dans des opérations dangereuses répétitives ou pénibles, ils sont partout dans notre environnement, et sont tellement nombreux et différents, en effet ils ont tous comme point commun de pouvoir accomplir une tâche sans la présence d'un humain.

Dans ce chapitre on a donné une description des systèmes automatisés et de leurs différentes parties, la décomposition des systèmes et l'explication de leurs fonctionnements, et enfin on a présenté les objectifs et les conséquences de l'automatisation.

---

# **Chapitre 2 :**

# **Les Automates Program-**

# **mables Industriels**

## 2.1 Introduction

Les automates programmables industriels sont apparus à la fin des années soixante, à la demande de l'industrie automobile américaine (GM), qui réclamait plus d'adaptabilité de leurs systèmes de commande.

L'Automate Programmable Industriel (API) est un appareil électronique programmable adapté à l'environnement industriel. Il réalise des fonctions d'automatisme pour assurer la commande de pré actionneurs et d'actionneurs à partir des informations logiques, analogiques ou numériques.

On nomme Automate Programmable Industriel, API (*en anglais Programmable Logic Controller, PLC*) un type particulier d'ordinateur, robuste et réactif, ayant des entrées et des sorties physiques, utilisé pour automatiser des processus comme la commande des machines sur une ligne de montage dans une usine, ou le pilotage de systèmes de manutention automatique. Là où les systèmes automatisés plus anciens employaient des centaines ou des milliers de relais et de cames, un simple automate suffit. On nomme automaticiens les programmeurs de ces Automates Programmables Industriels.

## 2.2 Définition

Un automate programmable industriel est un dispositif, une machine, un appareil électronique qui comporte une mémoire programmable par un utilisateur automaticien non-informaticien,

Destiné pour automatiser et piloter les commandes des processus en ambiance industrielle comme la commande des machines sur une chaîne de montage dans une usine en temps réel par traitement séquentiel des procédés ou parties opératives. Un automate programmable est adaptable à un maximum d'application, est organisé pour dialoguer simplement et rationnellement avec les constituants de tous ordres répartis sur la machine par leur langage spécialisé, leur programmation relevait du technicien en électronique industrielle.

À l'aide d'un langage adapté, pour le stockage interne des instructions composant les fonctions d'automatismes, par exemple :

- Logique séquentielle et combinatoire
- Temporisation
- Comptage, décomptage, comparaison
- Calcul arithmétique
- Réglage, asservissement, régulation,

Il envoie des ordres vers les *pré-actionneurs* (partie opérative ou *PO*), il permet de commander, mesurer et contrôler au moyen de modules d'entrées et de sorties (logiques, numériques ou analogiques) différentes sortes de machines ou de processus en environnement industriel.

## 2.3 Architecture general d'un automate programmable industriel

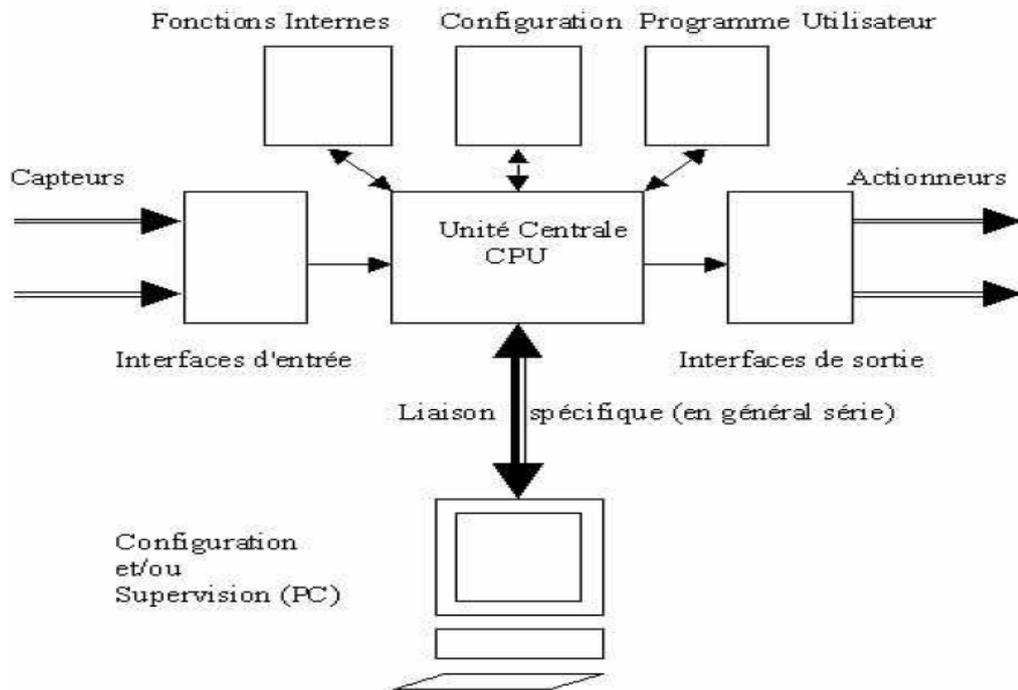


Figure 0-1: Structure général d'un API.

### 2.3.1 Unité centrale

Elle gère l'ensemble du processus, elle contient le processeur, les mémoires vives et des mémoires mortes pour une taille débutant à 40 K octets. Elle est programmable directement par console ou par le biais d'une liaison série et d'un logiciel adapté. Cette CPU peut être en RUN (elle exécute le programme) ou en STOP (toutes les sorties sont au repos, contacts ouverts).

### 2.3.2 Configuration

Elle contient les paramètres liés à la structure de l'API et à la structure du réseau informatique.

### 2.3.3 Fonctions Internes

Ce sont des fonctions préprogrammées livrées avec l'API qui permettent par exemple d'assurer des temporisations, des régulations. Ces fonctions peuvent être résidentes dans l'automate ou disponibles dans le logiciel de programmation.

### 2.3.4 Programme Utilisateur

C'est la loi de commande, il assure la gestion des sorties en fonction de l'état des entrées et éventuellement du temps. Ce programme est exécuté sous forme cyclique par l'API, le temps de cycle est dépendant de la taille du programme et ne doit pas excéder la centaine de millisecondes.

### 2.3.5 Supervision

C'est un ordinateur standard. Il contient le logiciel de programmation (Orphée pour April et Step7 pour Siemens). Ce logiciel permet d'écrire le programme, de le compiler et de le transférer à l'automate. L'ordinateur peut également servir de poste opérateur pour assurer la conduite de l'unité. Un autre logiciel est alors nécessaire pour assurer le dialogue avec l'automate et une interface opérateur conviviale. Si la liaison entre le PC et la CPU est rompue, l'API continue de dérouler son programme.

### 2.3.6 Interfaces

Elles assurent le lien avec le procédé. Ces interfaces peuvent alimenter les boucles d'entrées ou de sorties, dans ce cas, l'automate sera doté d'une alimentation 24V continue. Elles peuvent être garnies de contacts secs, dans ce cas, une alimentation extérieure devra être intégrée aux boucles d'entrée et de sortie. [7]

## 2.4 La structure interne d'un API

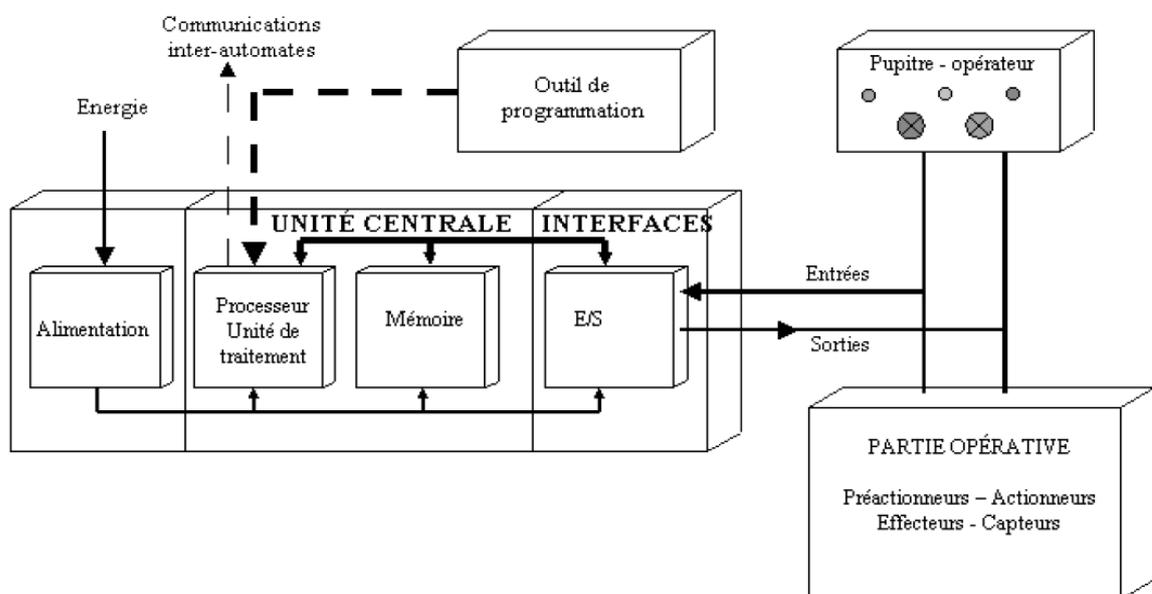


Figure 0-2 : Structure interne d'un A P I.

Un automate programmable est constitué essentiellement de 5 modules [7] :

### 2.4.1 L'unité centrale

L'unité centrale représente le cœur de la machine, et comprend le regroupement du processeur et de la mémoire centrale. Elle commande l'interprétation et l'exécution des instructions programmées. Les instructions sont effectuées les unes après les autres, séquencées par une horloge.

➤ **Le processeur** : Un processeur est l'unité fonctionnelle capable d'interpréter et d'exécuter les instructions du programme. Dans un API le processeur gère l'ensemble des échanges informationnels en assurant :

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

Pour réaliser ces différentes fonctions, le processeur se compose :

- d'une Unité Logique (UL) qui traite les opérations logiques ET, OU et Négation.
- d'une Unité Arithmétique et Logique (UAL) qui traite les opérations de temporisation, de comptage et de calcul.
- d'un Accumulateur qui est un registre de travail dans lequel se range une donnée ou un résultat.
- d'un Registre d'Instruction qui contient, durant le temps de traitement, l'instruction à exécuter.
- d'un Décodeur d'Instruction qui décode l'instruction à exécuter en y associant les microprogrammes de traitement.
- d'un Compteur Programme ou Compteur Ordinal qui contient l'adresse de la prochaine instruction à exécuter et gère ainsi la chronologie de l'exécution des instructions du programme.

➤ **La mémoire** : La mémoire centrale est l'élément fonctionnel qui peut recevoir, conserver et restituer. Elle est découpée en zones où l'on trouve :

- La zone mémoire programme (programme à exécuter) ;
- La zone mémoire des données (état des entrées et des sorties, valeurs des compteurs, temporisations) ;
- Une zone où sont stockés des résultats de calcul utilisés ultérieurement dans le programme ;
- Une zone pour les variables internes.

Ces mémoires peuvent être :

- Durant la phase d'étude et de mise au point du programme :
- des mémoires vives RAM (*Random Access Memory*) volatiles.

- Des mémoires mortes ROM (*read only memory*).
- des mémoires EAROM (*Electrically Alterable Read Only Memory*) non volatiles et effaçables partiellement par voie électrique.
- Durant la phase d'exploitation :
- des mémoires vives RAM qui imposent un dispositif de sauvegarde par batterie rechargeable pour éviter la volatilité de leur contenu en cas de coupure de courant.
- Des mémoires mortes ROM à lecture seulement ou PROM programmables à lecture seulement.
- des mémoires reprogrammables EPROM (*Erasable PROM*) effaçables par un rayonnement ultraviolet et EEPROM (*Electric Erasable PROM*) effaçables électriquement.

## 2.4.2 Le module d'entrées

Un module d'entrées doit permettre à l'Unité centrale de l'automate, d'effectuer une "lecture" de l'état logique des capteurs qui lui sont associés (module 4, 8, 16 ou 32 entrées). • chaque entrée correspond une voie qui traite le signal électrique pour élaborer une information binaire, le bit d'entrée qui est mémorisé. L'ensemble des bits d'entrées forme le "mot" d'entrées. Périodiquement, le Processeur de l'automate programmable vient questionner (adresser) le module : le contenu du mot d'entrées du module est alors recopié dans la mémoire DONNEES de l'automate programmable.

Un module d'entrées est principalement défini par sa modularité (nombre de voies) et les caractéristiques électriques acceptées (tension, nature du courant...).

- **Les cartes d'entrées logiques** : Les cartes d'entrées logiques (cartes d'entrées tout ou rien) permettent de raccorder à l'automate les différents capteurs logiques tels que :
- fins de course
  - capteurs de proximité inductifs ou capacitifs
  - capteurs photoélectriques

Elles assurent l'adaptation, l'isolement, le filtrage et la mise en forme des signaux électriques. Une diode électroluminescente située sur la carte donne l'état de chaque entrée.

- **Les cartes d'entrées analogiques** : Les cartes d'entrées analogiques permettent de gérer des grandeurs analogiques en faisant varier un code numérique au sein du module. Les entrées analogiques disposent d'un seul convertisseur analogique /numérique, elles sont scrutées les unes à la suite des autres par un multiplexeur à relais.

### 2.4.3 Le module de sorties

Un module de sorties permet à l'automate programmable d'agir sur les actionneurs. Il réalise la correspondance : état logique signal électrique. Périodiquement, le processeur adresse le module et provoque l'écriture des bits d'un mot mémoire sur les voies de sorties du module. L'élément de commutation du module est soit électronique (transistors, triac) soit électromécanique (contacts de relais internes au module).

➤ **Les cartes de sorties logiques** : Les cartes de sorties logiques (tout ou rien) permettent de raccorder à l'automate les différents pré-actionneurs tels que :

- Les contacteurs
- Les voyants
- Les distributeurs
- Les afficheurs...

Les tensions de sorties usuelles sont de 5 volts en continu ou de 24, 48, 110, 220 volts en alternatif. Ces cartes possèdent soit des relais, soit des triacs, soit des transistors. L'état de chaque sortie est visualisé par une diode électroluminescente.

➤ **Les cartes de sortie analogiques** : Les cartes de sortie analogiques permettent de gérer des grandeurs analogiques en faisant varier un code numérique au sein du module. Ces modules assurent la conversion numérique/analogique. Les sorties analogiques peuvent posséder un convertisseur par voie. Le nombre de voies sur ces cartes est de 2 ou 4.

### 2.4.4 Le module d'alimentation

Compose de blocs qui permettent de fournir à l'automate l'énergie nécessaire à son fonctionnement. A partir d'une alimentation en 220 volts alternatifs, ces blocs délivrent des sources de tension dont l'automate a besoin : 24V, 12V ou 5V en continu. En règle générale, un voyant positionné sur la façade indique la mise sous tension de l'automate.

### 2.4.5 Le module de communication

Comprend les consoles, les boîtiers de tests et les unités de dialogue en ligne :

➤ **Les consoles** : Il existe deux types de consoles. L'une permet le paramétrage et les relevés d'informations (modification des valeurs, et visualisation), l'autre permet en plus la programmation, le réglage et l'exploitation. Cette dernière dans la phase de programmation effectue :

- L'écriture.
- La modification.

- L'effacement.
- Le transfert d'un programme dans la mémoire de l'automate ou dans une mémoire REPRM.

La console peut également afficher le résultat de l'autotest comprenant l'état des modules d'entrées et de sorties, l'état de la mémoire, de la batterie, etc. Les consoles sont équipées souvent d'un écran à cristaux liquides. Certaines consoles ne peuvent être utilisées que connectées à un automate, d'autres peuvent fonctionner de manière autonome grâce à la mémoire interne et à leur alimentation.

➤ **Les boîtiers de tests** : Destinées aux personnels d'entretien, ils permettent de visualiser le programme

Ou les valeurs des paramètres. Par exemple :

- Affichage de la ligne de programme a contrôlé.
  - Visualisation de l'instruction (code opératoire et adresse de l'opérande).
  - Visualisation de l'état des entrées.
  - Visualisation de l'état des sorties.
- **Les unités de dialogue en ligne** : Elles sont destinées aux personnels spécialistes du procédé et non de l'automate programmable, elle leur permet d'agir sur certains paramètres :
- Modification des constantes, compteurs temporisations.
  - Forçage des entrées/sorties.
  - Exécution de parties de programme.
  - Chargement de programmes en mémoire à partir de cassettes.

## 2.5 Domaines d'utilisation des API

On utilise les API dans tous les secteurs industriels pour la commande des machines (convoyage, emballage ...) ou des chaînes de production (automobile, agroalimentaire...) ou il peut également assurer des fonctions de régulation de processus (métallurgie, chimie...). Il est de plus en plus utilisé dans les domaines du bâtiment (tertiaire et industriel) pour le contrôle du chauffage. De l'éclairage de la sécurité ou des alarmes. [5]

Alors on peut dire que les domaines d'application de l'automate programmable dans :

- Industrie automobile.
- Industrie sidérurgique.
- Industrie papetière et d'emballage.
- Industrie manufacturière.

- Centrales énergétiques et bâtiments.
- Industrie agro-alimentaire.
- Génie procédés.
- Construction mécanique générale.
- Construction des machines spéciales.
- Plasturgie.

### 2.5.1 Exemples de domaines d'application

- **Exemple 1** : L'industrie automobile, pour effectuer l'assemblage et la peinture des carrosseries.

Le processus de fabrication se structure autour de plusieurs flux :

- un flux principal de fabrication de la caisse peinte ;
- une ligne d'assemblage des organes mécaniques et des équipements sur la caisse, à laquelle aboutissent les flux de fabrication des sous-ensembles et des organes.

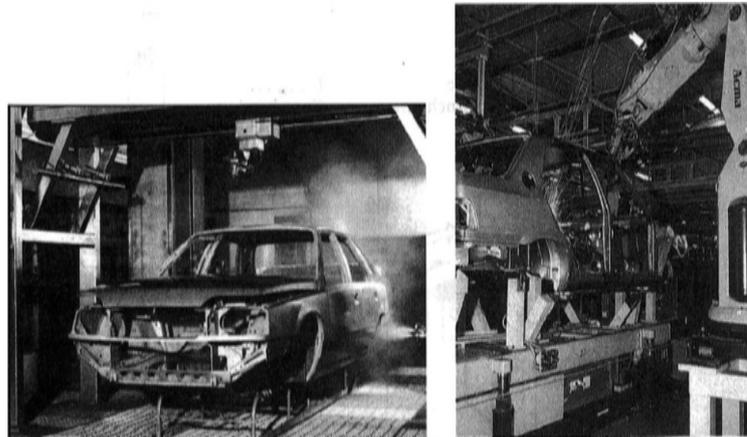


Figure 0-3 : L'industrie d'automobile.

- **Exemple 2 :** Le conditionnement, par ex. le déplacement d'objets suivant un angle quelconque ou le conditionnement sur palette après emballage.

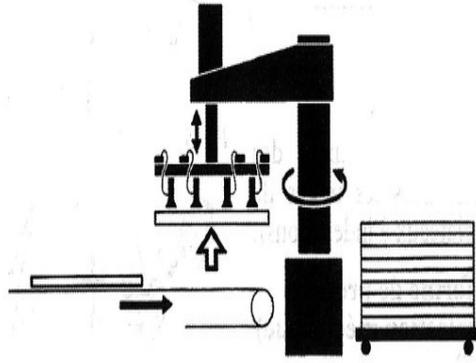


Figure 0-4: appareil d'emballage.

- **Exemple 3 :** L'industrie du bois avec les opérations de débit, de sciage et d'usinage du bois.

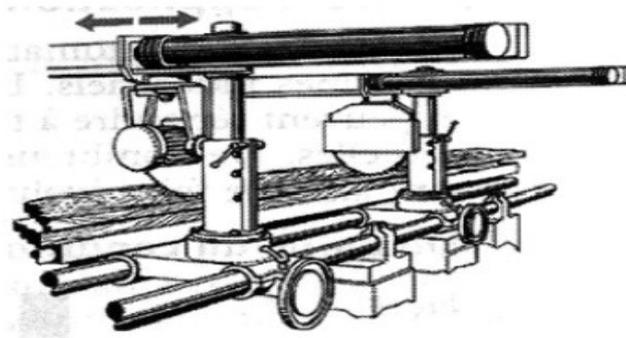


Figure 0-5: l'usinage du bois.

- **Exemple 4 :** Machine-outil : l'automatisation est ici assez importante. L'une des principales applications est dans les unités de perçages.

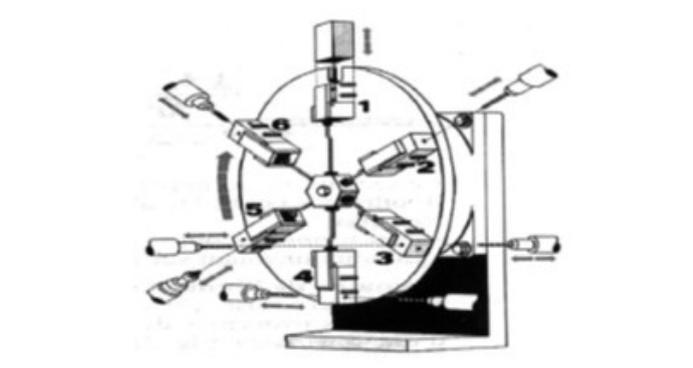


Figure 0-6 : l'unité de perçage.

- **Exemple 5** : Contrôle de produits, détection de défauts en bout de chaîne de production

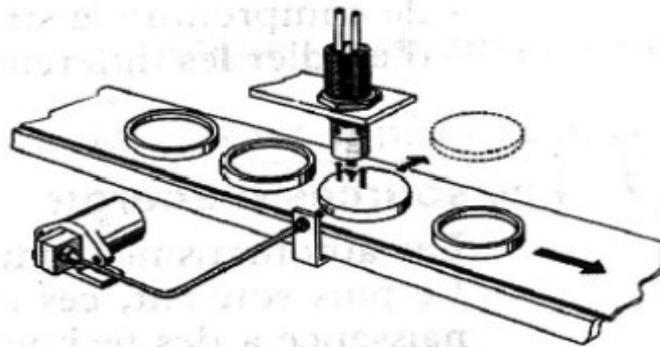


Figure 0-7: machine de détection des défauts.

- **Exemple 6** : Automatisation de services ; Ouvertures programmées de portes et fenêtres, gestion centralisée de bâtiments.

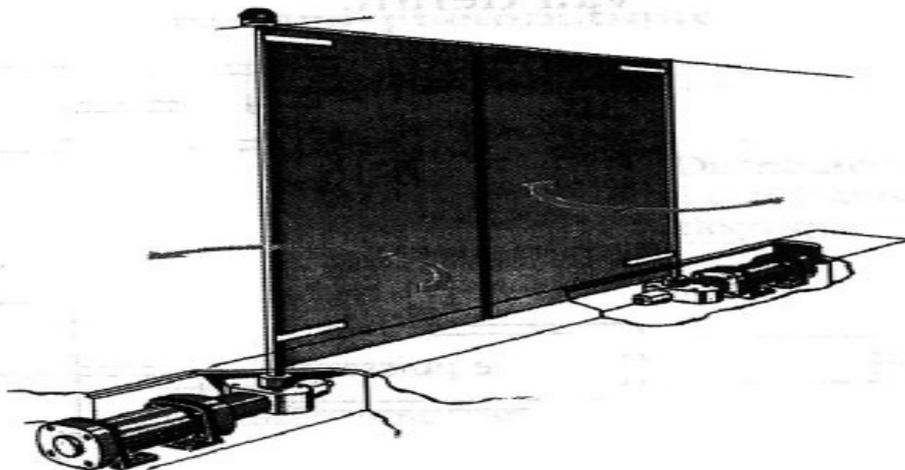


Figure 0-8: portail automatisé.

## 2.6 Langage de programmation

L'écriture d'un programme consiste à créer une liste d'instructions permettant l'exécution des opérations nécessaires au fonctionnement du système. L'API traduit le langage de programmation en langage compréhensible directement par le microprocesseur. Ce langage est propre à chaque constructeur. Il est lié au matériel mis en œuvre.

Chaque automate possède son propre langage. Cependant, les constructeurs proposent tous une interface logicielle répondant à la norme CEI 61131-3 qui définit cinq langages de programmation

utilisables : le grafcet et langage LADDER et langage ST (*Structured Text*) et langage IL (Instruction List) et langage FBD (Boîtes fonctionnelles).

### 2.6.1 Le GRAFCET

Afin de simplifier la programmation des API, on a recours à un système d'écriture par organigramme spécialisé appelé *GRAFCET* (Graphe Fonctionnel de Commande des Étapes et Transitions). Celui-ci est particulièrement adapté pour la programmation des API.

Il se compose de cases correspondantes aux diverses opérations, elles sont reliées entre elles par des traits indiquant le sens de déroulement des opérations. Le passage d'une case à l'autre ne s'effectuant que si l'étape précédente est active et la transition validée.

Ce système très fonctionnel comporte un nombre restreint de symboles conventionnels et permet la correction d'une partie de séquence sans remettre en cause les autres, facilitant ainsi les modifications. De plus ce système étant conventionnel à l'avantage d'être facilement interprété par n'importe quelle personne le connaissant.

Des variantes de programmation peuvent intervenir dans le sens de la réflexion propre à chaque individu, comme pour tout autre langage informatise, qu'il s'agisse d'automates ou d'ordinateurs. [7]

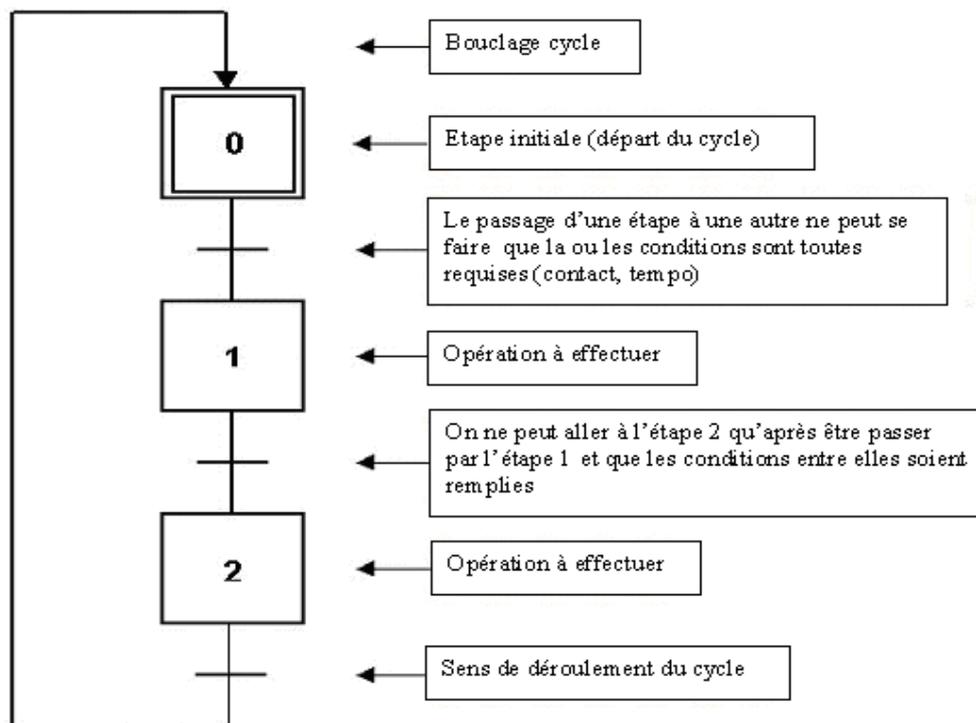


Figure 0-9: Exemple d'un programme en grafcet.

Lorsque le mot GRAFCET en lettre capitale est utilisé, il fait référence à l'outil de modélisation. Lorsque le mot grafcet est écrit en minuscule, il fait alors référence à un modèle obtenu à l'aide des règles du GRAFCET présenté sous forme d'organigramme. Son but est la description du fonctionnement de l'automatisme contrôlant le procédé. C'est tout d'abord un outil graphique puissant directement exploitable. Il est aussi un langage pour la plupart des API existantes sur le marché. Il comprend :

- des étapes associées à des actions,
- des transitions associées à des réceptivités,
- des liaisons orientées reliant étapes et transitions.

#### ▪ **Les règles d'évolution du GRAFCET**

##### ✓ ***Règle N°1 – Condition initiale***

A l'instant initial, seules les étapes initiales sont actives.

##### ✓ ***Règle N°2 – Franchissement d'une transition***

Pour qu'une transition soit validée, il faut que toutes ses étapes amont (immédiatement précédentes reliées à cette transition) soient actives. Le franchissement d'une transition se produit lorsque la transition est validée, si et seulement si la réceptivité associée est vraie.

##### ✓ ***Règle N°3 – Evolution des étapes actives***

Le franchissement d'une transition entraîne obligatoirement dans cet ordre la désactivation de toutes ces étapes amont et l'activation de ses étapes avale.

##### ✓ ***Règle N°4 – Franchissement simultané***

Toutes les transitions simultanément franchissables à un instant donné sont simultanément franchies.

##### ✓ ***Règle N°5 – Conflit d'activation***

Si une étape doit être simultanément désactivée par le franchissement d'une transition avale et activée par le franchissement d'une transition amont, alors elle reste active. On évite ainsi des commandes transitoires (néfastes au procédé) non désirées.

## **2.6.2 Langage Ladder**

Ladder est le mot anglais qui signifie l'échelle. (*Ladder Diagramme (LD)*) ou le langage Ladder ou schéma à contacts est un langage graphique très populaire auprès des automaticiens pour pro-

grammer les automates programmables industriels. Il ressemble un peu aux schémas électriques voir figure 2-10.

L'idée initiale du Ladder est la représentation de fonction logique sous la forme de schémas électriques. Cette représentation est originalement matérielle : quand l'automate programmable industriel n'existait pas, les fonctions étaient réalisées par des câblages.

Le Ladder a été créé et normalisé dans la norme CEI 61131-3. Il est encore aujourd'hui souvent utilisé dans la programmation des automates programmables industriels, bien qu'ayant tendance à être délaissé en faveur de langages plus évolués, et plus adaptés aux techniques modernes de programmation, tels que le (*Structured Text (ST)*) par exemple, ou encore le Grafcet, plus adapté à la programmation de séquences. [8]

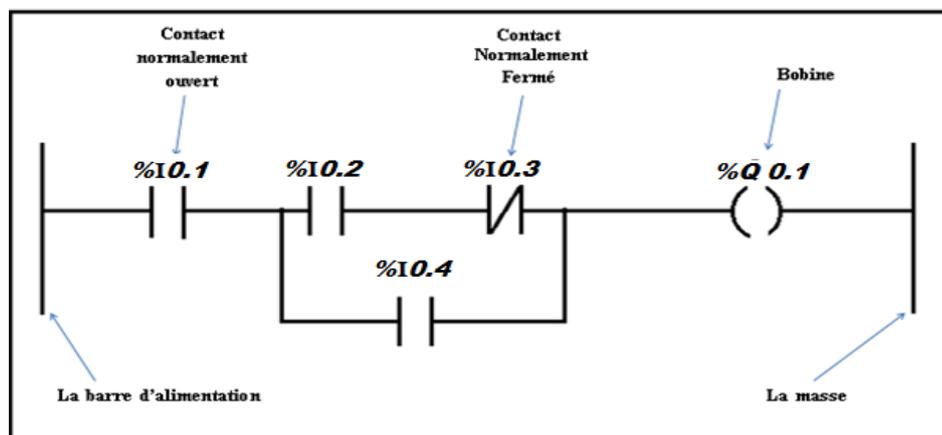


Figure 0-10: Exemple d'un programme en Ladder.

### ▪ Principe

Un programme Ladder se lit de haut en bas et l'évaluation des valeurs se fait de gauche à droite. Les valeurs correspondent en fait, si on le compare à un schéma électrique, à la présence ou non d'un potentiel électrique à chaque nœud de connexion.

En effet, le Ladder est basé sur le principe d'une alimentation en tension représentée par deux traits verticaux reliés horizontalement par des bobines, des contacts et des blocs fonctionnels, d'où le mot 'Ladder' (échelle), (voir. Figure 2-10).

C'est un langage volontairement simple et graphique pour être compréhensible. Il a permis dans les années 1990 d'être utilisé sans lourde formation par un électricien. Il est aujourd'hui un peu dépassé. La puissance de calcul des CPU actuelles permettent de travailler directement en langage objets avec des notions de classe et d'héritage.

▪ **Les composants du langage**

Il existe 3 types d'élément de langage :

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

Le tableau suivant donne les principaux éléments (contacts et bobines) d'un réseau LD. [8]

Tableau 0-1 : les principaux éléments d'un réseau LD.

Objet graphique	nom
-   -	Contact normalement ouvert
- / -	Contact normalement fermé
- P -	Contact fermé au front montant
- N -	Contact fermé au front descendant
-( )-	Bobine normalement ouverte
-( / )-	Bobine normalement fermée
-( S )- ou -( L )-	Bobine Latch (maintenu à 1 une fois actionné)
-( R )- ou -( U )-	Bobine Reset (remise à 0 de la bobine latch)
-( P )-	Bobine active au front montant de son entrée
-( N )-	Bobine active au front descendant de son entrée
<-return>	Retour inconditionnel (vers le sous-programme appelant)
<-cond-<-return>	Retour conditionnel
>>Label	Saut inconditionnel
<-cond->>Label	Saut conditionnel

**2.7 Logiciel de programmation**

Un automate programmable lit des entrées, commande des sorties et résout une logique basée sur un programme. La création d'un programme d'un automate zelio consiste à écrire une série d'instructions rédigées dans un des langages de programmation zelio.

▪ **Zelio soft**

Le module Zelio Soft ou Logic trouve sa place partout où son intelligence pratique simplifie la vie. Dans les petits automatismes pour l'industrie et le résidentiel, l'électricien appréciera sa simpli-

acité de mise en œuvre et sa facilité de programmations grâce à l'utilisation de symboles électriques, Ladder ou Zelio.

Zelio Logic combine dans un même boîtier des fonctions directement utilisables comme des temporisations, des compteurs, des horloges, des relais auxiliaires, combinables à volonté pour réaliser un petit automatisme simple. Grâce à son encombrement modulaire, il s'intègre parfaitement dans un coffret ou dans un tableau de distribution électrique et se programme directement à partir de son interface opérateur intégré ou par un logiciel PC d'une extrême simplicité. [12]

Le langage de programmation à contact permet de faire très simplement la relation entre le schéma électrique et l'automatisme à réaliser. Au-delà d'une application nécessitant deux ou trois temporisations, la solution Zelio Logic est performante en coût d'intégration et de mise en œuvre. Elle offre une solution économique, compacte, flexible et simple à installer, adaptée aux besoins. Il peut se trouver dans l'industrie (machines de contrôle de fin de chaîne, petites machines de conditionnement, d'emballage, machines automatiques) ou dans des installations du bâtiment (gestion de l'éclairage, ventilation, démarrage de moteurs en cascade, compresseurs, petits systèmes de maintenance).

Cette solution apporte un confort supplémentaire dans le réglage des différents paramètres utilisés : réglages des temporisations par saisie numérique, verrouillages de certains réglages par l'installateur, affichage de messages de défauts sur l'interface du Zelio Logic en fonction de l'état de fonctionnement du système par exemple.

#### ▪ **Programmation sur logiciel Zelio Software**

Zelio Software est un logiciel de programmation de type ladder destiné à la programmation des API de marque Schneider que nous allons utiliser dans la réalisation de notre projet. [12]

L'écriture d'un programme sur logiciel Zelio Software consiste à suivre les étapes suivantes :

**Étape 1 :** Ouvrir le logiciel de programmation Zelio Software et créer un nouveau fichier de programmation.

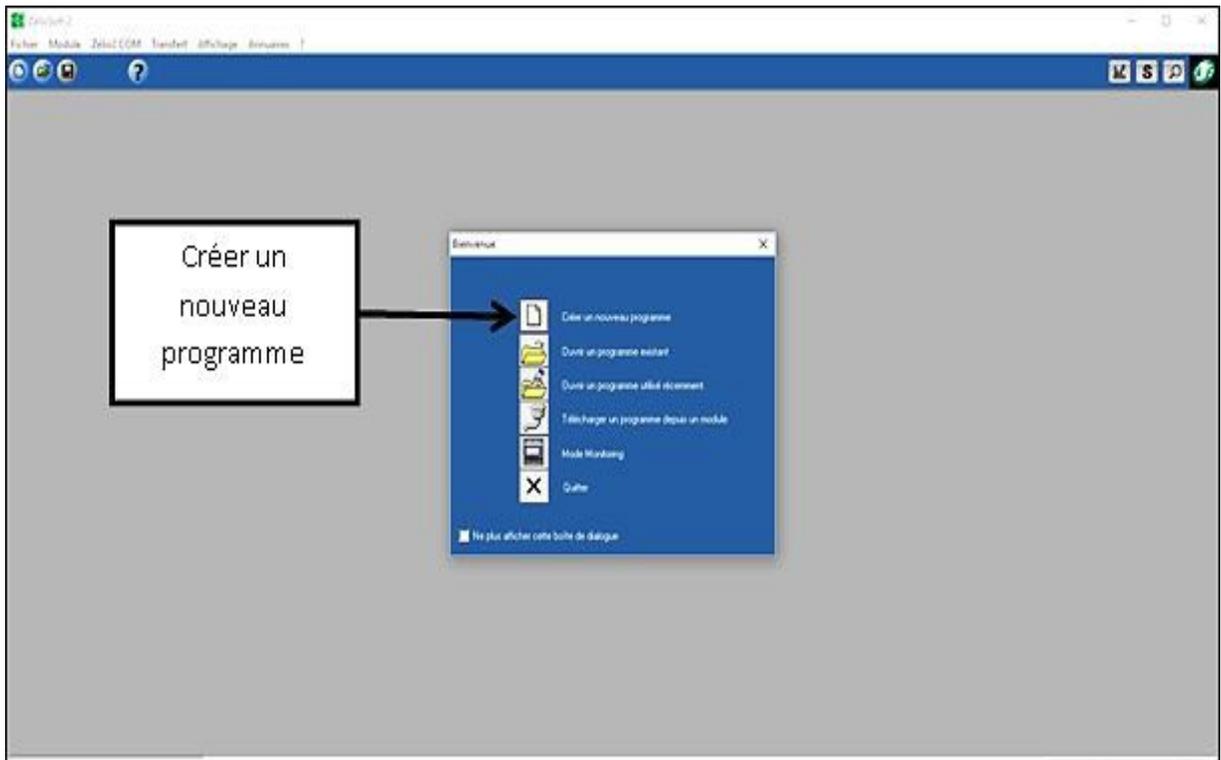


Figure 0-11: création d'un fichier de programmation.

**Etape 2 :** choisir la catégorie d'automates parmi les catégories disponibles.

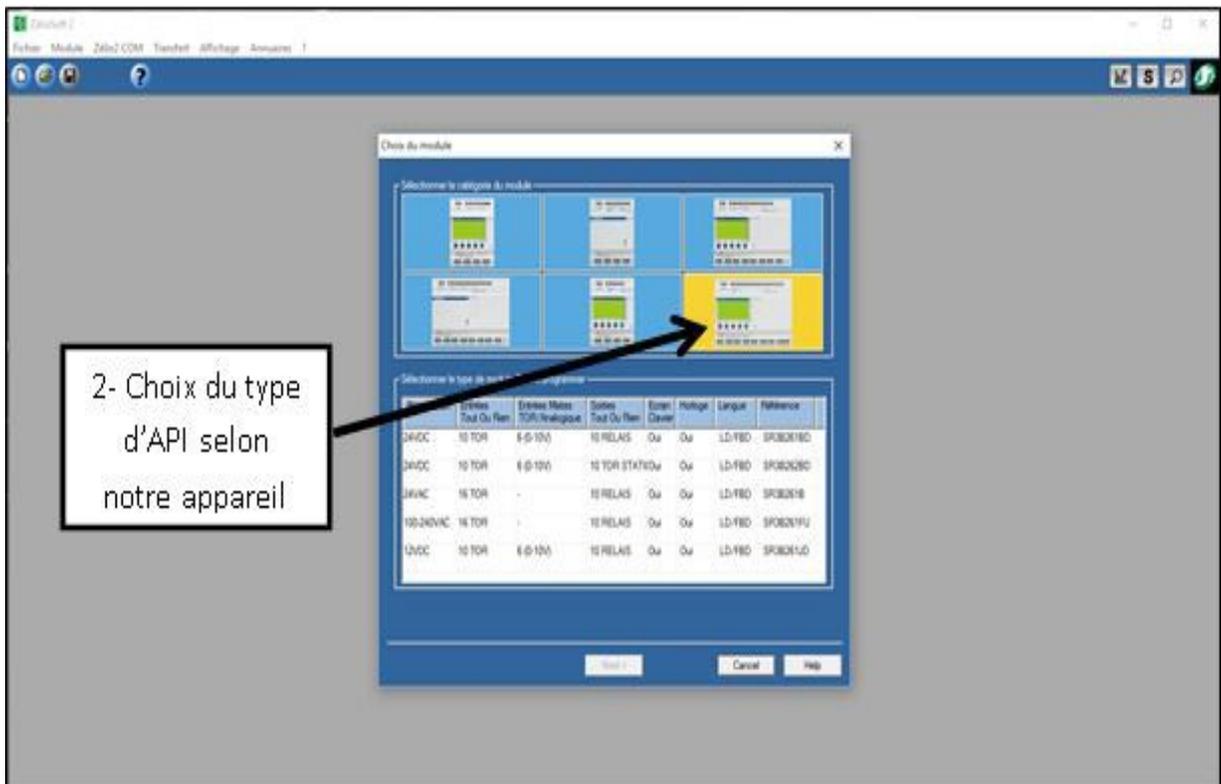


Figure 0-12: choix de la catégorie d'API disponible.



**Etape 5** : choisir le langage de programmation Ladder ou FBD.

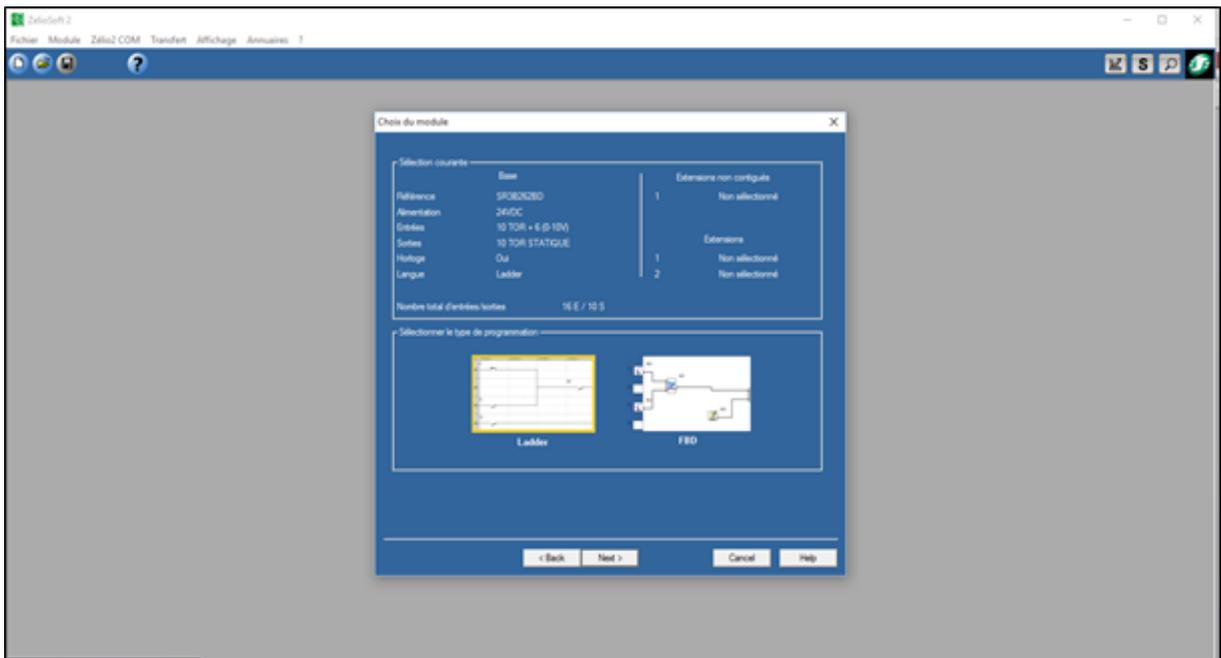


Figure 0-15: langage de programmation choisi.

Une page d'édition de programme Ladder est affichée.

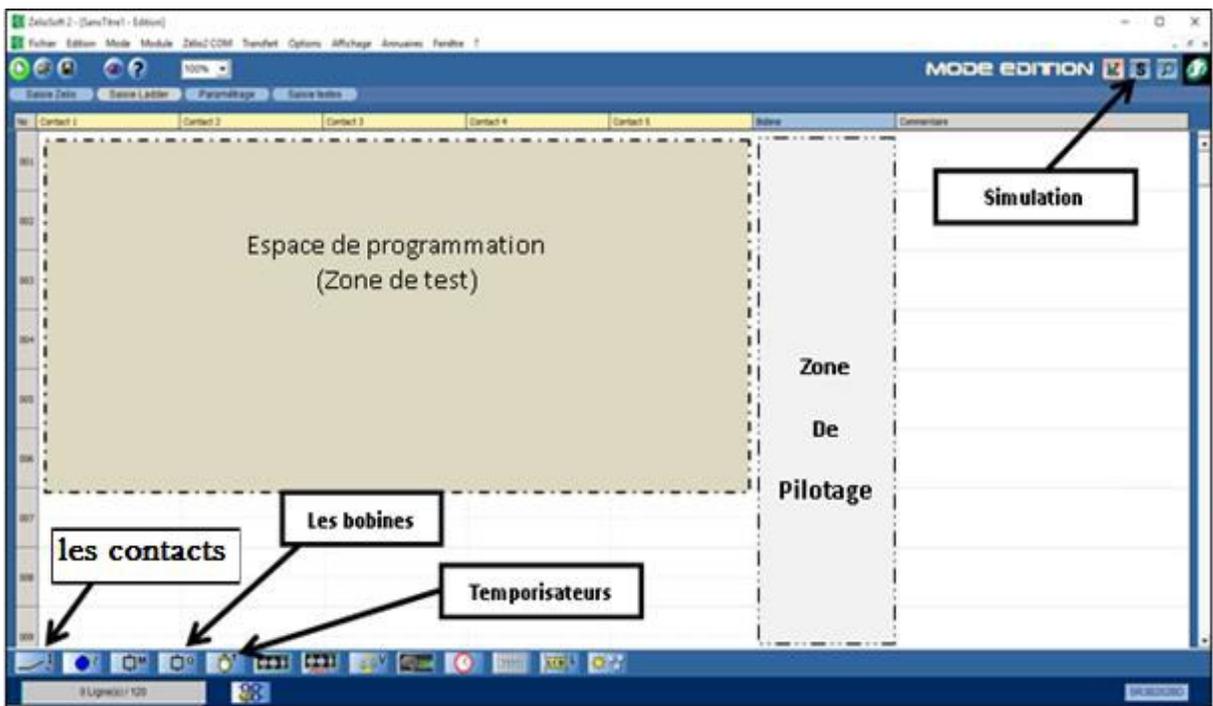


Figure 0-16: page d'édition de programme Ladder.

**Etape 6** : écrire le programme en langage Ladder.

Exemple : marche et arrêt d'un moteur.

Pour ajouter un contact de marche (entrée):

- Cliquer sur le bouton I et choisir un contact.
- Faire glisser le contact choisi dans la zone de test à l'endroit désiré.

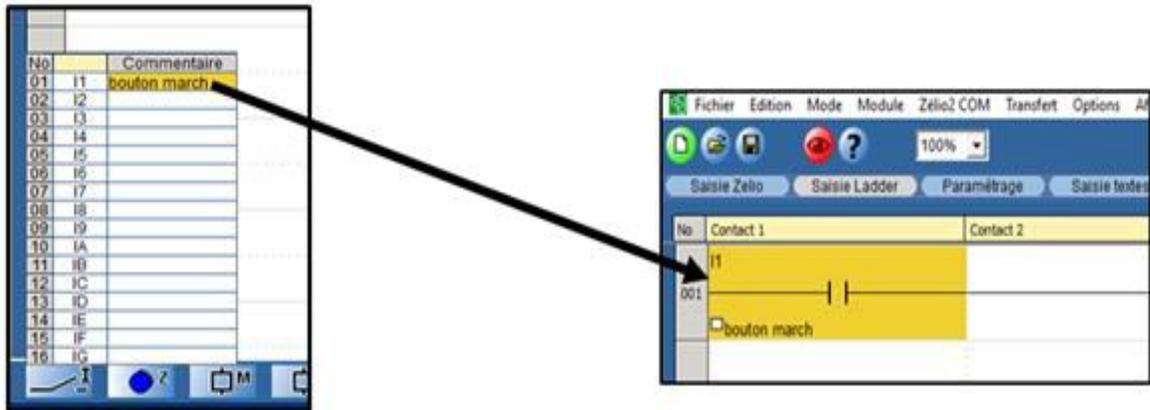


Figure 0-17: Exemple de marche et arrêt d'un moteur.

- Cliquer, par le bouton droit de la souris, sur le contact choisi, pour déterminer l'état fermé ou ouvert du contact au repos, l'état de contact par défaut est ouvert.

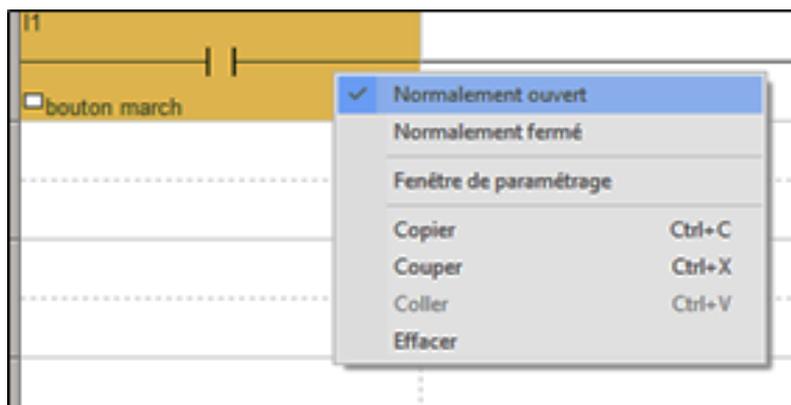


Figure 0-18: détermination l'état du contact.

De la même façon, on ajoute un contact fermé pour l'arrêt.

Ajouter une bobine (sortie) :

- Cliquer sur le bouton Q,
- Glisser la bobine choisie dans la zone de pilotage.

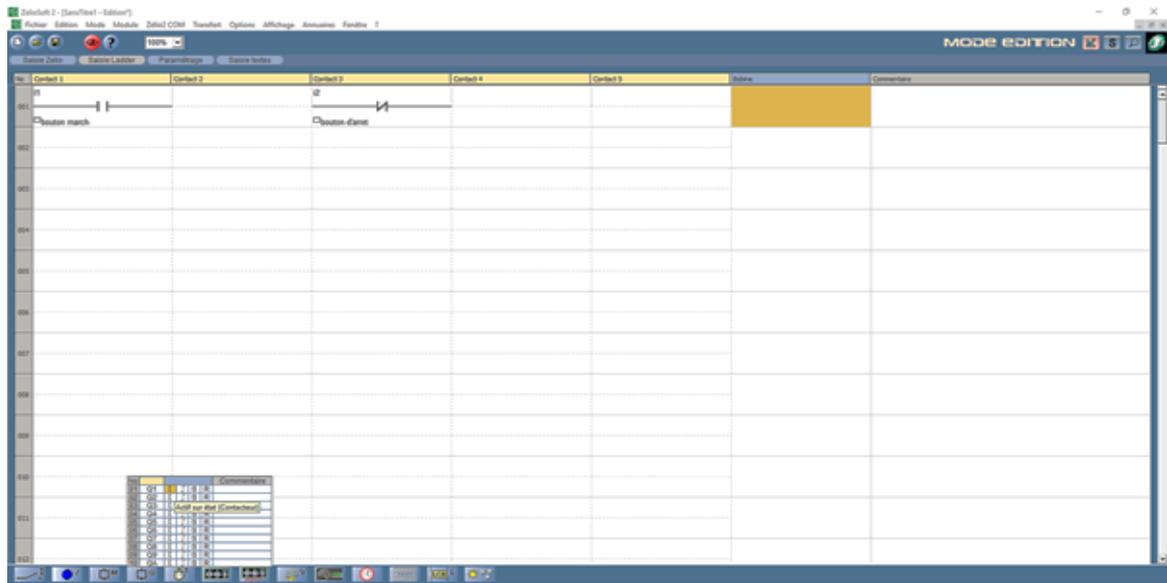


Figure 0-19: insertion d'autres symboles.

Ajouter un contact de maintien, ouvert au repos, en parallèle avec le contact de marche.

Réaliser les liaisons électriques en cliquant sur les tronçons pour relier les différents éléments entre eux. On obtient le schéma Ladder de la figure suivante.

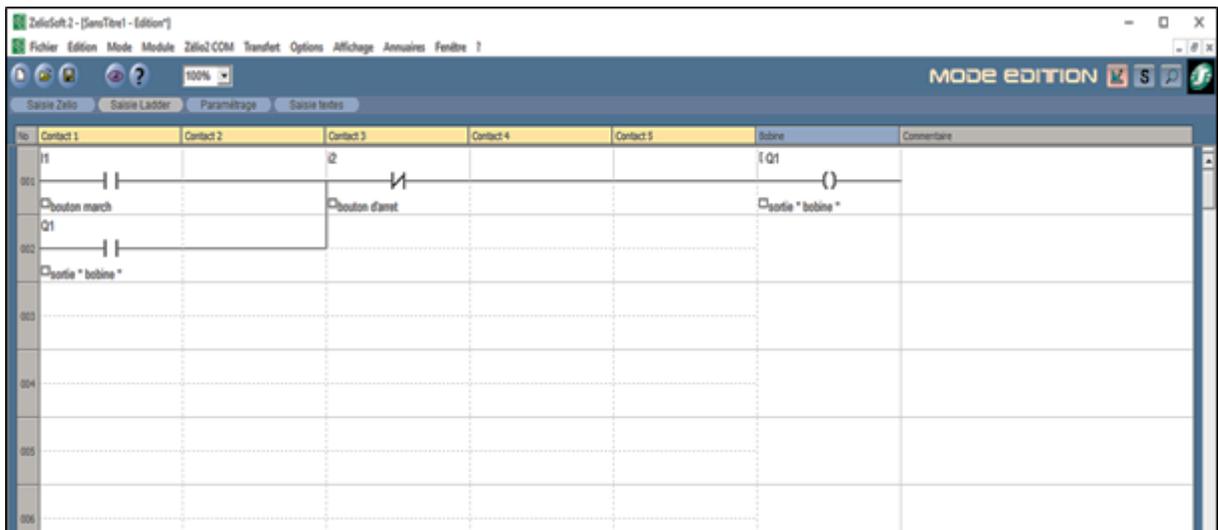


Figure 0-20: réalisation d'un schéma en Ladder.

**Etape 7 : Simulation**

Pour passer en mode simulation, appuyer sur le bouton S.



Pour passer en mode Run, appuyer sur le bouton Run

Cliquer sur le contact de marche pour alimenter la bobine. Le contact de maintien se ferme et la bobine reste alimentée.

Pour arrêter le moteur, cliquer sur le contact d'arrêt.

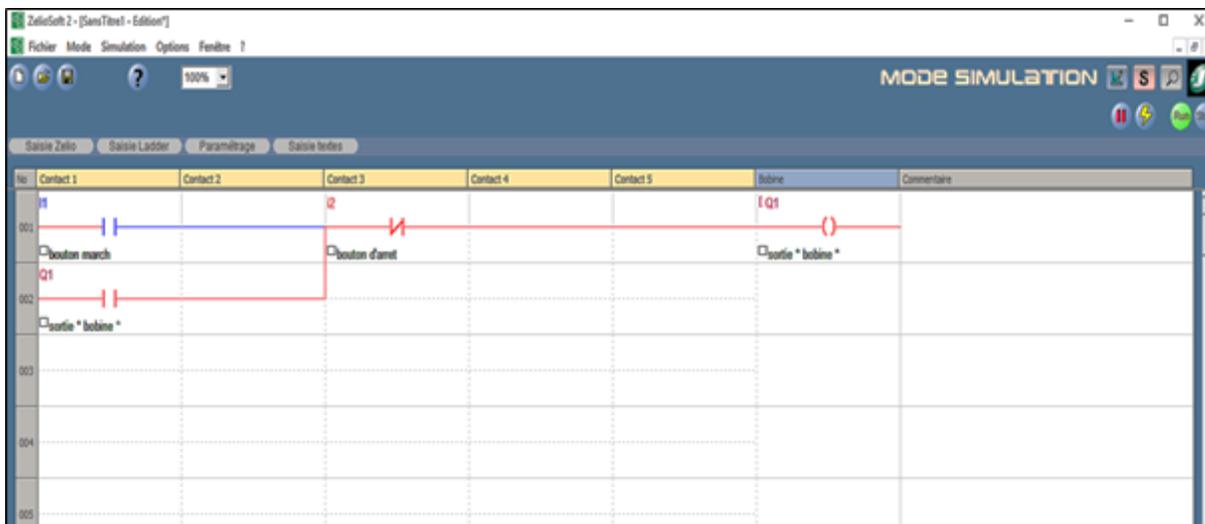


Figure 0-21: simulation d'un programme en Ladder.

**Etape 8 :** Transfert du programme dans le module

On ne peut transférer le programme que si l'on est dans le mode édition.

En mode édition, appuyer sur le bouton édition :



La page suivante est affichée. Appuyer sur le bouton transfert. Choisir l'option transférer programme. Puis choisir l'alternative PC/ Module.

Cliquer sur la touche OK.



Figure 0-22: Transfert du programme dans l'API.

**Etape 9 :** Choisir *mode RUN après chargement* pour démarrer l'automate directement après le chargement du programme dans celui-ci.

Choisir également *Mode monitoring après le chargement* pour visualiser directement sur l'écran du PC l'évolution IN SITU du programme.

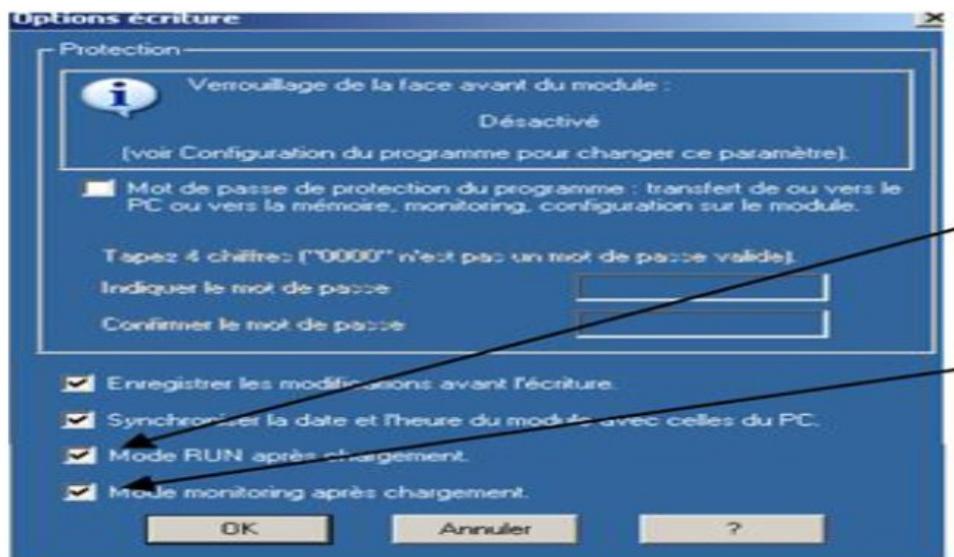


Figure 0-23: mise en marche d'API.

## 2.8 Différents types d'API

Les automates programmables peuvent être de type compact ou modulaire. Dans le type compact, on distingue les modules de programmation (LOGO de Siemens, ZELIO de Schneider, MILLENIUM de Crouzet ...) des micro-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 analogiques) et recevoir des extensions en nombre limité. Ces automates, de fonctionnement simple, sont généralement destinés à la commande de petits automatismes.

Dans le 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 la puissance, la capacité de traitement et la flexibilité sont nécessaires.

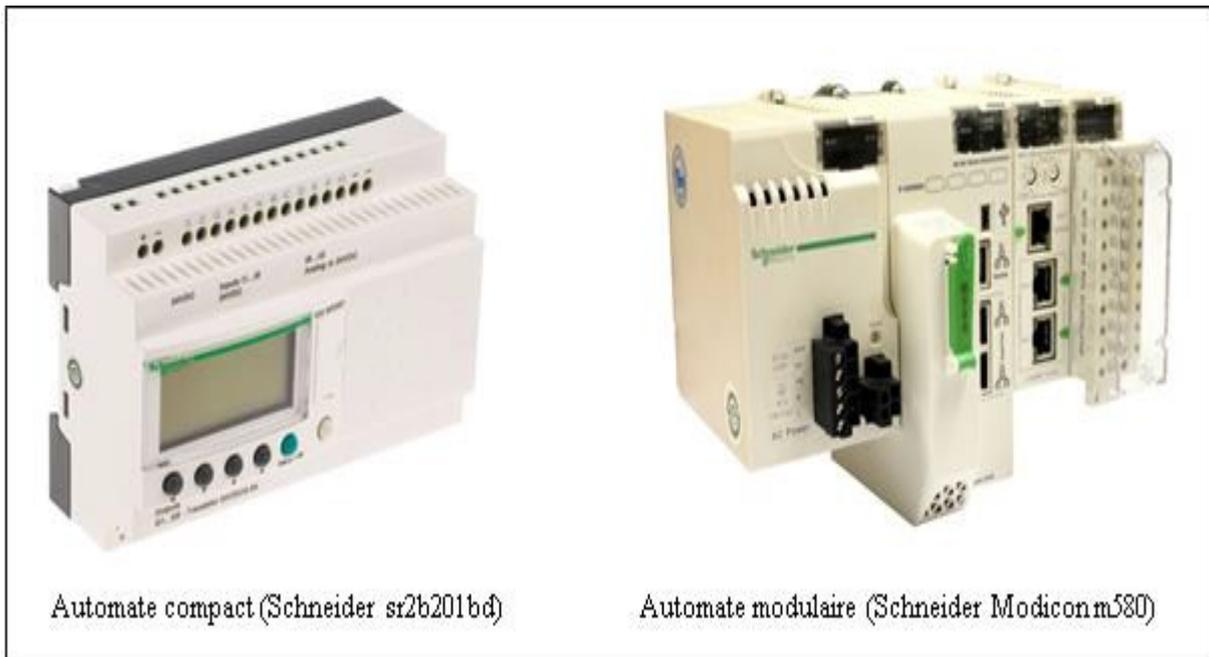


Figure 0-24: Automate programmable industriel.

Les automates compacts permettent de commander des sorties en T.O.R et gèrent par fois 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. [8]

## 2.9 Critères de choix d'un automate

Le choix d'un automate programmable est en premier lieu le choix d'une société ou d'un groupe. Les contacts commerciaux et les expériences vécues sont déjà un point de départ. Les grandes sociétés privilégieront deux fabricants pour faire jouer la concurrence et pouvoir se retourner en cas de perte de vitesse de l'une d'entre elles. Le personnel de maintenance doit toutefois être formé sur ces matériels et une trop grande diversité des matériels peut avoir de graves répercussions.

Un automate utilisant des langages de programmation de type GRAFCET est également préférable pour assurer les mises au point et dépannages dans les meilleures conditions. La possession d'un logiciel de programmation est aussi source d'économies (achat du logiciel et formation du personnel), des outils permettant une simulation des programmes sont également souhaitables. Il faut ensuite quantifier les besoins :

- Nombre d'entrées / sorties : le nombre de cartes peut avoir une incidence sur le nombre de racks dès que le nombre d'entrées / sorties nécessaires devient élevé.

- Type de processeur : la taille mémoire, la vitesse de traitement et les fonctions spéciales offertes par le processeur permettront le choix dans la gamme souvent très étendue.
- 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 ...*). [8]

## 2.10 Sécurité

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

Placé au cœur du système automatisé, l'automate se doit d'être un élément fiable car un dysfonctionnement de celui-ci pourrait avoir de graves répercussions sur la sécurité des personnes, de plus les coûts de réparation et un arrêt de la production peuvent avoir de lourdes conséquences sur le plan financier. Aussi, l'automate fait l'objet de nombreuses dispositions pour assurer la sécurité des biens et des personnes en entreprise.

### **Contraintes extérieures :**

L'automate est conçu pour supporter les différentes contraintes du monde industriel et à fait l'objet de nombreux tests normalisés

### **Coupures d'alimentation :**

L'automate est conçu pour supporter les coupures d'alimentation et permet, par programme, d'assurer un fonctionnement correct lors de la réalimentation (reprises à froid ou à chaud)

### **Mode RUN/STOP :**

Seul un technicien peut mettre en marche ou arrêter un automate et la remise en marche se fait par une procédure d'initialisation (programmée)

### **Contrôles cycliques :**

- Procédures d'autocontrôle des mémoires, de l'horloge, de la batterie, des tensions d'alimentations et des entrées / sorties

- Vérification du temps de scrutation à chaque cycle appelée **Watchdog** (*chien de garde*), et enclenchement d'une procédure d'alarme en cas de dépassement de celui-ci (réglé par l'utilisateur)

### **Visualisation :**

Les automates offrent un écran de visualisation où l'on peut voir l'évolution des entrées / sorties

## **2.11 Conclusion**

Le choix d'un automate dépend de nombre d'entrées/sorties et des autres caractéristiques sou-mentionnées, le dimensionnement des alimentations électriques, les modes d'exécution d'un programme, les limites des divers types de communication, car ce sont des points où sont parfois commises des erreurs qui entraînent des surcoûts d'installation ou limitent fâcheusement les performances obtenues.

Dans ce chapitre nous avons présenté les langages de programmation de l'API à savoir le Grafcet et le Ladder, nous avons illustré la dynamique de Grafcet. Ainsi que les principes du Ladder. Suive d'un exemple illustré de programmation des modes marche/arrêt d'un moteur sous le logiciel Zelio Software de Schneider. Aussi, l'automate fait l'objet de nombreuses dispositions pour assurer la sécurité.

---

# **Chapitre 3 :**

# **Automatisation des processus**

# **industriels**

### **3.1 Introduction**

Les systèmes automatisés remplacent l'homme dans des opérations dangereuses répétitives ou pénibles (robot aspirateur, pilote automatique,.....), et permettent d'augmenter la précision et la productivité.

Dans ce chapitre nous allons présenter un aperçu de ces systèmes. On commencera par l'exposer des cahiers des charges en suite on construira nos système par modélisation graphique (GRAFCET) et en traduisons ces modèles vers le Ladder. Par la suite, nous réalisons notre objectif (gestion de parking).

### **3.2 Exemple 1 : Contrôle continu de bouteilles**

#### **3.2.1 Description**

Remplisseuse est un monobloc complètement automatique de lavage /stérilisation /remplissage /bouchage des bouteilles, disponible en plusieurs modèles de 2 à 8 range .il est caractérisée par :

Changement de format automatique par sélection au clavier.

Becs mobiles de rinçage qui pénètrent dans les bouteilles dans les deux premières sessions de lavage, pour en augmenter l'efficacité.

Remplissage de type volumétrique avec débitmètres et vannes spéciales qui garantissent un écoulement optimale et donc des tempes de remplissage réduits (environ 7 sec par 5 jets).

Brossage extérieur des bouteilles.

Dans la version standard, le cycle de lavage est composé de 6 différentes zones d'arrosage afin de garantir un nettoyage et une stérilisation parfaits de la bouteille.la boucheuse est prévue pour des bouchons à enfoncer (bouchons à visser sur demande).

#### **3.2.2 Cahier des charges fonctionnels**

Dans cet exemple, on met en œuvre un programme de contrôle qui Détecte la position d'une bouteille via un interrupteur de fin de course, attend 0,5 seconde, puis remplit la bouteille jusqu'à ce qu'un CP1 détecte un état rempli.

Après la bouteille est rempli, le programme de contrôle attendra 0,7 seconde avant de passer à la prochaine bouteille.

Le programme comprendra des circuits de démarrage et d'arrêt pour le moteur de sortie et le début du processus. Le tableau montre l'affectation des adresses E / S.

L'illustre la mise en œuvre (Ladder) API de l'application du remplissage des bouteilles.

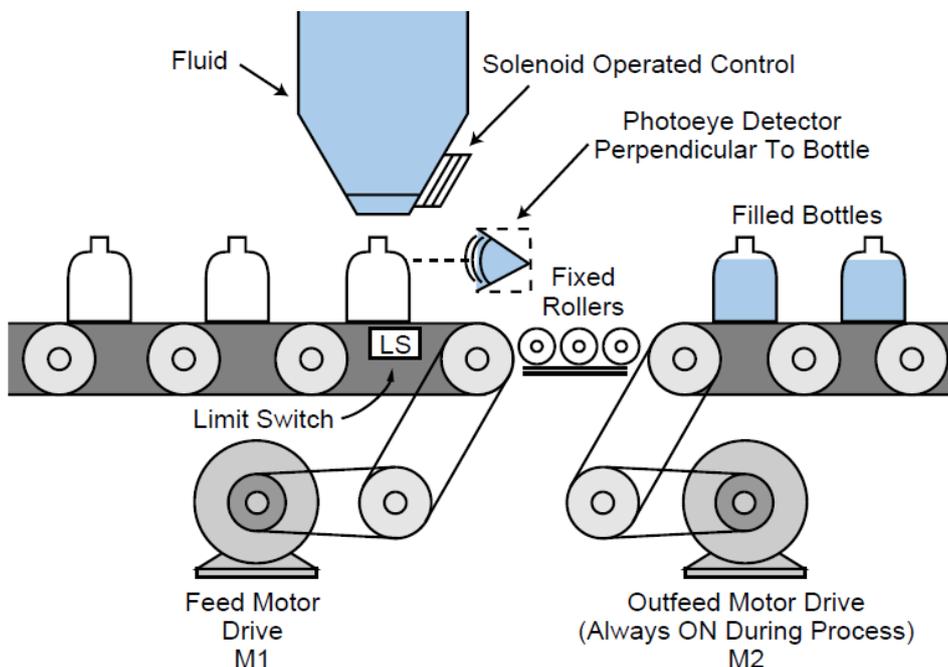


Figure 0-1: Chaîne de remplissage bouteille.

Une fois que le bouton-poussoir de démarrage est pressé, le moteur de sortie (M2) s'allume jusqu'à ce que le bouton poussoir d'arrêté soit pressé.

Le moteur d'alimentation M1 Sera activé une fois le démarrage du système (M2 ON) ; Il s'arrêtera --lorsque le fin cours (LS) détecte une position correcte de la bouteille.

Lorsque la bouteille est en position et 0,5 secondes se sont écoulées, l'électrovanne ouvrira la soupape de remplissage et Reste allumé jusqu'à ce que le capteur niveau (CP1 ou CPE) détecte un niveau approprié.

La bouteille Reste en position pendant 0.7 secondes, alors le capteur RB (bouteille rempli) interne activé va démarrer Le moteur d'alimentation.

Le moteur de fonctionnement reste fonctionné jusqu'à ce que la fin de course détecte Une autre bouteille.

Tableau 0-1: Entrée sortie de la chaîne de remplissage.

Entrée	Description 1	Sortie	Description 2
PB 1	Début processus	M 1	Moteur d'entre
PB 2	Arrête processus	M 2	Moteur de sortie

LS	Position de bouteille	EV	Electro vanne
CP1	Niveau de liquide		

### 3.2.3 GRAFCET

Le grafcet du cycle de fonctionnement est représenté par la Figure 3-2 :

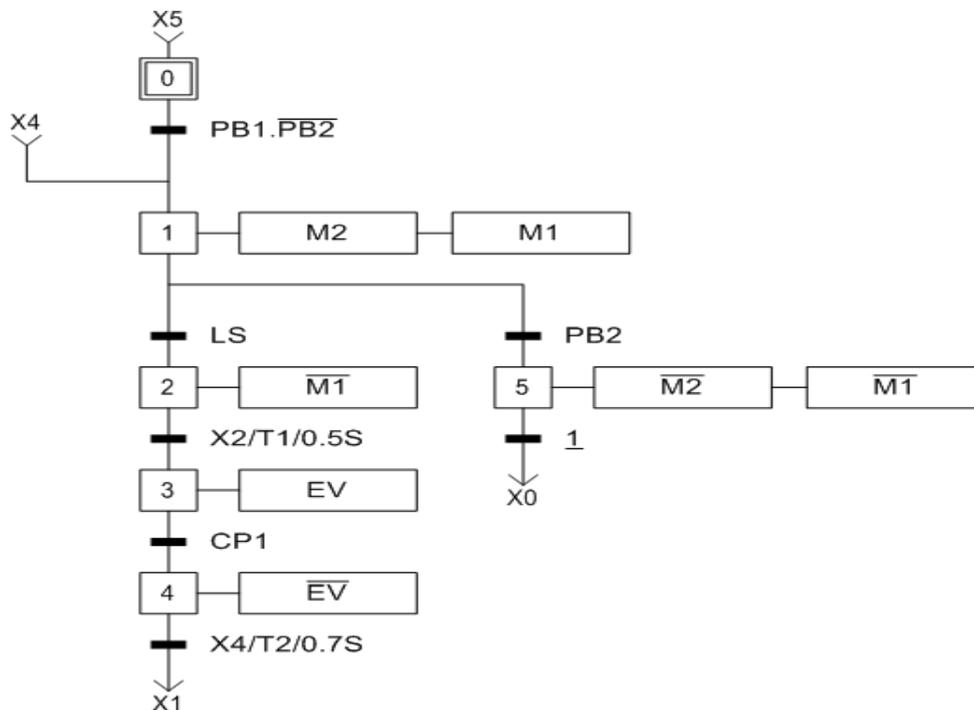


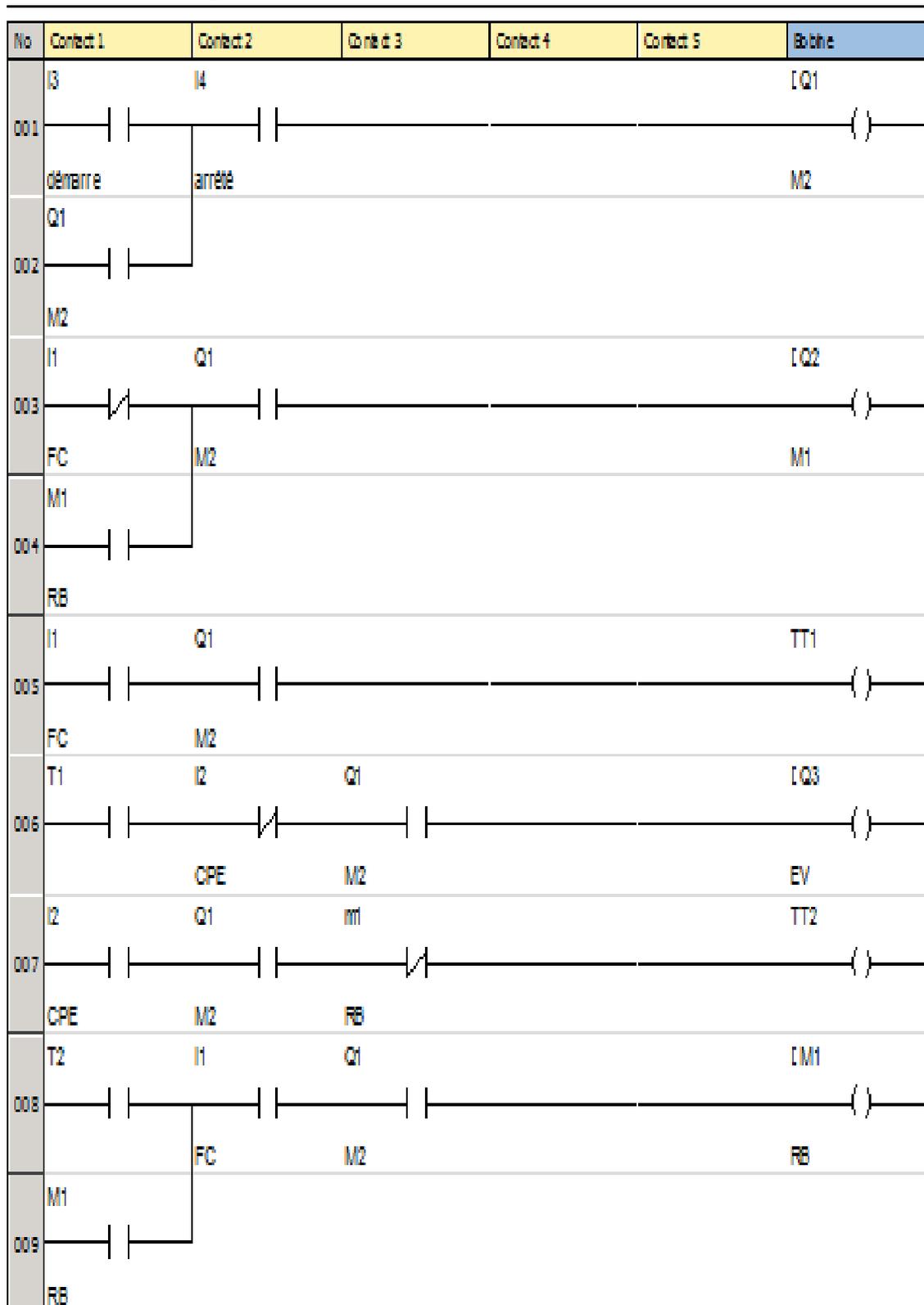
Figure 0-2: grafcet de la chaine de remplissage des bouteilles.

### 3.2.4 Programmation

La programmation des différentes actions du cycle de fonctionnement est effectuée en langage Ladder par le logiciel Zelio Software. Dans ce qui suit, nous présentons le programme complet.

❖ Schéma du programme en langage Ladder

## Schéma du programme



### 3.3 Exemple 2 : Barrière parking

#### 3.3.1 Description

Un parking, ou parc de stationnement, voire simplement stationnement, est un espace ou un bâtiment spécifiquement aménagé pour le stationnement des véhicules. On en trouve le plus souvent à côté des bâtiments publics (gare, aéroport), des lieux de travail, des centres commerciaux ou devant les grandes surfaces pour accueillir les usagers.

A la sortie ou l'ouverture d'un parking, on dispose une barrière automatique qui peut être fermée (barre horizontale) ou ouverte (barre verticale). Cette barrière peut être actionnée soit manuellement par des boutons poussoirs qui commandent l'ouverture et la fermeture, soit automatiquement par un dispositif à carte électronique ou API.

#### 3.3.2 Cahier des charges fonctionnel de Barrière Parking

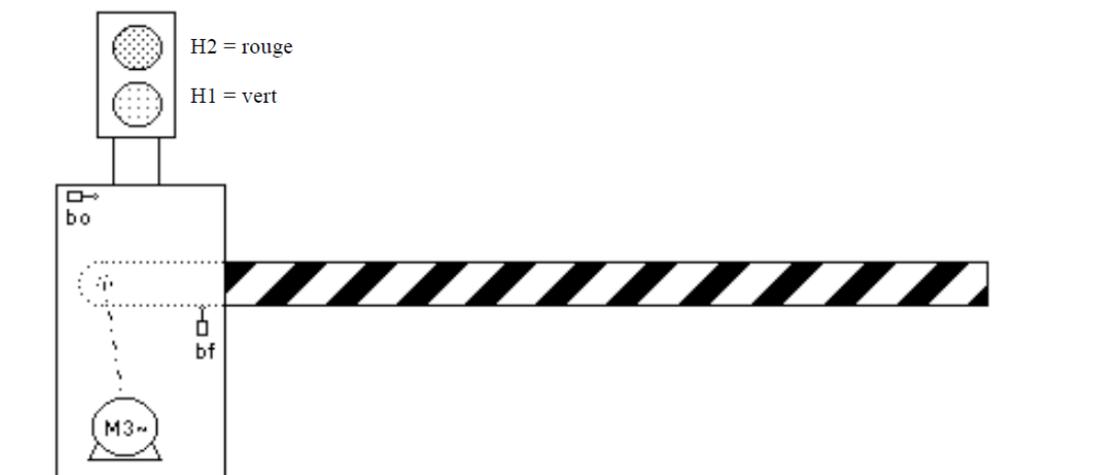


Figure 0-3: La Barrière parking.

En attente de véhicule, la barrière est fermée et le voyant est vert.

Entrée de véhicules dans le parking :

Lorsqu'un véhicule se présente à l'entrée du parking (capteur "cap1" actionné), le system commande l'ouverture de la barrière.

Deux cas se présentent :

➤ **L'automobiliste passe** Après le passage du véhicule.

Dès que le véhicule n'actionne plus le capteur "capteur 1", le feu vert s'éteint.

Dès que "capteur 2 " n'est plus actionné, on comptabilise le véhicule puis la barrière s'abaisse.

➤ **L'automobiliste ne passe pas.**

La barrière doit pouvoir fonctionner quelle que soit la longueur du véhicule (il est sous-entendu que les capteurs "cap1" et "cap2" peuvent être actionnés en même temps, mais également peuvent ne pas être actionnés en même temps dans le cas d'un véhicule de faible longueur).

Une fois cinq véhicules passés il n'est plus possible d'accéder au parking.

Sorties de véhicules du parking :

Lorsqu'un véhicule se présente à la sortie du parking (capteur "cap3" actionné), le conducteur commande l'ouverture de la barrière.

Décomptage de cap4 lors de la présence d'un véhicule et du front descendant de "cap 4".

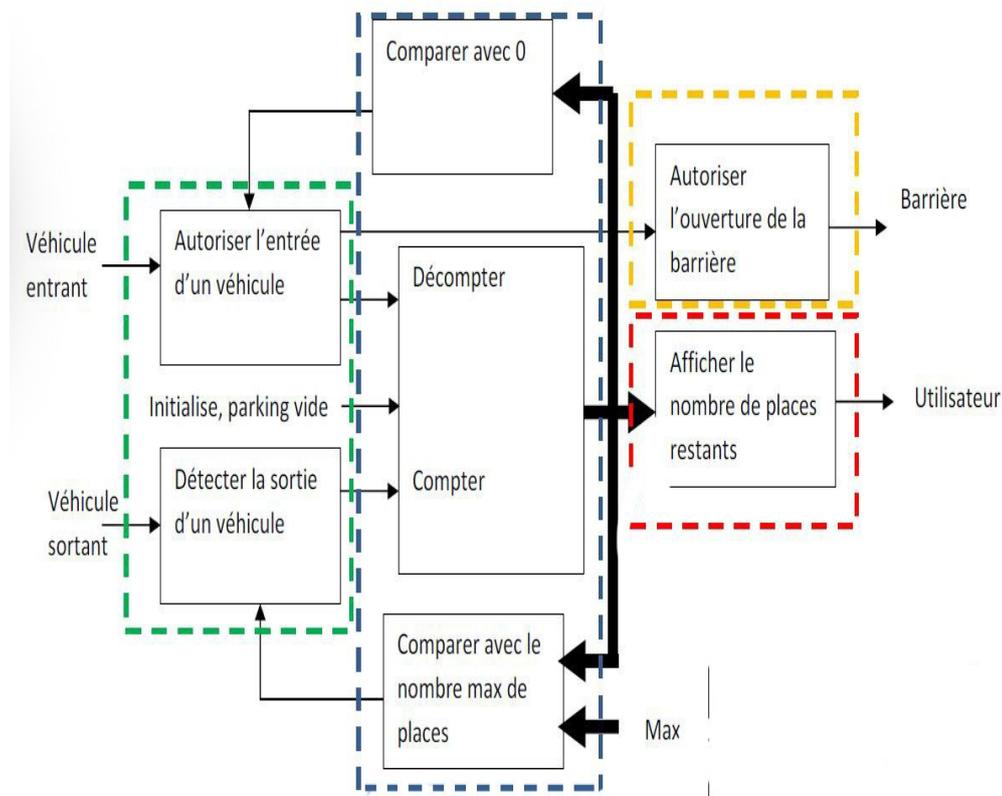


Figure 0-4: Schémas descriptif de gestion de parking.

La mise en service de notre réalisation (la gestion de parking) est montrée respectivement aux figures suivantes :

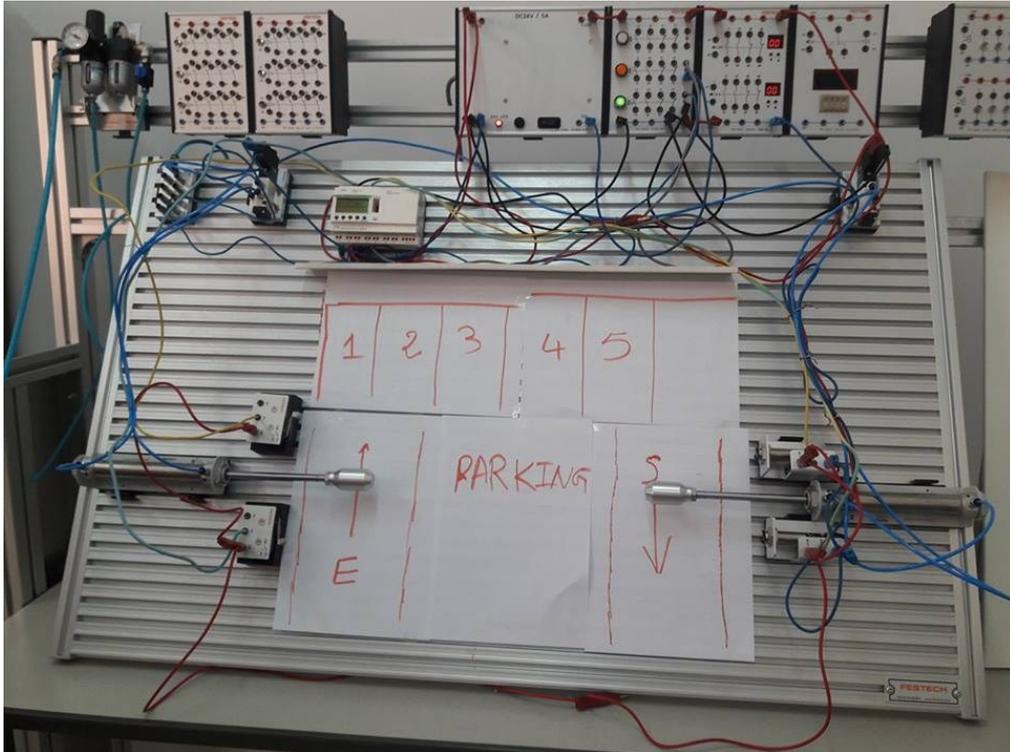


Figure 0-5: Les places sont disponibles

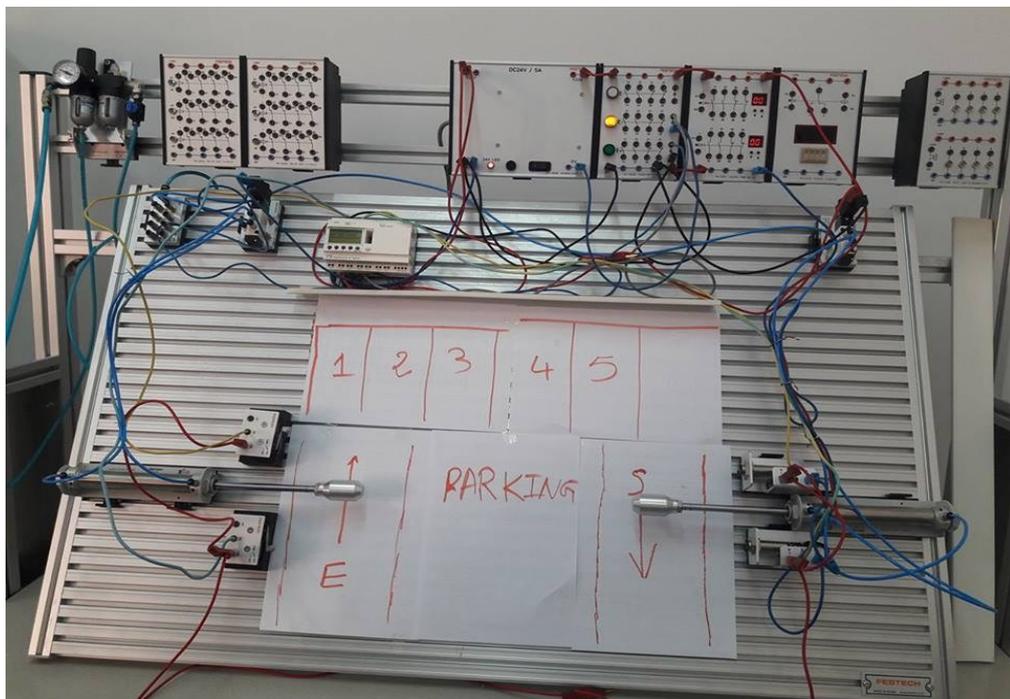


Figure 0-6: Les Places sont occupées

### 3.3.3 Schémas pneumatique

La Figure 0-7 représente le schéma pneumatique :

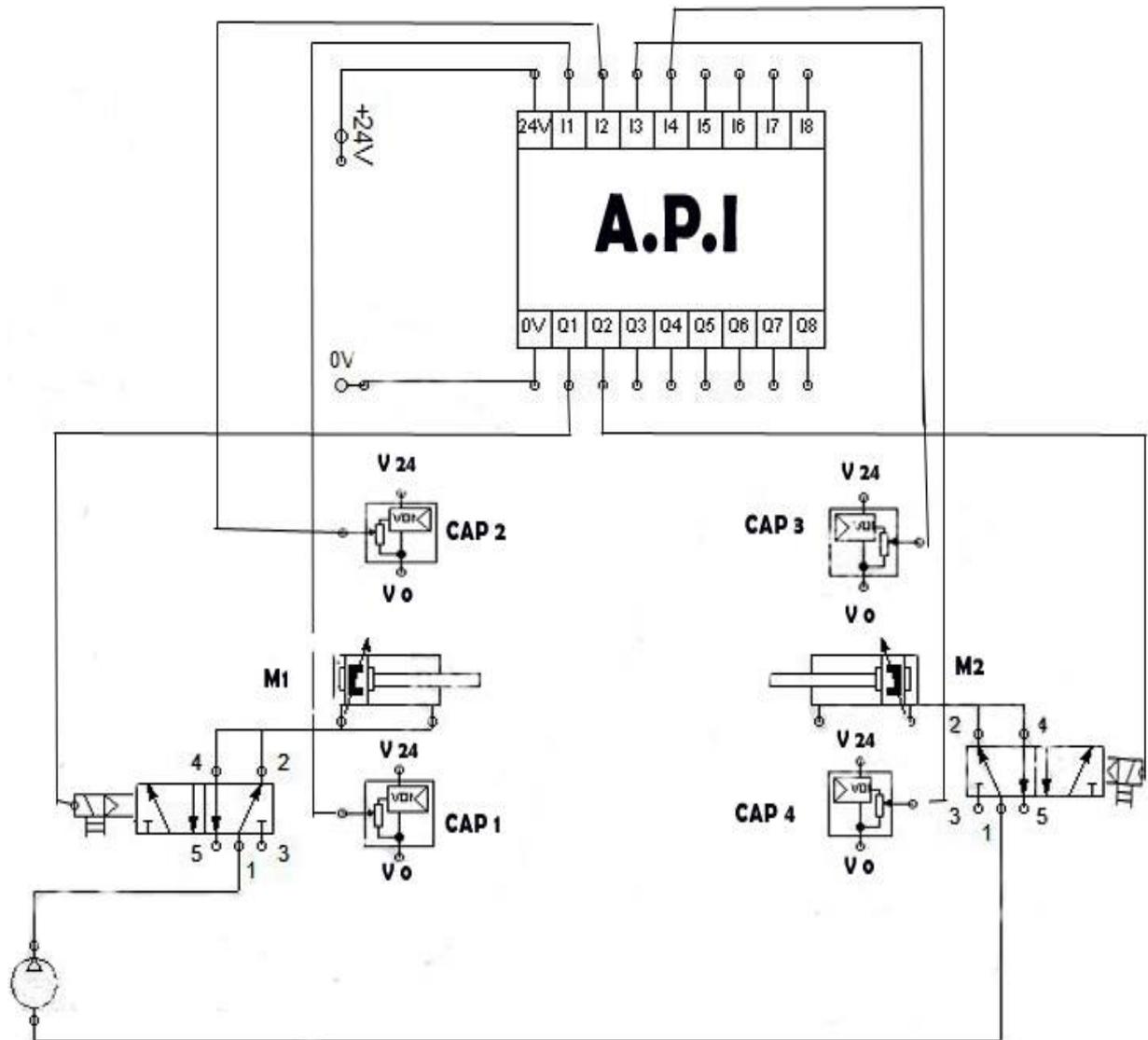


Figure 0-8: Schémas pneumatique dessinée.

Tableau 0-2: Les entrées/sorties de Barrière parking.

Entrée	Description	sortie	description
Cap 1 entrée	Extérieur de Park	M1	Actionneur entrée
Cap2 entrée	Intérieure de Park	M2	Actionneur sortie
Cap3 sortie	Intérieure de Park	Voyant vert	Park vide
Cap4 sortie	Extérieur de Park	Voyant rouge	Park occupé

### 3.3.4 GRAFCET

$$x_1 = \overline{cap1} . \overline{cap2} . \overline{cap3} . \overline{cap4}$$

$$v1 = c1 + 1 \geq 5$$

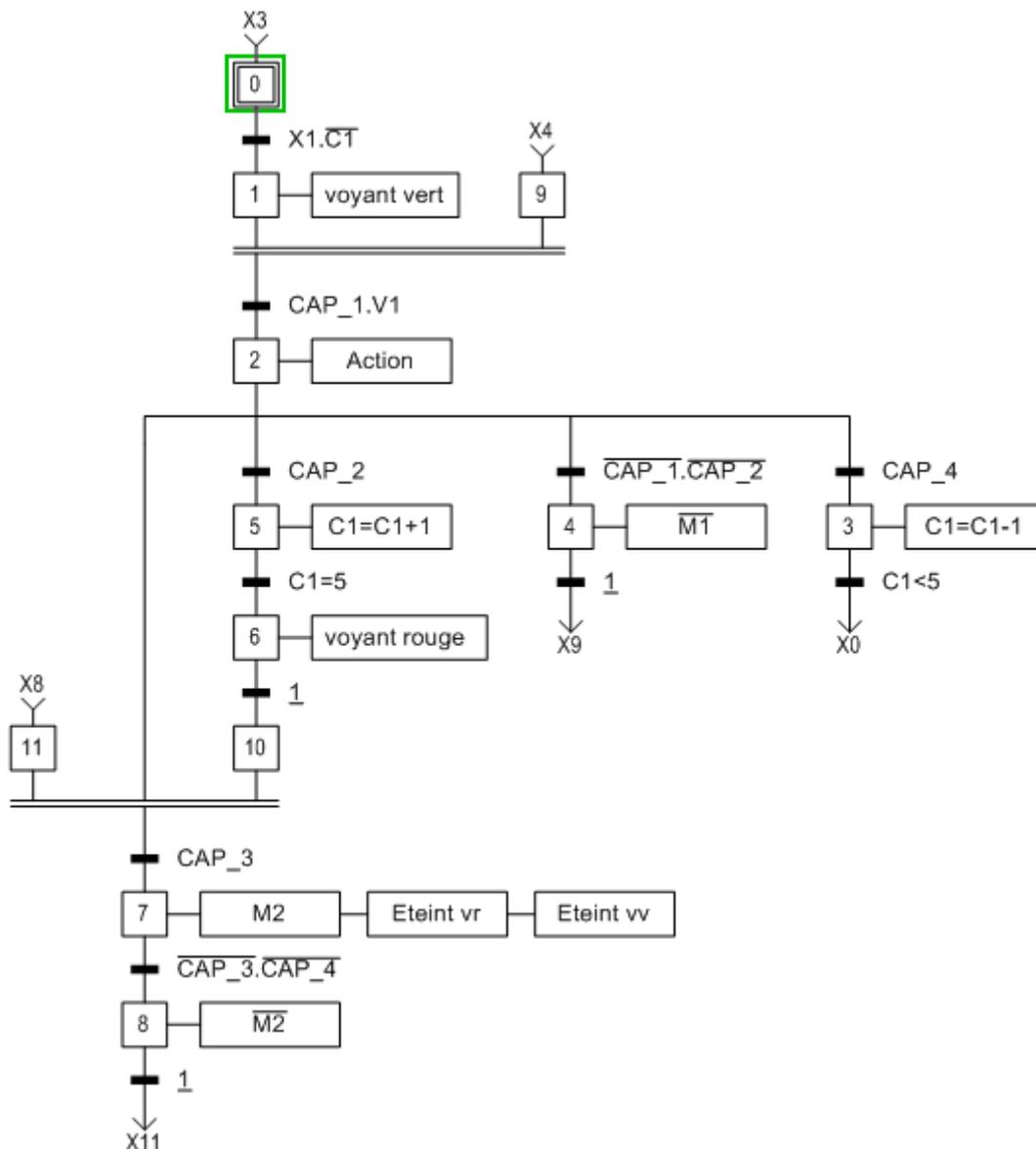


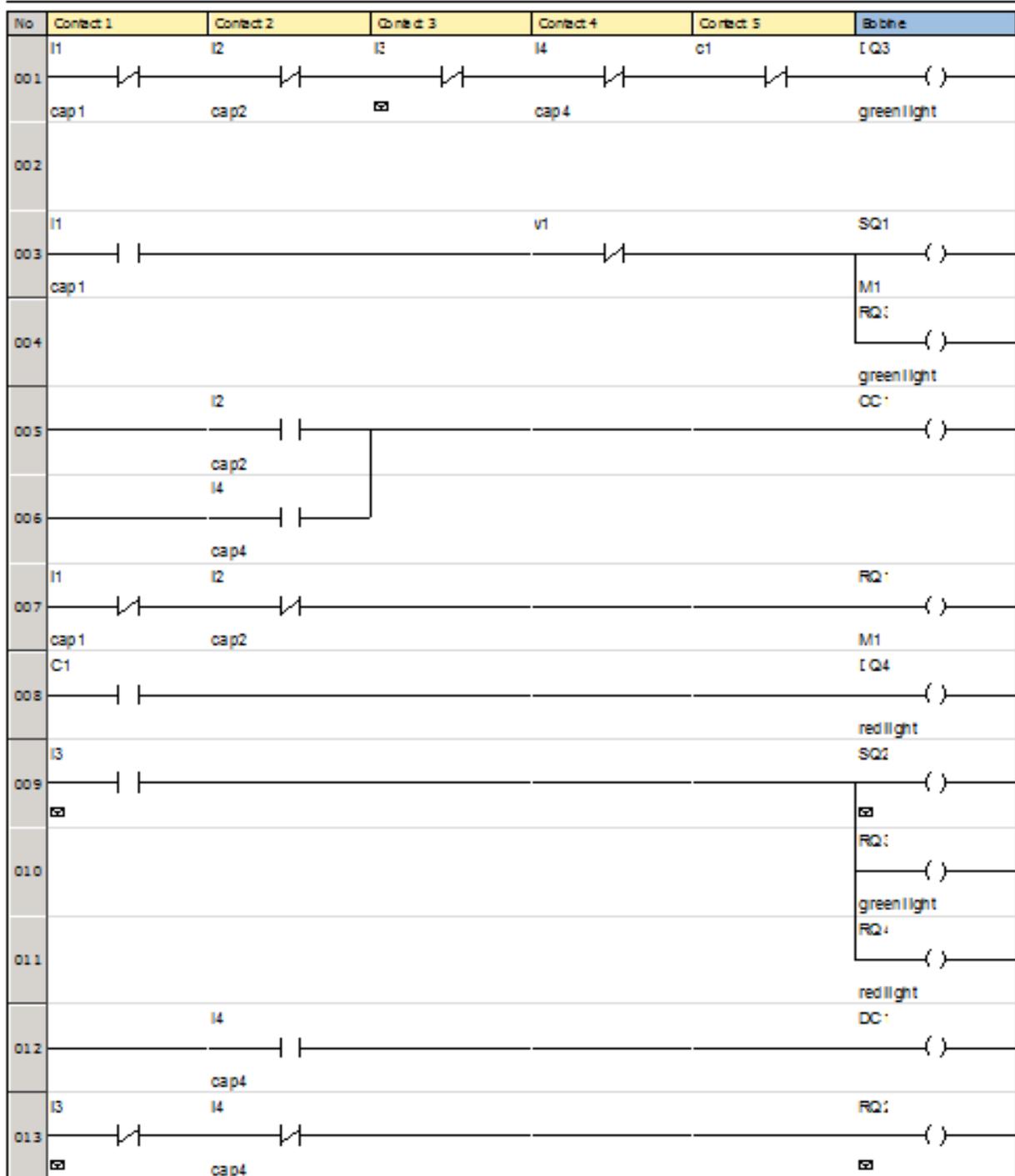
Figure 0-9: Le GRAFCET principal de parking.

### 3.3.5 Programmation

La programmation des différentes actions du cycle de fonctionnement est effectuée en langage Ladder par le logiciel Zelio Software. Dans ce qui suit, nous présentons le programme complet.

❖ **Schéma du programme en langage Ladder**

### Schéma du programme



### 3.4 Exemple 3 : l'ascenseur

#### 3.4.1 Description

Appareil élévateur installé à demeure, desservant des niveaux définis, comportant une cabine, dont les dimensions et la constitution permettent manifestement l'accès des personnes, se déplaçant, au moins partiellement, le long de guides verticaux.

### Trois types d'ascenseurs

- **L'ascenseur « classique »** : Une cabine d'ascenseur peut recevoir, en fonction de son volume, un certain nombre de personnes.
- **L'ascenseur de charge** : Une cabine d'ascenseur qui peut à la fois recevoir des charges et des personnes (monte voiture, monte-charge accompagné...)
- **Le monte-charge** : Monte-charge : appareil élévateur, destiné exclusivement au transport des charges, desservant des niveaux définis, comportant une cabine inaccessible aux personnes par ses dimensions et sa constitution ou accessible uniquement pendant les opérations de chargement, se déplaçant, au moins partiellement, le long de guides verticaux ou dont l'inclinaison sur la verticale est inférieure à 15 degrés.

Pour remplir la condition d'inaccessibilité, les dimensions de la cabine doivent être au plus égales à : surface = 1,00 m<sup>2</sup> ; profondeur = 1,00 m ; hauteur = 1,20 m.

Une hauteur supérieure à 1,20 m est admise, si la cabine comporte des compartiments fixes répondant chacun aux prescriptions ci-dessus.

#### ➤ Différents modes de traction

- **Ascenseur électrique à adhérence** : Certainement le plus répandu, il est entraîné par des câbles de traction dont les extrémités sont fixées à la cabine et au contrepoids. Entre les deux se trouve la poulie de traction entraînée par un moteur électrique.

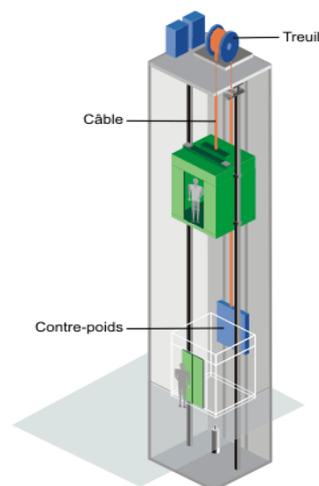


Figure 0-10: Ascenseur électrique à adhérence.

- **Ascenseur électrique à Tambour** : Ce mode de traction utilisé principalement pour limiter l'encombrement, il n'y a pas de contrepoids dans la gaine et donc la surface de la cabine peut être augmentée par rapport à un ascenseur à adhérence.



Figure 0-11: Ascenseur électrique à Tambour.

- **L'ascenseur « Hydraulique »** : Soit en attelage direct ou moufle, cet ascenseur est utilisé lorsque la place est réduite et que la course de la cabine n'est pas très élevée, il est également répandu pour les monte-voitures, monte charges ou ascenseurs de charges.

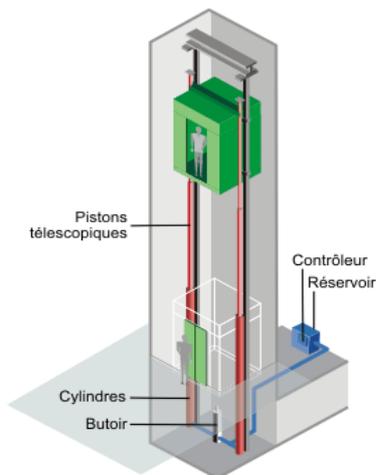


Figure 0-12: L'ascenseur « Hydraulique ».

### 3.4.2 Cahier des charges d'ascenseur relié à l'automate

Ce mode de fonctionnement permettra de réaliser un scénario particulier en créant un programme qui sera transféré ensuite dans l'automate qui lui-même actionnera les différents composants de l'ascenseur.

On considère un ascenseur desservant 3 étages et dont la boîte d'appel contient deux boutons de demande, l'utilisateur appelle la cabine avec le bouton poussoir appel palier.

Un utilisateur désirant Monter appuie sur le bouton dont la flèche est dirigée vers le haut, un utilisateur désirant Descendre appuie sur le bouton dont la flèche est dirigée vers le bas.

Si la direction de l'ascenseur est la même que celle demandée, et que l'étage d'où provient la demande n'est pas encore atteint, l'ascenseur s'arrête au passage pour prendre le passager.

L'utilisateur entre dans la cabine et il appuiera sur le bouton poussoir concernant l'étage désiré. L'ascenseur monte ensuite à l'étage désiré. L'utilisateur sort alors de la cabine.

L'ouverture des portes est automatique et leur fermeture suite à une demande ne peut s'effectuer que 30s après l'ouverture pour permettre l'évacuation aisée de la cabine.

Tableau 0-3: Entrée/sortie d'ascenseur.

Entrée	Sortie
BIE1	M
BIE2	D
BIE3	OUV
BEH1	V1
BEH2	V2
BEB2	V3
BEB3	Fe
OP	
PF	
Cap etag1	
Cap etag2	
Cap etag3	

Tableau 0-4: les ordres de commandes d'ascenseur.

---

<b>Appel intérieur (cabine)</b>	Bouton intérieur étage 1 Bouton intérieur étage 2 Bouton intérieur étage 3	BIE1 BIE2 BIE3
<b>Appel extérieur vers le haut</b>	Demande extérieur haut étage 1 Demande extérieur haut étage 2	BEH1 BEH2
<b>Appel extérieur vers le bas</b>	Demande extérieur bas étage 2 Demande extérieur bas étage 3	BEB2 BEB3
<b>Montée</b>	Monter la cabine	M
<b>Descente</b>	Descendre la cabine	D
<b>Ouverture</b>	Ouvrir la porte	OUV
<b>Allumage des voyants de l'afficheur</b>	Allumer voyant étage 1 Allumer voyant étage 2 Allumer voyant étage 3	V1 V2 V3

### 3.4.3 GRAFCET



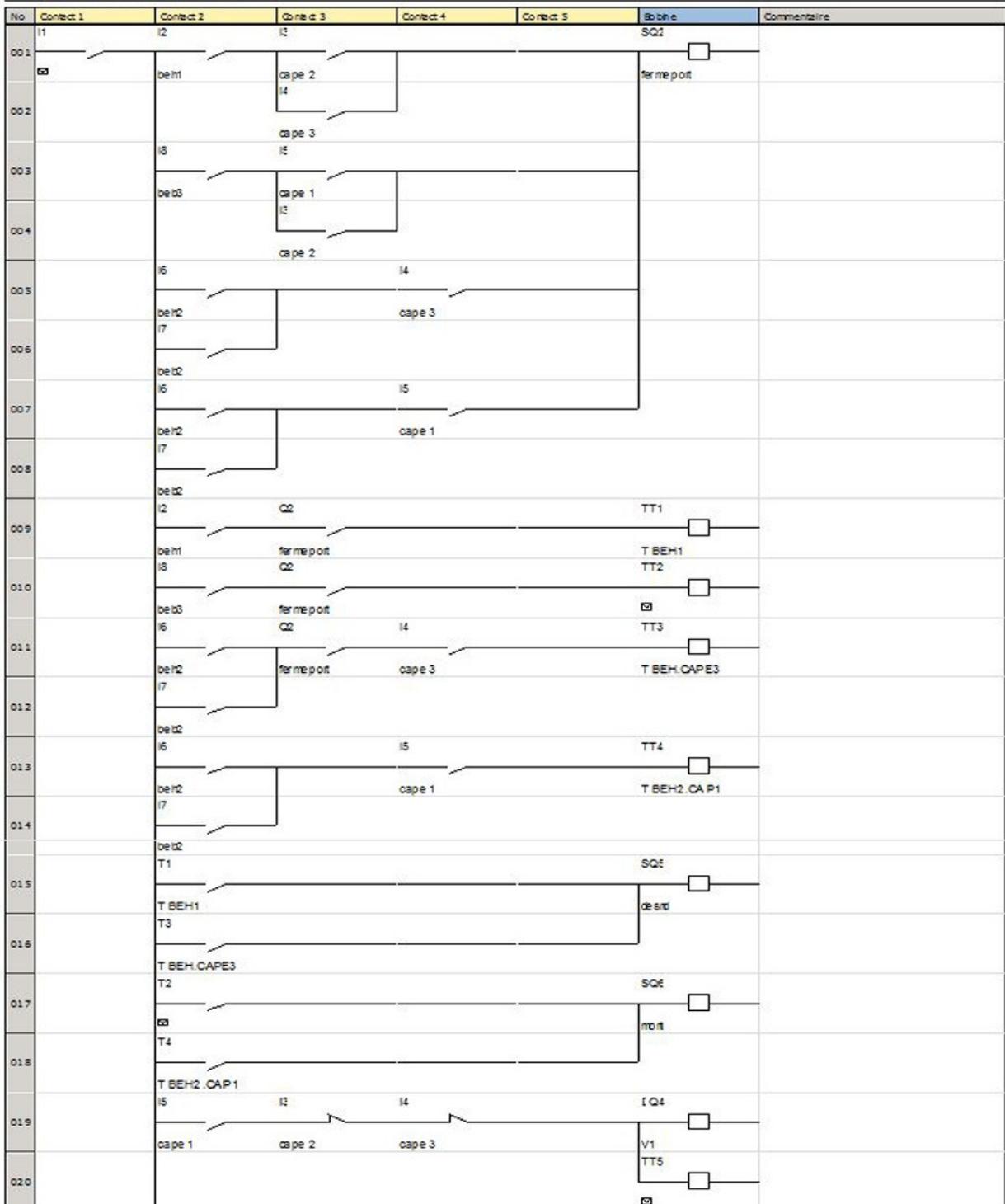
Figure 0-13: LE GRAFCET principal d'ascenseur.

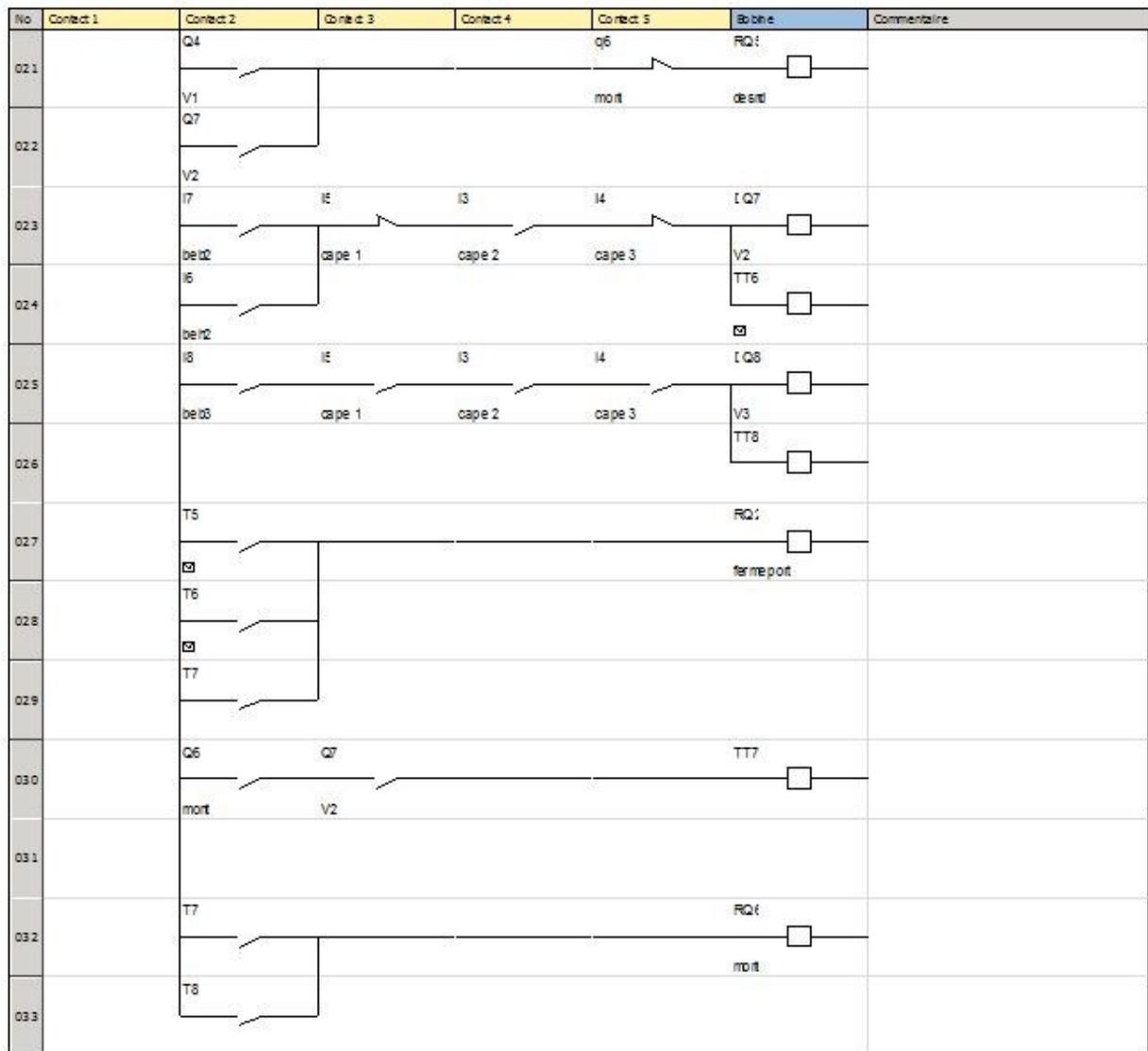
### 3.4.4 Programmation

La programmation des différentes actions du cycle de fonctionnement est effectuée en langage Ladder par le logiciel Zelio Software. Dans ce qui suit, nous présentons le programme complet.

#### ❖ Schéma du programme en langage Ladder

Schéma du programme





### 3.5 conclusion

L'automatisation d'un système quelconque nécessite une bonne modélisation assurant le bon fonctionnement de ce système ainsi que la commande et la supervision.

Dans ce chapitre nous avons fait tout d'abord une description générale des systèmes, leurs différents types et modes de fonctionnement, puis on a présenté un cahier de charges de chaque système, et en fin on a pu créer le model graphique associé à ce cahier de charges.

## Conclusion générale

Le travail réalisé au cours de ce mémoire nous a permis d'approfondir nos connaissances, et de mettre en pratique nos connaissances théoriques dans le domaine des automatismes industriels. Il nous a également aidé de bien comprendre le fonctionnement des systèmes automatisés,

En premier lieu, on a abordé une étude théorique sur les systèmes automatisés et montré les performances de ces systèmes. Ensuite on a présenté les caractéristiques de l'automate programmable industriel, puis on a appliqué la modélisation d'un cahier de charges d'un ascenseur, parking, et remplissage des bouteilles ; ces derniers répondent au besoin du client. Pour automatiser le processus on a proposé une solution programmée basée sur l'élaboration d'une plateforme de modélisation pour commander l'ascenseur et superviser son fonctionnement par un pupitre sous zelio software.

Au cours de ce projet de fin d'étude, nous avons pu concrétiser notre idée qui consiste à réaliser un parking totalement automatisé. Nous sommes arrivés à réaliser ce projet grâce aux connaissances théoriques que nous avons acquies au cours de notre formation académique d'une part, et d'autre part, grâce aux informations que nous avons apprises durant notre stage de formation qui s'est déroulé dans l'entreprise Condor Electronics. Nous avons appris beaucoup de choses sur le fonctionnement des systèmes automatisés. Nous avons également appris beaucoup de choses sur les API et leur programmation.

Vu l'importance de cette étude pour l'automatisation des différents systèmes, le coût et le manque de moyens nous a pas permis d'appliquer le modèle graphique sur une maquette assimilant le fonctionnement de notre système, donc essayer de valider les résultats trouvés sur un banc d'essai réel améliorera beaucoup cette étude au futur.

# Bibliographies

- **Les fichiers PDF :**

[1] BUTS de l'automatisation, I:\TRAVAIL\AUTOM\\_Buts\_de\_l'automatisme.DOC.2017.

[2] Jacques AIAHCE, Jean-Marc CHEREU, "les systèmes automatisés", *cours science industriel.2017.*

[3] Richard Giroux, Commande des processus industriels et automatisation industrielle, École polytechnique de Montréal, 2001

[4] ANALYSE FONCTIONNELLE ET TEMPORELLE, ANALYSE FONCTIONNELLE INTERNE – SYNTHÈSE, BTS MS.2017.

[5] COURS : Les Automates Programmables Industriels, [www.gecif.net](http://www.gecif.net). 2017

[6] F.HAMMOUCHI. Initiation aux Automates Programmables Industriels API Siemens S7\_300 année universitaire, 2014/2015.

- **Mémoires / Thèses :**

[7] ETUDE D'ASCENSEUR COMMANDE PAR AUTOMATE PROGRAMMABLE. Université sidi Mohammed ben Abdallah.

[8] M.MAATOU et A. BELLAGH, automatisation et réalisation à petite échelle (maquette) d'une chaîne transporteuse de briques, université hassiba benbouali de chlef, MEMOIRE DE MASTER. Juin 2016

[9] BELKACEM.H, RAIS.A, Système de Contrôle Distribué (DCS) avec l'exploitation de l'automate programmable AC800 F (ABB), MEMOIRE DE MASTER. Juin 2012. Université Mohamed Khider Biskra

[10] M.DILMI. Contribution à la modélisation des systèmes automatisés par un outil graphique. No. Réf. : MAC21/JUIN/2014. UNIVERSITE FERHAT ABBAS –SETIF.

[11] « Les automates programmables industriels»

Cours automatisation : de Mr.Benzouaoui.S, enseignant de département d'électronique.

- **Les sites Web :**

[12] «logiciel zelio software».2014

[http://www.graczyk.fr/lycee/IMG/pdf/11-12\\_DR\\_Tutoriel\\_Zeliosoft.pdf](http://www.graczyk.fr/lycee/IMG/pdf/11-12_DR_Tutoriel_Zeliosoft.pdf)

[http://sitelec.org/download\\_page.php?filename=schneider/zelio.pdf](http://sitelec.org/download_page.php?filename=schneider/zelio.pdf)

<http://www.schneider-electric.fr>

# Annexe

## Modules logiques Zelio Logic 1(compactes et modulaires)

### Présentation

Les modules Zelio Logic sont destinés à la réalisation de petits équipements D'automatisme. Ils sont utilisés dans les secteurs d'activité de l'industrie et du tertiaire.

#### Pour l'industrie :

Automatismes de petites machines de finition, de confection, d'assemblage où D'emballage, Petits équipements d'automatisme fonctionnant à 48 V (application de Levage, ...), Automatismes décentralisés sur les annexes de grosses et moyennes machines (Domaines du textile, du plastique, de la transformation de matériaux, ...), Automatismes pour machines agricoles (irrigation, pompage, serre, ...).

#### Pour le tertiaire/bâtiment :

Automatismes de barrières, de volets roulants, de contrôle d'accès, Automatismes d'éclairage, Automatismes de compresseurs et de climatisation..., Leur compacité et leur facilité de mise en œuvre en font une alternative compétitive Aux solutions à base de logique câblée ou de cartes spécifiques.

### Programmation

La simplicité de leur programmation, garantie par l'universalité des langages, Satisfait aux exigences de l'automaticien et répond aux attentes de l'électricien.

La programmation peut être effectuée :

De façon autonome en utilisant le clavier du module Zelio Logic (langage à Contacts), Sur PC avec le logiciel "Zelio Soft 2".

Sur PC, la programmation peut être réalisée soit en langage à contacts (LADDER), Soit en langage blocs fonctions (FBD), voir pages 8 à 12.

Le rétroéclairage de l'afficheur LCD (1) se fait par l'activation de l'une des 6 touches De programmation du module Zelio Logic ou par programmation à l'aide du logiciel "Zelio Soft 2" (exemple : clignotement lors d'un dysfonctionnement).

L'autonomie de l'horloge, assurée par une pile lithium, est de 10 ans.

La sauvegarde des données (valeurs de présélection et valeurs courantes) est Garantie par une mémoire Flash EEPROM (10 ans)

### **Modules logiques compacts**

Les modules logiques compacts répondent aux besoins d'automatismes simples.

Les entrées/sorties sont au nombre de :

12 ou 20 E/S, alimentées en à 24 V ou c 12 V, 20 E/S, alimentées en à 48 V, 10, 12 ou 20 E/S, alimentées en a 100...240 V ou c 24 V

### **Modules logiques modulaires et extensions**

Les entrées/sorties pour les modules logiques modulaires sont au nombre de :

26 E/S, alimentées en c 12 V, 10 ou 26 E/S, alimentées en à 24 V, a 100...240 V ou c 24 V  
Pour plus de performance et de flexibilité, les modules Zelio Logic modulaires Peuvent recevoir des extensions afin d'obtenir un maximum de 40 E/S :

Extensions de communication réseau Modbus ou Ethernet, alimentées en c 24 V Par le module Zelio Logic de même tension.

Extension d'entrées/sorties analogiques avec 4 E/S, alimentée en c 24 V par le Module Zelio Logic de même tension,

Extensions d'entrées/sorties TOR avec 6, 10, ou 14 E/S, alimentées par le module Zelio Logic de même tension.



- 1 Module Zelio Logic modulaire (10 ou 26 E/S)
- 2 Extension entrées/sorties TOR (6,10 ou 14 E/S) ou analogiques (4 E/S)



*Module Zelio Logic compact*

## Communication

Outils de programmation par câbles et sans fil, Les outils de programmation permettent de connecter le module Zelio Logic au PC Équipé du logiciel “Zelio Soft 2” :

- Liaison par câbles :

Câble SR2 CBL01 sur port série 9 contacts Ou Câble SR2 USB01 sur port USB.

- Liaison sans fil :

Interface Bluetooth SR2 BTC01.



*Cable de liaison*



*Interface Bluetooth*

- Cartouche mémoire :

Le module Zelio Logic peut recevoir une cartouche mémoire de sauvegarde qui Permet de dupliquer le programme dans un autre module Zelio Logic (chargement et Mise à jour du logiciel embarqué uniquement avec la cartouche mémoire R2 MEM02).

La cartouche mémoire permet aussi d’effectuer une sauvegarde du programme en Prévision d’un remplacement du produit.

Lorsqu'elle est associée à un module sans afficheur et sans touches, la copie du Programme contenu dans la cartouche est automatiquement transférée dans le Module Zelio Logic à la mise sous tension.



*Cartouche mémoire*

Extensions de communication réseau Modbus esclave et Ethernet serveur Les modules d'extension de communication réseau Modbus et Ethernet permettent Une connexion aux équipements d'automatismes tels que des afficheurs ou des Automates programmables.



*Extension de communication Modbus*



*Extension de communication Ethernet*

- Interface de communication Modem :

L'offre "Interface de communication Modem" de la gamme Zelio Logic se compose :

D'une interface de communication Modem SR2 COM01 connectée entre un Module Zelio Logic et un Modem, De Modems RTC (1) SR2 MOD01 ou GSM (2) SR2 MOD02, Du logiciel "Zelio Logic Alarme" SR2 SFT02.

Cette offre est dédiée à la surveillance ou à la télécommande à distance de Machines ou d'installations fonctionnant sans personnel.

L'interface de communication Modem, alimentée en c 12...24 V, permet de stocker Les messages, les numéros de téléphone et les conditions d'appel

106 900 52E



*Interface de communication  
Modem*

109 000



*Modem RTC*

PF 0100 52E



*Modem GSM*