REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE

MINISTERE DE L'ENSEIGNEMENT SUPERIEUR ET DE LA RECHERCHE SCIENTIFIQUE

UNIVERSITE YAHIA FARES DE MEDEA



Faculté de **T**echnologie

Département du **G**énie **E**lectrique

Mémoire de fin d'études de Master

Filière : électronique

Spécialité : instrumentation

Conception et réalisation d'un système d'automatisation d'un parking d'automobile

Présentés par :

soualhi manal

saidi randa

Proposés et Dirigés par :

KADDOUR Benyoucef

HENINI Noureddine

Promotion 2022

SOMMAIRE

Remerciements

Dédicaces

Résumé

Listes des figures ..... A

Liste des abréviations C

Introduction générale 1

CHAPITRE I : Etat de l'art Sur les Parking Automatisé 2

I.1 .Introduction 3

I.2 Définition du parking 3

I.3. Le but des parkings 3

I.4. Le Système parking automatisé des voitures 3

I.5. L'évolution de l'utilisation des parkings 4

I.6 .Les avantages d'un parking automatisés 4

I.7.Les Inconvénients des systèmes parking automatisés 5

I.8.Types de parkings de stationnement des voitures 5

A) -Parking de surface 5

B) -Parking fermé ou souterrain 6

C) -Parking à étages 6

D) -Parkings automatiques 7

E) -Parking automatisé dans une tour 7

Ⅰ.9 Conclusion 8

Chapitre II : Automatisme et Automate Programmable Industriel 9

II.1. Introduction : 10

II.2. Notion de système automatise : 11

II.2.1. Point de vue historique : 11

II.2.2. Point de vue technique : 11

II.2.3. Point de vue économique et social : 11

II.3. Définition de l'API : 11

II.4. Architecture d'un automate programmable : 12

II.4.1. Aspect extérieur de l'automate programmable : 12

II.4.2. Structure interne des automates programmables : 13
Ⅱ.4.3. Fonctions réalisées : 16
II.5. Nature des informations traitées par l'automate : 16
II.5.1. Tout ou rien (TOR) : 16
II.5.2. Analogique : 16
II.5.3. Numérique : 17
II.6. Langages de la programmation : 17
II.6.1. Programmation à l'aide du GRAFCET (SFC : Sequential Function Chart) : . 17
II.6.2. Langage à contacts (LD : Ladder Diagramme) : 19
II.6.3. Langage Blocs Fonctionnels (FBD : Function Bloc Diagram) : 20
II.6.4. Langage ST : 20
II.7. L'environnement de l'API 21
II.8. Critères de choix d'un automate : 22
II.9. Les avantages et les inconvénients : 22
Ⅱ.10 L'automate zelio : 22
II.11 Découvert l'automate zelio : 23
II.12 De quoi est constitué l'automate Zelio? 23
II.13 Domaines d'utilisation de zilio : 24
F) Définition : 24
II.15 Modes de programmation 25
G) Modules logiques avec écran : 25
H) Modules logiques sans écran : 25
Ⅱ.16 Langages utilisés 26
A) Langage à contacts 26
B) Mode FBD 27
II.17 Mode de fonctionnement 27
A) Mode saisie 27
B) Mode mise au point 27
Ⅱ.18 Test des programmes 28
Ⅱ.19 Conclusion 29
Chapitre(Ⅲ) : 30
Réalisation pratique 30
Ⅲ.1. Introduction : 31
Ⅲ.2. Les entrées de l'automate zelio : 31

Ⅲ.3. Les sorties de l'automate zelio :.....31
Ⅲ.4. Réalisations de clavier d'étage : 32
Ⅲ.4.1 Clavier d'étage 32
Ⅲ.4.2. Horloge d'étage :.....34
Ⅲ.5. Réalisation de clavier de chambre : 35
Ⅲ.5.1 Clavier de chambre: 35
Ⅲ.5.2 horloge de chambre : 36
Ⅲ.6. Circuit de commande et puissance pour Moteur asynchrone : 37
Ⅲ.7. Circuit de commande vérin : 38
Ⅲ.8. Le programme de Parking :.....41
Ⅲ.8.1 monté et décente verticalement (ascenseur):.....41
Ⅲ.8.2 Programme de rotation:.....41
Ⅲ.8.3 Programme d'entrée et sortie le véhicule:.....41
Ⅲ.9 Conclusion:.....45
Conclusion générale.....46
Liste de la référence bibliographique.....47
Annex :.....48

Remerciement

Nous tenons à remercier le dieu, le tout puissant, de nous

Avoir donné le courage et la patience jusqu'à l'achèvement de ce

Travail.

Nous exprimons notre profonde reconnaissance et nos sincères

Remerciements à notre promoteur Mr Ben Youcef Kaddour pour sa

disponibilité et

Son encadrement, ainsi que pour son soutien tout au long de l'année.

Egalement, nous remercions les membres de jury, qui ont

Accepté d''évaluer ce travail.

Nous aimerions également remercier tous nos amis (e) de leurs

Soutient et aide et qui nous ont donnés la force pour résisté.

Dédicaces

Je dédie ce travail

A mes très chers parents, source de vie e d'amour et motivation

A mes chers frères et sueurs

A toute ma famille, mes proches et à ceux qui me donnent

De l'amour et de la vivacité

A tous mes amis qui 'ont toujours encouragé, et à qui je

Souhaite plus de succès

Manal soualhi

Je dédie ce travail

A mes très chers parents, pour son encouragements et

Sont sacrifices

A mes chers frères et sœurs et leur an fans, source de jolie

Et de bonheur

Aoute ma famille, source d'espoir et de motivation

A tout ceux qui m'aiment et tous ceux que j'aime

saidi randa

LISTES DES FIGURES :

Chapitre(Ⅰ) :

Fig (Ⅰ .1) : parking de surfacé…… 5

Fig (Ⅰ .2) : parking fermé…….. 6

Fig(Ⅰ .3) : parking à étages…… 7

Fig (Ⅰ .4) : parking automatique…… 7

Fig (Ⅰ .5) : Parking automatisé dans une tour……. 8

Chapitre(Ⅱ) :

Fig (Ⅱ .1) : Automates Programmables Industriels (API) …… 12

Fig (Ⅱ .2) : API type modulaire Schneider (zelio) et siemens (LOGO)……. 13

Fig (Ⅱ. 3) : Structure et interne de l'ApI……...…… 13

Fig (Ⅱ .4) : Architecture de l'unité centrale……..…… 14

Fig (Ⅱ .5) : Partie opérative et partie commende……. 16

Fig (Ⅱ.6) : Grafcet divergence et convergence en ET……..… 17

Fig (Ⅱ.7) : Grafcet divergence et convergence en OU……...…. 18

Fig (Ⅱ.8) : Grafcet saut d'étape……..... 18

Fig (Ⅱ.9) : Grafcet reprise d'étape…… 19

Fig (Ⅱ.10) : Exemple programme LD avec de liaison multiple a 'droite…….. 19

Fig (Ⅱ.11) : Exemple programme FBD avec de saut et d'étiquette…….. 20

Fig (Ⅱ.12) : Exemple programme ST ……. 21

Fig (Ⅱ.13) : structure d'un système automatisée…… 21

Fig (Ⅱ.14) :L'automate zelio……. 23

Fig (Ⅱ .15) : Utilisé zlio softe…… 24

Fig(Ⅱ.16) : Modules logiques avec écran …… 25

Fig (Ⅱ.17) : Modules logiques sans écran …….. 25

Fig (Ⅱ.18) : Langage à contacts…… 26

Fig (II.19) : Mode FBD ……. 27

Chapitre(Ⅲ) :

Fig (Ⅲ.1) : Clavier d'étage……. 32

Fig (Ⅲ.2) : Horloge d'étage……..…….. 34

Fig (Ⅲ.3) : Clavier de chambre…….. 35

Fig (Ⅲ.4) : Horloge des chambres……. 36

Fig (Ⅲ.5) : Circuit de puissance de moteur M1 avec deux sens de marche…….. 37

Fig (Ⅲ.6) : Technique de rentrer et sortir le véhicule…….…… 38

Fig (Ⅲ.7) : Un distributeur 5/3 bistable à commande électrique et a centre fermé 38

Fig (Ⅲ.8) : Circuit de commande de vérin double effet par le distributeur 5/3 a centre fermé avec la commande Q3 et Q4 de module zelio……. 39

Fig (Ⅲ .9) : La fonction CAM …… 40

Fig (Ⅲ.10) : Choix de nombre de position…… 40

Fig (Ⅲ.11) : La fonction Archive…….. 40
Fig (Ⅲ.12) : Programme de parking……. 42
Fig (Ⅲ.13) : Le sous-programme Mo1……..... 43
Fig (Ⅲ.14) : Le sous-programme Mo2……...…. 44

LISTE DES ABREVIATIONS

API	Automate programmable industriel.
SPA	Système parking Automatisé.

ملخص

أصبحت مواقف السيارات ضرورة في المدن الكبرى، فهي اقتصادية، مفيدة جدا وتوفر خدمات رائعة للغاية. ينتشر هدا العمل الموضعي في سياق الأتمة الصناعية الحديثة، واستكشفنا الية نظام برمجة زيليو شنايدر. كما وضعنا مفهوما تقنيا لتحسين تدفق وقوف السيارات. يقدم هدا العمل موقف سيارات تحت الأرض يتكون من (16) طابق مع(8) غرف في كل طابق.

الكلمات الدالة: موقف، نظام الي، الية البرمجة التلقائية، زيليو.

Résumé

Les parkings sont devenus une nécessité dans grandes agglomérations, très économique est très bénéfique, est rend service très remarquable.

Ce travail d'actualité s'étale dans le cadre d'automatisme industriel récents, on a exploré un API Schneider (Zelio), et faire un concept technique pour améliorer le flux de stationnement.

Ce travail présent un parking sous terrain à 16 étages avec 8 positions dans chacun des étages.

Mots clés ;

API, Zelio, Parking, Système Automatisé,……

Summary

Car parks have become a necessity in large cities, very economical, very beneficial, and provide a very remarkable service.

This topical work is spread out in the context of recent industrial automation, we explored a Schneider API (Zelio), and made a technical concept to improve the parking flow.

This work presents an underground car park with 16 floors with 8 positions in each floor.

Key words ;

API, Zelio, Parking, Automated System,……

Introduction générale

INTRODUCTION GENERALE

La recherche d'une place pour garer un véhicule est devenue un problème quotidien auquel sont confrontées les personnes, En raison du La croissance rapide de La population urbaine dans différentes villes du monde entraîne une augmentation du nombre de véhicules, provoquant des embouteillages. Le parking est un moyen important pour régler cette situation.

Afin de guidez les utilisateurs vers les places libres, rapide et facile. Les solutions des stationnements intelligents révolutionnent. Fluidifier la recherche d'une place implique une réduction de trafic, moins de pollution et une plus grande satisfaction pour les conducteurs et donc bénéficiez d'une gestion plus performent au travers d'une productivité élevée des places de stationnement ainsi que d'un contrôle très efficace.

Il existe différents types de parkings suivant les caractéristiques des villes. Ou on trouve le stationnement au sol et sous-sol et les parkings à plusieurs étages, mais tous partagent la présence de plusieurs inconvénients connexes, tels que la taille ou la petitesse du l'espace, ce qui rend difficile la sortie ou l'entrée de la voiture.

Pour surmonter les problèmes ci-dessus, l'automatisation des parkings des voitures est devenue nécessaire.

C'est dans cette optique que s'inscrit ce projet :

Le parking proposé contient 16 étages sous-sol avec 8 positions de stationnement dans chaque étage

On propose une solution d'un système de parking automatisé des voitures pour résoudre ou bien diminuer les problèmes posés au stationnement des voitures dans les parkings conventionnels.

Ce travail est détaillé dans ces trois chapitres :

- Dans le premier chapitre, nous allons citer l'état de l'art sur les systèmes de parking ainsi leurs avantages et leurs inconvénients.

- Dans un deuxième chapitre, est consacré à l'automatisme et automate programmable industriel.

-Dans le dernier chapitre, La réalisation pratique.

CHAPITRE I

ETAT DE L'ART SUR LES PARKING AUTOMATISE

I.1 .Introduction

Les villes sont prises avec la question apparemment insoluble du stationnement : les quartiers les plus attractifs suscitent une demande toujours croissante de stationnement. Or, la croissance de l'offre de stationnement s'avère incompatible avec de nombreux objectifs des municipalités, comme la lutte contre l'étalement urbain ou l'essor du transport en commun et des déplacements actifs. Mieux appréhender cette question exige de revoir la place de l'automobile dans la ville, notamment de reconsidérer la quantité et la localisation des cases de stationnement. Pour cela on va étudier les systèmes de parking de voiture qui ont pour objectif de minimiser l'espace des bonnes conditions de stationnement.

I.2 Définition du parking

Le parking d'automobiles est un ensemble des voitures immatriculées, Stationner pendant un temps déterminé dans une surface clôture.
On trouve les parkings dans différents lieu (aéroports, hôpitaux, les grande marchés, etc…) [1]

I.3. Le but des parkings

- Éliminer le problème de la surpopulation.

- Optimisation de la surface de parking.

- Innovation de la surface de parking.

- Une bonne gestion de parking (temps, sécurité, etc…).

I.4. Le Système parking automatisé des voitures

Parking automatisé est une méthode de garer et récupérer les voitures automatiquement qui utilisent généralement un système de palettes et des ascenseurs. L'intention est de voitures compactes plus dans le même espace de stationnement. Parking automatisé peuvent être situés au-dessous ou au-dessus du sol ou sur ou une combinaison des deux. Le système automatisé de parking facilite la manière de garer et récupérer les voitures, ainsi la distance parcoure par chaque véhicule est réduite.

Selon le cabinet de consultants environnementaux Waterman Group, les parcs de stationnement utilisant un système robotisé pour le rangement des véhicules génèrent moins de CO2 que les parcs classiques. [1]

Le parking automatisé doit aussi varier les moyens de paiement. Par exemple, tout le monde n'a pas forcément de monnaie au moment de payer le parking. Il faut développer les paiements par carte ou à l'aide Smartphones. L'identification des véhicules entrent dans

le parking, ainsi aucun paiement n'est demandé, un capteur détecte la voiture et son identifiant à l'entrée et à la sortie du parking et calcule donc automatiquement le prix à payer par le conducteur. Une facture mensuelle ou trimestrielle peut donc être mise en place.

Au fil des années, le monde automobile évolue et, on voit apparaitre de plus en plus de voiture électrique. Ces voitures électriques sont bénéfiques contre la pollution mais il faut qu'elles soient rechargées. C'est pourquoi des zones de rechargement doivent être prévues, et plus spécialement des places de parking ou l'on peut les recharge. En effet, il serait très avantageux que l'utilisateur puise recharge sa voiture pendant son stationnement. [2]

I.5. **L'évolution de l'utilisation des parkings**

Une recherche estime que les revenus atteindront 356,5 millions de dollars par année d'ici 2020 dans le monde. Ainsi, il a beaucoup de potentiel et connaîtra une croissance énorme dans les années à venir. Avec les demandes de stationnement croissantes, les entreprises proposent des solutions innovantes et intelligentes pour répondre à ces exigences. [2]

I.6 .**Les avantages d'un parking automatisés**

- Premièrement, le propriétaire des places de parking est assuré que seules les voitures autorisées y ont accès. La sécurité de son établissement est donc garantie. Il peut aussi, s'il le désire, contrôler les accès facilement grâce aux informations centralisées sur le tableau de bord.

- Deuxièmement, le conducteur louant la place de parking ne doit s'encombrer de rien. Il cherche une place de parking à proximité, la réserve, encode les données de son véhicule et paie sa location de façon sécurisée. Enfin, ces informations seront envoyées à votre tableau de bord et il n'aura plus qu'à se présenter aux portes du parking où son entrée sera autorisée par la reconnaissance automatique de sa plaque d'immatriculation.

- Troisièmement, la vérification est automatique et entièrement robotisée par la technologie algorithmique de la reconnaissance. Le propriétaire ne doit donc plus s'occuper de la gestion de son parking lui-même. Il ne doit plus non plus employer de préposé qui s'en occupera. Le logiciel de SPA le fait pour lui. De plus, l'utilisateur peut entrer et sortir rapidement et facilement de la place de parking qu'il a loué.

- Quatrièmement, optimisation de l'espace, par rapport aux parkings conventionnels, SAP exige beaucoup moins de surface et de volume pour garer un nombre équivalent de voiture.

- SAP économique jusqu'à 50% au plus du volume de construction par rapport au stationnement conventionnement en éliminant l'exigence de grands espaces de stationnement pour permettre l'accès aux piétons, ouvrir les portes des voitures, les rampes de véhicules, les véhicule, les voies de circulation, les trottoirs, les escaliers, les ascenseur de passagers, etc.

- Cinquièmement, flexibilité de conception, les possibilités d'application d'un ASP sont pratiquement illimitées : au-dessus du sol, sous terre, indépendants, en construction ou intégrées dans des bâtiments existants. Ils peuvent créer un stationnement dans espaces ou le stationnement conventionnel est impossible ou peu pratique, comme des zones très étroites ou irrégulières. Une parcelle mesurant seulement environ 22 X 22 mètres permet la construction d'un SAP avec jusqu'à 200 places de stationnement. [4]

I.7.Les Inconvénients des systèmes parking automatisés

- la consommation d'énergie est élevée pour fonctionner tellement système.

- Ce système est plus complexe à construire.

- Les clients doivent payer de gros tarifs.

- Les couts de construction sont très élevés. [4]

I.8.Types de parkings de stationnement des voitures

A) -Parking de surface

Se situe de plaine pied, à l'extérieur, sur l'espace public ou privé. Ce type de parking comprend le stationnement en voirie (places le long d'une rue, d'un quia, etc.) et les espaces dégagés à cette fin entre des bâtiments, ou établis sur des anciens champs, des anciens terrains vagues, etc.



. **Fig (I .1) :** parking de surfacé [6]

B) **-Parking fermé ou souterrain**

En ville ou sous les aéroports, sous les bâtiments de certaines zones d'activité, souvent sur plusieurs niveaux, ils permettent d'économiser le foncier. Chaque niveau s'apparente à un Parking classique, à la différence que l'air y est plus confiné et pollué, [5Que le sol n'y est pas lessivé par les pluies, qu'il peut être recouvert d'un revêtement particulier et que l'on y retrouve des piliers à intervalles réguliers pour soutenir la structure.

Des rampes permettent de passer d'un niveau à l'autre. Des ascenseurs ou des escaliers permettent aux occupants des véhicules, une fois ceux-ci garés, de remonter à la surface.

Dans plusieurs pays (dont France), les parkings souterrains sont désormais obligatoires pour toutes les constructions d'immeubles dans certains zonages urbains, avec des prescriptions en matière d'aération, lutte contre l'incendie, sorties de secours, etc.



Fig (I .2) : parking fermé. [6]

C) **-Parking à étages**

Un parking à étages, aussi nommé parking silo, est un bâtiment construit en extérieur. Son mode de fonctionnement ressemble beaucoup à celui du parking souterrain à la différence qu'il ne demande pas de lourds travaux de creusement. Là aussi des rampes permettent de monter ou de descendre les étages en voiture, et il y a des ascenseurs et des escaliers pour les piétons, Les parking silo se prêtent particulièrement bien à un fonctionnement mutualisé. [5]



Fig(Ⅰ.3) : parking à étages. [6]

D) -**Parkings automatiques**

Les parkings Automatique sont généralement des ouvrages souterrains ou en élévation dont les rampes intérieures sont remplacées par des systèmes de levage de translation des véhicules. [5]



Fig (Ⅰ.4) : parking automatique. [7]

E) -**Parking automatisé dans une tour**

Un superbe parking construit dans une tour compte par étages illustré dans la figure 1.5, ceux-ci son fonctionnement par une palette automatisée pouvant d'aller chercher les voitures dans leur places, ou de faire le contraire. [4]



Fig (Ⅰ .5) : parking automatisé dans une tour. [7]

Ⅰ .9 Conclusion

Dans ce chapitre nous avons exploré les différents types de parking ainsi leurs avantage et inconvénients et on est intéressés au système de parking automatisé des voitures dans une tour, car dans ce type on peut maximiser l'emplacement des voitures, il assure une bonne gestion et sécurité Sans oublier que ce système de parking peut être bâtit au-dessus ou au-dessous du sol qui va assurera une Optimisation des surfaces.

CHAPITRE II

AUTOMATISME ET AUTOMATE PROGRAMMABLE INDUSTRIEL

II.1. Introduction :

L'automate programmable industriel est une commande conçue autour d'un microprocesseur. Sa conception et son langage de programmation sont spécialement adaptés aux contrôles de processus industriel. L'abréviation PLC de l'Anglais ″Programmable logiciel Controller″. Sa définition est donnée par la norme NFC 63-850 :

« Appareil électrique qui comporte une mémoire programmable par un utilisateur automaticien (non informaticien) à l'aide d'un langage adapté, pour le stockage interne des instruction composant les fonctions d'automatisme comme par exemple :

- ✓ Logique séquentielle et combinatoire,
- ✓ Temporisation, comptage, décomptage, comparaison,
- ✓ Calcul arithmétique.
- ✓
- ✓ Réglage, asservissement, régulation, etc… pour commander, mesuré et contrôler au moyen de modules d'entrées et de sorties (logique, numérique, ou analogique) différentes sorties de machines ou de processus, en environnement industriel ».

Les automates programmables industriels sont des équipements électriques à câblage interne indépendant du processus à commander. L'automate programmable industriel est adapté à la machine ou à l'installation à commander au moyen d'un programme qui définit le mode de déroulement des opérations souhaité et d'un câblage direct entre les éléments d'entrée et de sortie de l'automate.

Les éléments d'entrée sont par exemple des auxiliaires de commande, des contacts de fin de course, des détecteurs de proximité mais aussi des tensions analogiques ou des détecteurs de courant (signaux en retour de la machine, signaux de commande issus du pupitre).

Les contacteurs, électrovannes, dispositifs de couplage, lampes, etc. sont des éléments de sortie (signaux de commande émis vers la machine, signaux de visualisation vers le pupitre).

Le nombre d'instructions traitées varie d'un automate à l'autre. Le programme est introduit à partir d'une console de programmation. Le jeu d'instruction d'un automate est orienté vers les problèmes de logique et vers les systèmes à évolution séquentielle.

Les API se caractérisent par rapport aux ordinateurs par leur fiabilité et leur facilité de maintenance. Les modules peuvent être changés très facilement et le redémarrage des API est très rapide.

Au cours de la dernière décennie, la baisse du prix des A.P.I et des micro-ordinateurs a modifié sensiblement le domaine d'emploi des A.P.I. Au déterminent de la logique câblée et des microprocesseurs. Les produits de type cartent à microprocesseurs n'ont pas eu les développements espérés.

II.2. Notion de système automatise :

II.2.1. Point de vue historique : L'évolution des systèmes automatises s'est faite en trois temps avec deux grandes ruptures. Chaque temps correspond à un type d'époque et à un type de logique de pensée. Ainsi on passe d'une vision imitative (Imitations des vivants au travers d'automates) à une vision mécaniste dans les années 50 (Opérationnalisation des machines, la machine remplace l'homme dans les tâches difficiles et répétitives, Ford) pour terminer vers une vision actuelle systémique (c'est l'ère de l'intelligence artificielle naît de la cybernétique de Wiener, la machine réfléchie, analyse et s'adapte, la spécialisation à fait place à la polyvalence et à la capacité d'adaptation).

II.2.2. Point de vue technique : La fonction globale d'un système automatisé est une modification sur l'environnement pour laquelle de l'énergie est mobilisée. Sa finalité est d'apporter une valeur ajoutée à la matière d'œuvre (énergie, information, matière) qui est mise en jeu. Les applications sont vastes : industrielles, environnement, domotique (système de régulation et de sécurité), gestion à distance. [8]

II.2.3. Point de vue économique et social : Les systèmes automatisés ont un impact économique très important dans le monde industriel (gain de temps du à la robotisation, meilleure productivité, rendement optimisé...), mais aussi ils amènent des incidences sociales fortes autant, pour le travailleur (évolution des métiers, responsabilité accentuée, sécurité renforcée, imagination accrue, adaptabilité par la polyvalence), que pour le citoyen (amélioration des tâches ménagères, communication variées, commande à distance…).[8]

II.3. Définition de l'API : Un automate programmable industriel (ou API) est un dispositif électronique programmable destiné à automatiser des processus tels que la commande de machines au sein d'une usine et à piloter des robots industriels par exemple. Il reçoit des données par ses entrées, celles-ci sont ensuite traitées par un programme défini, le résultat obtenu étant délivré par ses sorties. Ce cycle de traitement est toujours le même, quel que soit le programme, néanmoins le temps d'un cycle d'API varie selon la taille du programme et la puissance de l'automate. C'est l'unité centrale qui gère l'automate

programmable : elle reçoit, mémorise et traite les données entrantes et détermine l'état des données sortantes en fonction du programme établi. [8]

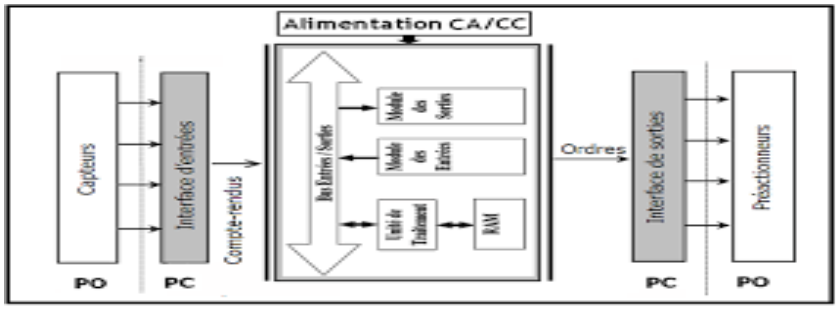


Fig (II.1) : Automates Programmables Industriels (API)

II.4. Architecture d'un automate programmable :

II.4.1. Aspect extérieur de l'automate programmable : Aspect extérieur peuvent être de type compact ou modulaire. De type compact : On distinguera les modules de programmation (LOGO de Siemens, ZELIO de Schneider, MILLENIUM de Crozet ...) 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 analogiques ...) et recevoir des extensions en nombre limité.

Ces automates, de fonctionnement simple, sont généralement destinés à la commande de petits automatismes. De 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.

II.4.2. Structure interne des automates programmables :



Fig (II.2): API type modulaire Schneider (Zelio) et siemens (LOGO)

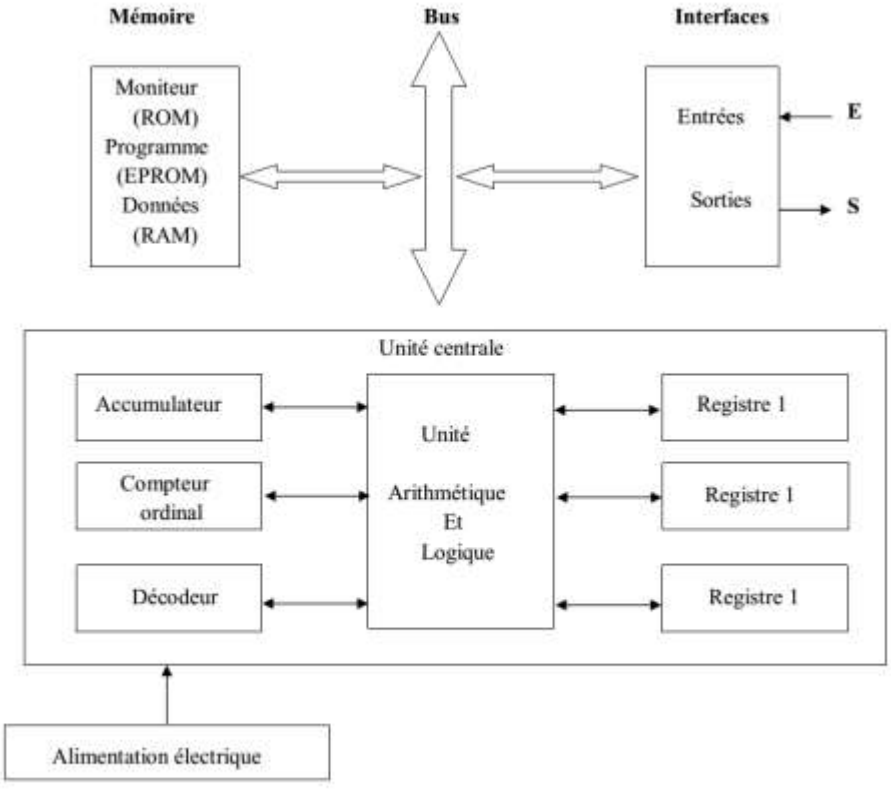


Fig (II.3): Structure interne de l'API

II.4.2.1. L'unité centrale : C'est l'ensemble des dispositifs nécessaire en fonctionnement logique interne de l'API, c'est le cœur à automate base de microprocesseur, elle réalise toutes les fonctions logiques, arithmétiques et de traitement numérique (transfert, comptage, temporisation ...).

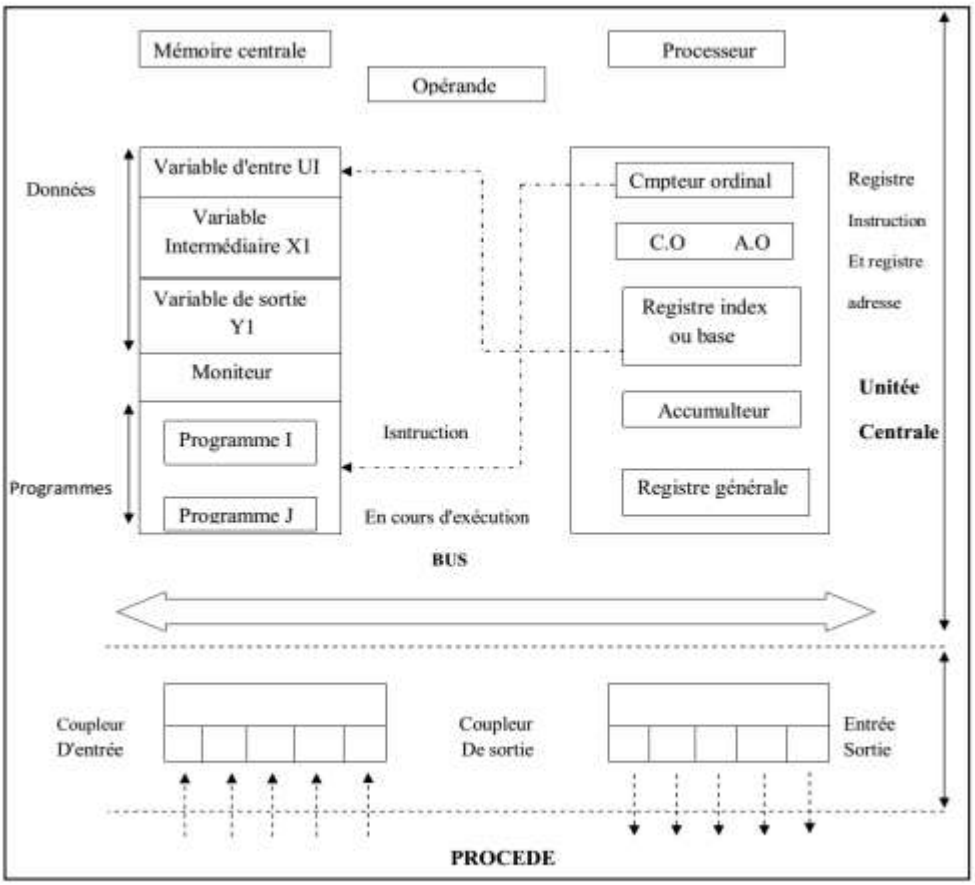


Fig (II.4): Architecteur de l'unité centrale [14]

Processeur : Cœur de l'appareil, dans l'unité central, ce n'est sans doute pas, paradoxalement, Le point le plus caractéristique, mais il conditionne tout de même largement les performances. Les premières API étaient équipées de processeurs spécifiques, à cycle de scrutation unique : on exécutait en permanence un programme gérant essentiellement des variables Binaires. On est passé ensuite à des processeurs plus performants, issus du monde de l'informatique. Cette évolution a permis de baisser les coûts, d'accroitre les possibilités opérationnelles. Les processeurs étant toutefois ponctuellement moins efficaces que les spécialisés, on peut penser que c'est aussi une des raisons pour laquelle les progrès (en temps de calcul par exemple) sur un ensemble d'opérations de base ont été moins spectaculaires qu'en informatique générale, avec une conséquence heureuse pour les utilisateurs : une longévité supérieure du matériel. Des progrès importants ont été accomplis sur d'autres points, sur lesquels nous reviendrons.

L'unité central UC est une carte électronique bâtie autour de la (ou des) « puce(s) » processeur(s), qui assure au moins les fonctions suivantes : opérations logiques sur bits (le bit, contraction de « binary digit », étant l'information élémentaire à deux états) ou sur mots (ensemble de bits, le plus souvent 16 pour les API) ; temporisation et comptage.

Il existe trois technologies de réalisation : la technologie câblée, la plus rapide mais aussi la plus coûteuse, réservée à des usages particuliers ; la technologie à microprocesseur, la plus économique dès lors que l'on utilise un microprocesseur standard produit en grand série ; la technologie mixte, certaine opération étant réalisée en câble pour en accroître la rapidité.
II.4.2.2. Module d'alimentation : Il assure la distribution d'énergie aux différents modules. **II.4.2.3. Le bus interne** :

Il permet la communication de l'ensemble des blocs de l'automate et des éventuelles extensions.

II.4.2.4. Mémoires :

Elles permettent de stocker le système d'exploitation (ROM ou PROM), le programme (EEPROM) et les données système lors du fonctionnement (RAM). Cette dernière est généralement secourue par pile ou batterie. On peut, en règle générale, augmenter la capacité mémoire par adjonction de barrettes mémoires type PCMCIAA l'intérieur de l'automate il y a un bloc d'alimentation fournissant les différentes tensions nécessaires aussi bien à la logique de commande (5 V à ± 15 V) qu'aux entrées sorties (24 V à 220 V). La tension normalisée la plus courante est de 24 V (alternatif ou continu).

Généralement l'alimentation des capteurs est fournie par l'API (en 24 V) alors que celle des prés actionneurs est fournie par une alimentation externe. En courant continu, la tension des signaux de sortie de l'API sont de l'ordre de 24, 48 ou110 V, alors que l'intensité du courant varie entre 0.5 et 2A. En Alternatif 50HZ, elles peuvent aller jusqu'à 220 V et 5A respectivement.

II.4.2.5. Interface d'entrées/sorties (E/S) :

Ils assurent le rôle d'interface de la partie commande (PC) dans le schéma classique de la figure 3, qui distingue une partie opérative, où les actionneurs agissent physiquement sur le processus, et une partie commande récupérant les informations sur l'état de ce processus et coordonnant en conséquence les actions pour atteindre les objectifs prescrits (matérialisés par des consignes).

Pour ce faire, ils doivent :

- regrouper des variables de même nature, pour diminuer complexité et coût ;
- assurer le dialogue avec l'U C ;
- « traduire » les signaux industriels en informations API et réciproquement, avec une protection de l'UC et un traitement adéquats. Beaucoup d'automates assurent cet interfaçage par des modules amovibles comportant un nombre fixé de voies d'un certain type.
- Alimentation électrique.
- Exemple : on pourra trouver un module « 16 entrées TOR 220 V alternatif ».

✓ Interface d'entrée : elle permet de recevoir les informations du S.A.P. ou du pupitre et de mettre en forme (filtrage,…) ce signal tout en l'isolant électriquement.

✓ Interfaces de sortie : elle permet de commander les divers prés actionneurs et éléments de signalisation du S.A.P. tout en assurant l'isolement électrique

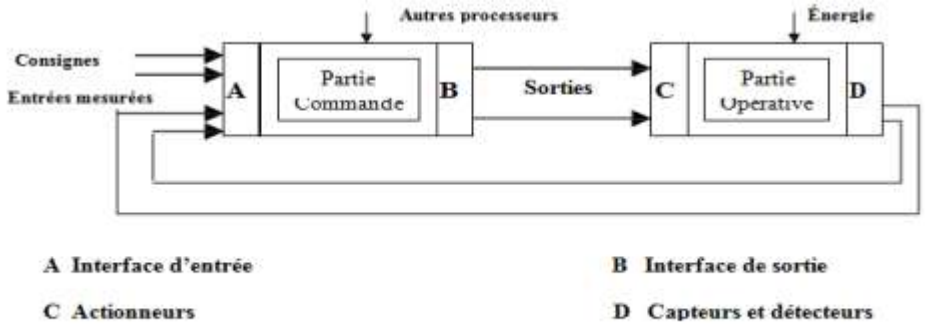


Fig (II.5): Partie opérative et partie commande.

Ⅱ.4.3. Fonctions réalisées : Les automates compacts permettent de commander des sorties en T.O.R et gèrent parfois des fonctions de comptage et de traitement analogique. Les automates modulaires permettent de réaliser de nombreuses autres fonctions grâce à des modules intelligents que l'on dispose sur un ou plusieurs racks. Ces modules ont l'avantage de ne pas surcharger le travail de la CPU car ils disposent bien souvent de leur propre processeur

II.5. Nature des informations traitées par l'automate :

Les informations peuvent être de type :

II.5.1. Tout ou rien (TOR) : L'information ne peut prendre que deux états (vrai/faux, 0 ou 1). C'est le type d'information délivrée par un détecteur, un bouton poussoir…etc.

II.5.2. 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…).

II.5.3. Numérique : L'information est contenue dans des mots codés sous forme binaire ou bien hexadécimale. C'est le type d'information délivrée par un ordinateur ou un module intelligent.

II.6. Langages de la programmation :

Il y a cinq langages de programmation principaux :

✓ Sequential Function Chart (SFC) ou GRAFCET pour les opérations de haut niveau
✓ Function Block Diagram (FBD) ou logigramme pour les opérations cycliques complexes.
✓ Ladder Diagram (LD) ou schéma à contacts pour les opérations cycliques booléennes.
✓ Structured Text (ST) pour les opérations cycliques.
✓ Instruction List (IL) pour les opérations de bas niveau.

II.6.1. Programmation à l'aide du GRAFCET (SFC : Sequential Function Chart) :

Le langage SFC (Sequential Function Chart), ou GRAFCET, est un langage graphique utilisé pour décrire les opérations séquentielles. Le procédé est représenté comme une suite connue d"étapes (états stables), reliées entre elles par des transitions. Une condition booléenne est attachée à chaque transition.

Les actions dans les étapes sont décrites avec les langages ST, IL, LD ou FBD

✓ **Structure de base** : Nous pouvons avoir dans un cycle machine complet avec des séquences simultanées, ou des choix de séquence.
✓ **Divergence et convergence en ET** :

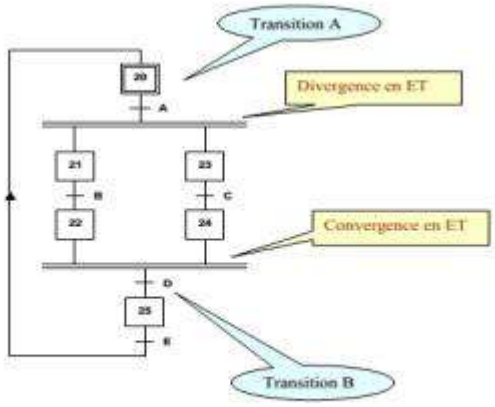


Fig (II.6): Grafcet Divergence et convergence en ET

- **Divergence en ET** : représentation par 2 trait identique et parallèle ; lorsque la transition A est franchie les étapes 21 et 23 sont actives. • **Convergence en ET** : La transition D sera active lorsque les étapes 22 et 24 seront actives, si la réceptivité associé à la transition D est vraie alors elle est franchie et l'étape 25 devient active et désactive les étapes 22 et 24. Le nombre de branche peut être supérieur à 2, après une divergence en ET on trouve une convergence en ET.

Divergence et convergence en OU (aiguillage).

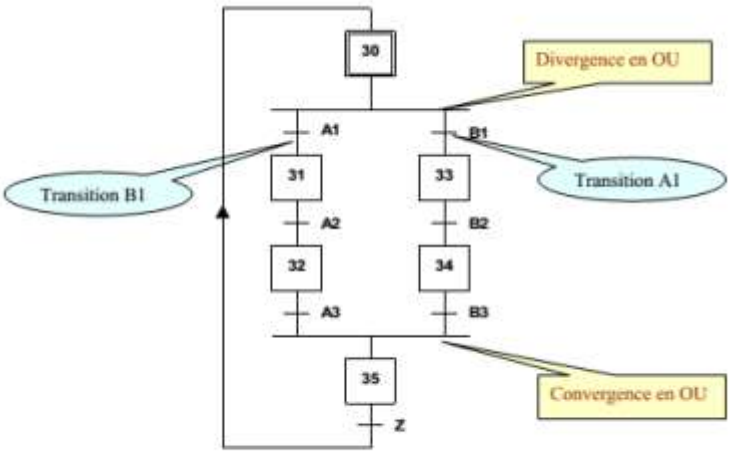


Fig (II.7): Grafcet Divergence et convergence en OU

- **Divergence en OU** : l'évolution du système se dirige vers une des branches en fonction des réceptivités A1, B1 et de leurs transitions associées.
- **Convergence en OU** : Après une divergence en OU on trouve une convergence en OU vers une étape commune dans l'exemple l'étape 35. Le nombre de branche peut être supérieur à 2, A1 et B1 ne peuvent pas être vrais simultanément.

✓ **Saut d'étape** : Le saut d'étape permet de sauter une ou plusieurs étapes en fonction de la progression d'un cycle

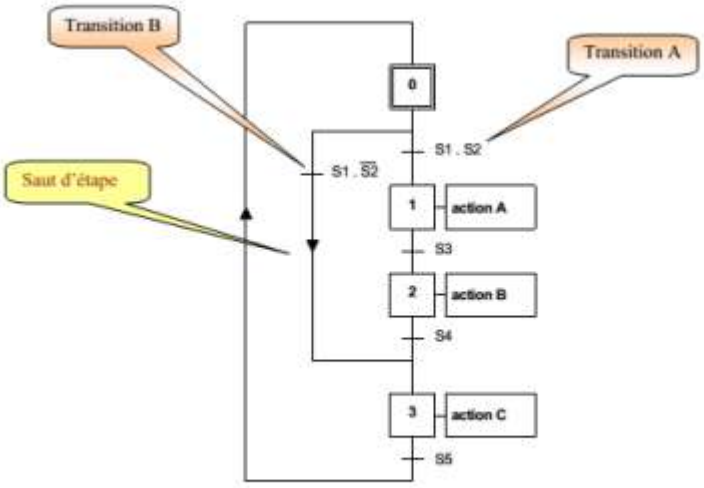


Fig (II.8): Grafcet Saut d'étape

Sur le Grafcet ci-dessus après l'étape initiale 0 un choix entre 2 transitions A et B s'effectue ; La transition A associé à sa réceptivité nous permet de continuer le cycle sur l'étape 1, La transition B associé à sa réceptivité nous permet de passer à l'étape 3, les étapes 1 et 2 sont ignorées lors du cycle.

✓ **Reprise d'étape** : La reprise d'étape permet de ne pas continuer le cycle mais de reprendre une séquence précédente lorsque les actions à réaliser sont répétitives.

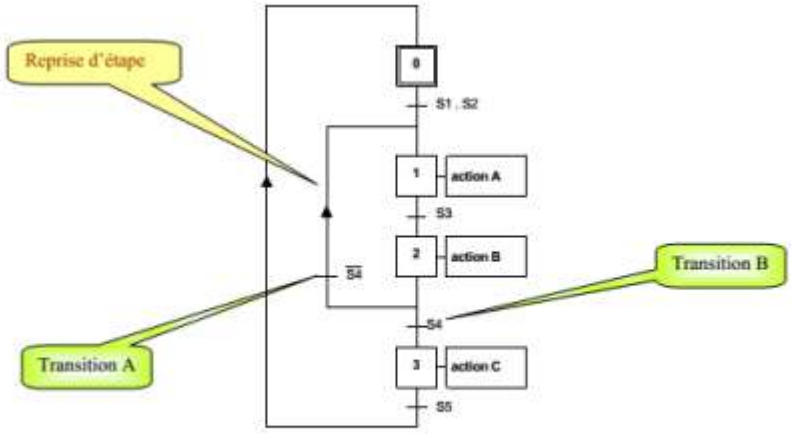


Fig (II.9): Grafcet Reprise d'étape

Sur le Grafcet ci-dessus après l'étape 2 un choix entre 2 transitions A et B s'effectue ; La transition A associé à sa réceptivité nous permet de reprendre le cycle sur l'étape 1, La transition B associé à sa réceptivité nous permet de passer à l'étape 3.

On peut également traduire un Grafcet en langage en contacts et l'implanter surtout type d'automate. Il existe d'autres langages de programmation des automates qui sont normalisés au plan mondial par la norme CEI 61131-3.

II.6.2. Langage à contacts (LD : Ladder Diagramme) : Le langage LD (ladder diagramme) est une représentation graphique d'équations booléennes combinant des contacts (en entrée) et des relais (en sortie). Il permet la manipulation de données booléennes, à l'aide de symboles graphiques organisés dans un diagramme comme les éléments d'un schéma électrique à contacts. Les diagrammes LD sont limités à gauche et à droite par des barres d'alimentation. Il est développé pour les électriciens. Il utilise les symboles tels que : contacts, relais et blocs fonctionnels et s'organise en réseaux (labels).

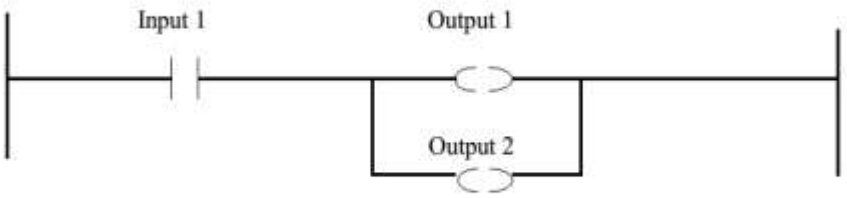


Fig (II.10): Exemple Programme LD avec de liaison multiple à droite.

II.6.3. Langage Blocs Fonctionnels (FBD : Function Bloc Diagram) :

Le langage FBD (function block diagram) est un Langage graphique ou des fonctions sont représentées par des rectangles avec les entrées à gauche et les sorties à droites. Les blocs sont programmés (bibliothèque) ou programmables. Utilisé par les automaticiens

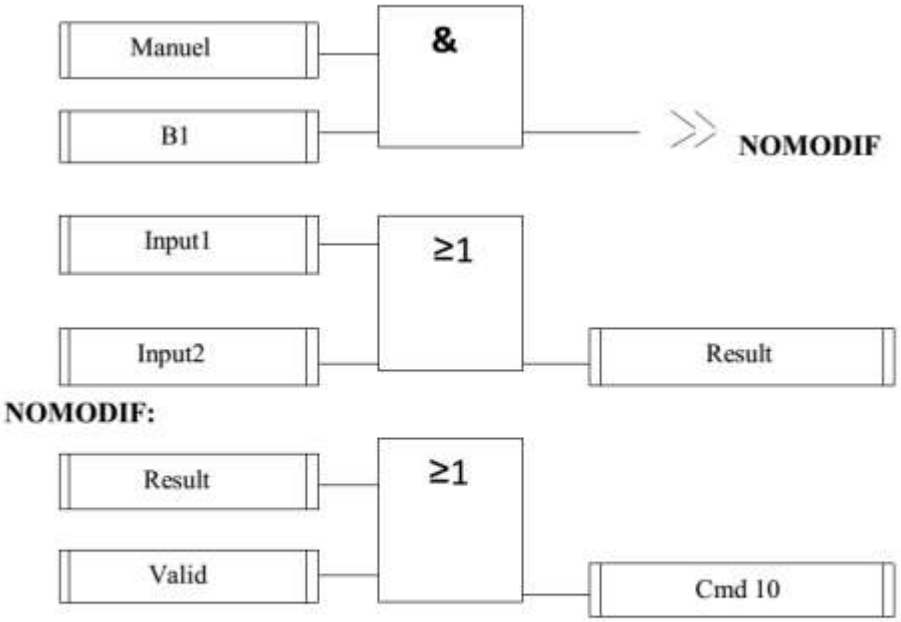


Fig (II.11): Exemple Programme FBD avec de saut et d'étiquette

II.6.4. Langage ST :

Le langage ST (structured text) est un langage textuel de haut niveau dédié aux applications d'‟automatisation. Ce langage est principalement utilisé pour décrire les procédures complexes, difficilement modélisables avec les langages graphiques. C'est le langage par défaut pour la programmation des actions dans les étapes et des conditions associées aux transitions du langage SFC.

Q: =false;

CV: =0

RETURN ;(*fin du programme*)

End if;

If RESET then

CV: =0;

```
Else
If (CV< PV)then
CV: = CV+1;
End if;
End if;
Q: = (CV>=PV);
```

Fig (II.12) : Exemple Programme ST

II.7. L'environnement de l'API

Pour conduire sa mission vis-à-vis d'un procédé, un API doit disposer des liaisons connectée Temporairement ou non, directement ou non, à l'API. Ces dispositifs peuvent être, par exemple, les périphériques et les auxiliaires des API (l'imprimante, les mémoires de masse…), les consoles d'exploitations, etc.…

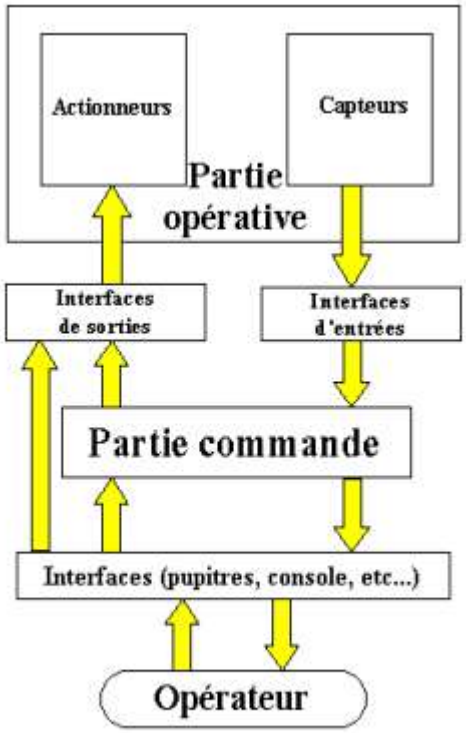


Fig (II.13): Structure d'un système automatisée

II.8. Critères de choix d'un automate :

Le choix d'un automate programmable est basé sur plusieurs points, nous citons :

- ✓ Le personnel de maintenance et le logiciel de programmation (achat du logiciel et formation du personnel).
- ✓ Nombre d'entrées /sorties.
- ✓ Type de processeur : la taille mémoire, la vitesse de traitement et les fonctions spéciales offertes par le processeur permettant le choix dans la gamme souvent très étendue.
- ✓ Fonctions ou modules spéciaux : certaines cartes 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, Ethernet...).

II.9. Les avantages et les inconvénients :

- Avantage :
- ✓ Améliorer les conditions de travail en éliminant les travaux répétitifs et les tâches pénibles.
- ✓ Améliorer la productivité en augmentant la production.
- ✓ Amélioration de la qualité des produits et réduction des coûts de production. Les automates programmables sont facilement programmables et ont un langage de programmation facile à comprendre alors la modification du programme est facile par rapport à la logique câblée.
- ✓ Simplification du câblage. ⎫ Augmenter la sécurité en effectuant les tâches dangereuses.
- ✓ La précision.
- ✓ La flexibilité (une machine peut s'adapter à plusieurs productions).
- ✓ Possibilité de communication avec l'extérieur (ordinateur, autre API).
- ✓ Plus économique.

- Inconvénients
- ✓ Plantage
- ✓ Il y a trop de travail requis dans les fils de connexion.
- ✓ Besoin de formation d'un personnel plus qualifié.
- ✓ Incidence sur l'emploi (licenciement – chômage).
- ✓ Une importante consommation d'énergie.

Ⅱ.10 L'automate zelio :

L'**automate Zelio** est une solution assez simple pour automatiser un équipement de par sa facilité de câblage et de mise en œuvre. Le **module ZelioLogic** propose des entrées et sorties directement intégrées dont le nombre et le type dépendra de la référence de l'appareil.

II.11 Découvert l'automate zelio :

L'automate Zelio de marque programmable regroupant dans un seul bloc les **Schneider Electric** est un seul bloc les entrées et les sorties. En fonction de la **référence de l'automate**, le type et la tension d'alimentation seront différents, il en va de même 3pour la tension des entrées. Les sorties quant à elles seront à relais jusqu'à 6 A.

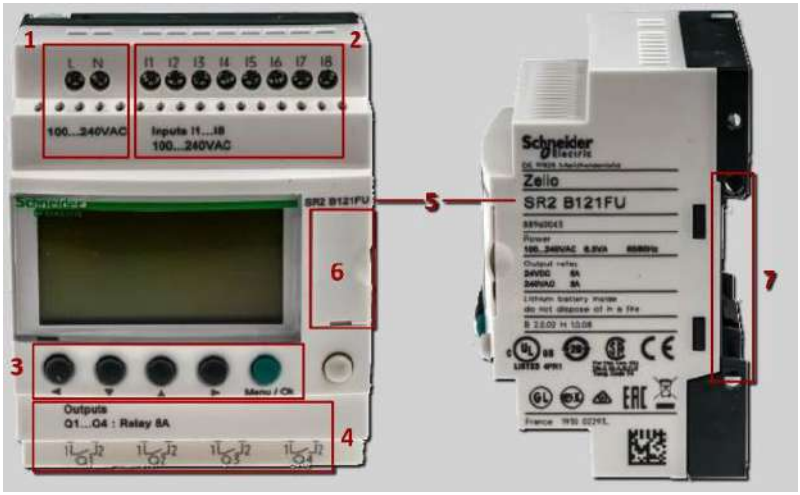


Fig (Ⅱ.14) :L'automate zelio.

II.12 De quoi est constitué l'automate Zelio?

• **1** Bornes à vis d'alimentation de l'automate (la tension varie en fonction de la référence de l'appareil).

• **2** Bornes à vis d'entrées du module Zelio (le nombre et la tension supportée dépendent de la référence de l'automate).

• **3** Touches de navigation et de sélection ou de validation.

• **4** Bornes à vis de sorties à relais de l'automate (leurs nombres dépendent de la référence du module Zelio).

• **5** Référence de l'automate Zelio.

• **6** Emplacement mémoire et prise de raccordement pour le câble de programmation à l'ordinateur.

• **7** Montage du module sur rail DIN [9]

II.13 Domaines d'utilisation de zilio :

zelio sont destinés à la réalisation de petits équipements d'automatismes il sont utilisées dans les secteurs d'activité de l'industrie et du tertiaire.

Ⅱ.14 Lgiciel zelio soft 2

F) Définition :

Le logiciel de programmation Zelio Soft 2 est conçu pour programmer les modules logiques de la gamme ZelioLogic.

Le langage zelio soft permet :

-la programmation en langage à contacts (LADDER) ou langage à blocs fonctions (FBD).

- la simulation, monitoring et la supervision.

- la chargement et le déchargement de programmes.

-l'édition de dossiers personnalisés.

-La compilation automatique de programme.

-L'aide en ligne.

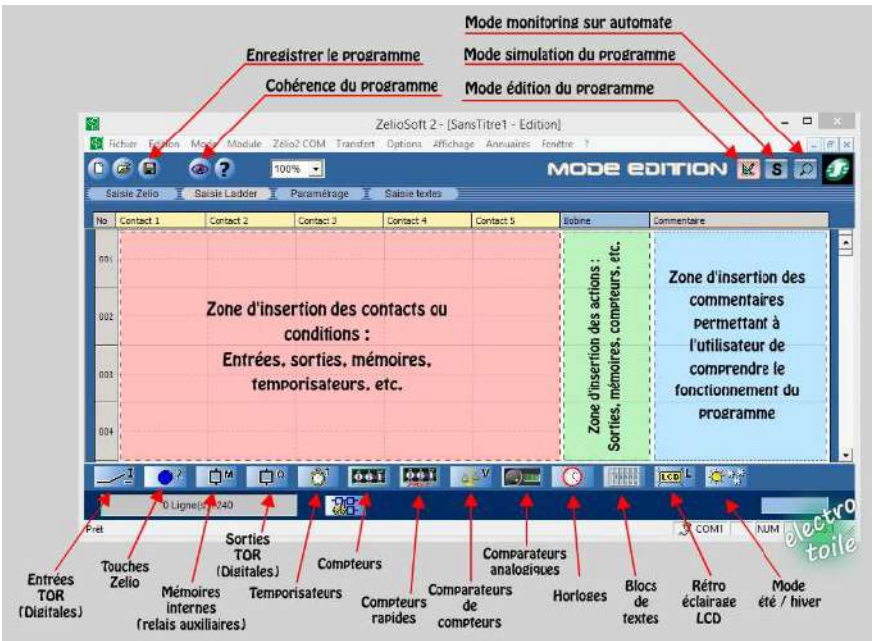


Fig (Ⅱ .15) : utilisé zlio softe [10]

II.15 Modes de programmation

G) **Modules logiques avec écran :**

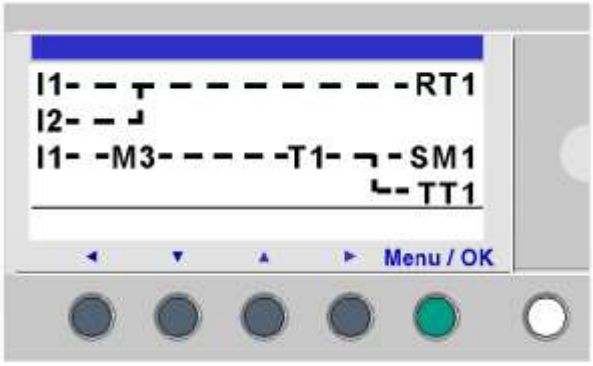
Il existe deux manières d'aborder la programmation des modules logiques avec écran :

- A partir de la face avant du module logique

Cette approche s'adresse aux personnes habituées à programmer directement sur des

-modules logiques.

NOTE : possible uniquement en mode L



Fig(Ⅱ.16) : Modules logiques avec écran [10]

H) **Modules logiques sans écran :**

Les modules logiques sans écran ne sont programmables qu'à partir de l'atelier de programmation

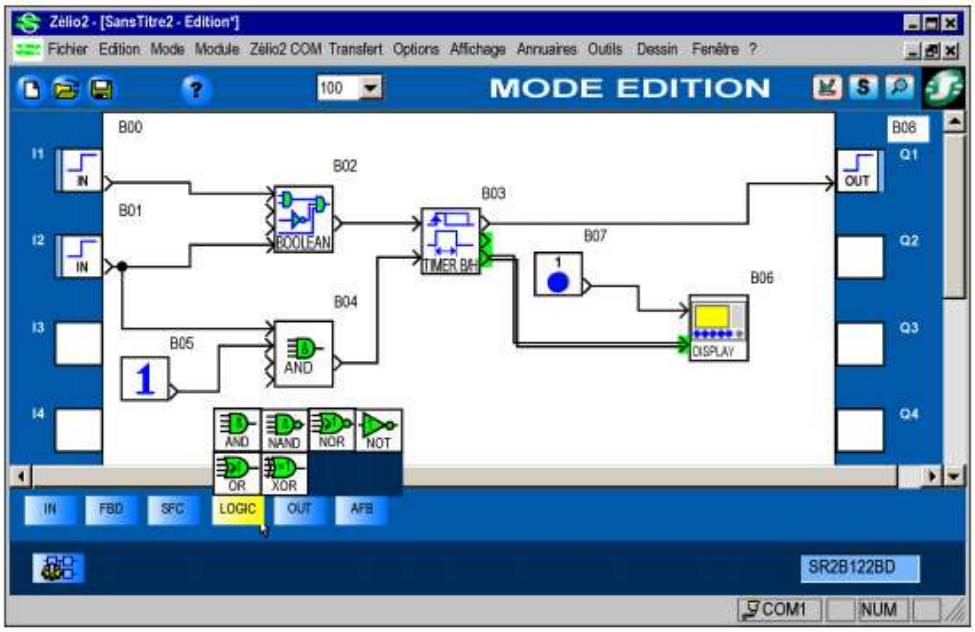


Fig (Ⅱ.17) : Modules logiques sans écran [10]

Ⅱ.16 Langages utilisés

Le module logique propose 2 modes de programmation :

- Mode LD : langage à contacts
- Mode FBD : langage en blocs fonction

Ces langages mettent en œuvre :

- des blocs fonction prédéfinis :

-temporisateurs

- compteurs
- des fonctions spécifiques :
- gestion du temps

-chaîne de caractères

-communication.

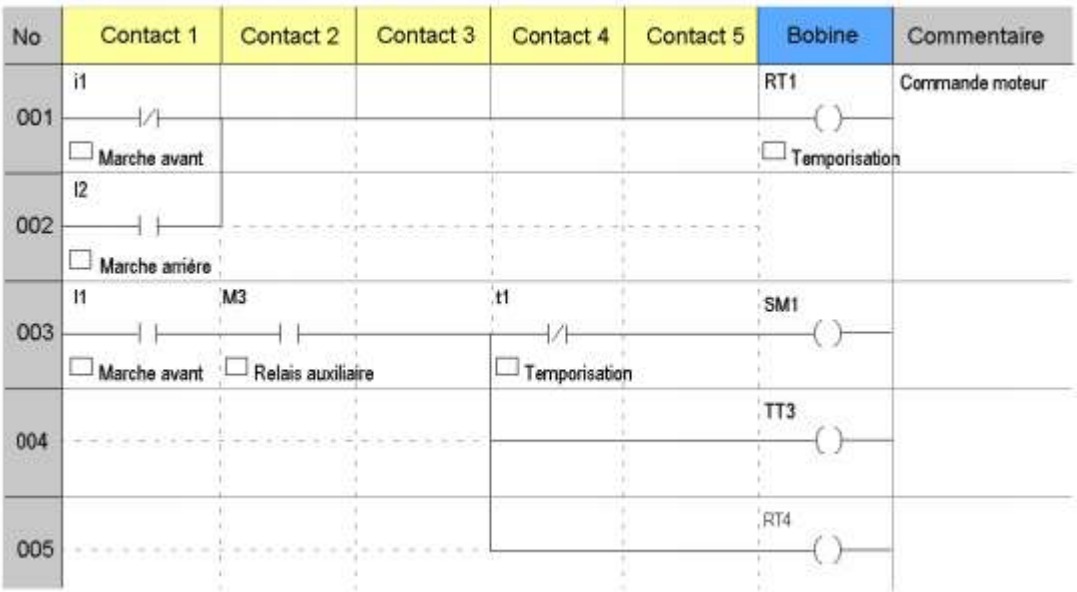
A) **Langage à contacts**

Le langage à contacts (LD) est un langage graphique. Il permet la transcription de schémas à relais et il est adapté au traitement combinatoire.

Il fournit des symboles graphiques : contacts, bobines et blocs.

L'exécution de calculs spécifiques est possible à l'intérieur des blocs opérations.

L'exemple suivant décrit un programme en langage à contacts dans ZelioLogic 2 :



Fig(Ⅱ.18) : Langage à contacts [10]

Le nombre maximum de lignes en langage à contacts est :

-120 lignes si une interface de communication SR2COM01 est sélectionnée dans la configuration.

B) **Mode FBD**

Le mode FBD permet une programmation graphique basée sur l'utilisation de blocs fonction prédéfinis.

Il offre une large gamme de fonctions : temporisateur, compteur, logique, etc.

Exemple de programme en FBD.

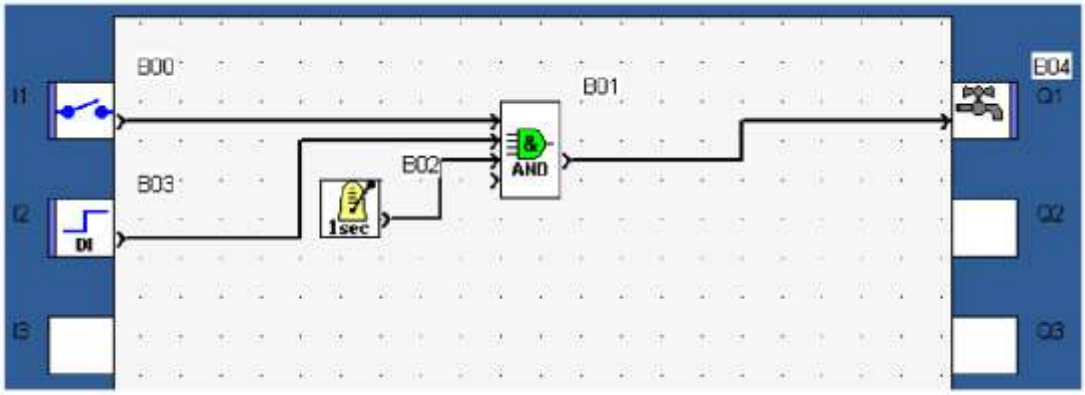


Fig (Ⅱ .19): Mode FBD [10]

II.17 Mode de fonctionnement

L'atelier de programmation autorise deux modes de fonctionnement :

A) **Mode saisie**

Le mode saisie sert à construire les programmes en langage LD ou FBD et correspond au développement de l'application.

B) **Mode mise au point**

Ce mode permet de finaliser l'application. Cette opération peut s'effectuer :

- En mode Simulation : le programme est exécuté en local directement dans l'atelier de programmation (simulé sur le PC).

Dans ce mode, chaque action sur le graphe (changement d'Etat d'une entrée, forçage d'une sortie) met à jour les fenêtres de simulation.

- En mode Monitoring : le programme est exécuté sur le module logique et l'atelier de programmation est connecté au module.

Les différentes fenêtres sont mises à jour cycliquement.

Dans ces deux modes, il est possible :

- d'afficher dynamiquement (dans les fenêtres Edition/Supervision/Face avant) l'état des sorties et les blocs fonction du programme Correspondant à la feuille de câblage ;

-de forcer les entrées/sorties pour tester le comportement du programme dans des conditions particulières. [10]

Ⅱ.18 Test des programmes

2modes de test sont proposés : simulation et monitoring.

Le mode **simulation** de « zelio soft » permet de tester l'ensemble des programmes sans module c'est-à-dire :

-activer les entrées « tout ou rien » (TOR).

-visualiser l'état des sorties.

-faire varier la tension des entrées analogiques.

-activer les touches de programmation.

-simuler le programme applicatif en temps réel ou en accéléré.

-visualiser en dynamique et en rouge les différents éléments actifs du programme.

Le mode **monitoring** de « zilio soft » permet de tester le programme exécuté par le module, c'est-à-dire :

-visualisé « en ligne » le programme.

- forcer les entrées, les sorties, les relais auxiliaires et les valeurs courantes des blocs fonctions.

- régler l'heure.

-passer du mode d'arrêt (STOP) au mode de marche (RUN) et inversement.

En mode simulation ou monitoring, la fenêtre de supervision permet de visualiser l'état des entrées/sorties du module dans l'environnement de votre application (dessin ou image). [10]

Ⅱ.19 Conclusion

L'automate programmable industriel a fait un bond en avant dans le domaine du contrôle de sorte que la plupart des fonctions industrielles opèrent sous le contrôle de cet élément. Dans ce chapitre, nous avons expliqué en détail ses propriétés, ses types, tous ses composés et ses méthodes de programmation

Le logiciel de portail d'automatisation entièrement intégré (ZELIO SOFT2), comme nous l'avons évoqué Sur les langages de programmation FBD et GRAFCET.

CHAPITRE Ⅲ

Réalisation pratique

Ⅲ.1. Introduction :

Dans ce chapitre, nous détaillons les étapes de programmation que nous avons réalisées et expliquons ses différentes étapes, en plus de l'expérience du travail que nous avons réalisé.

Ⅲ.2. Les entrées de l'automate zelio :

- I1 : correspond à l'entrée reset lors d'une défaillance sur le système on fait monté le système à l'étage 0.

- Et tournée le moteur à la position 0 puis en actionneur sur le bouton reset

- I2 : Entrée E (E signifier entrée) lors d'une demande d'entrée d'un véhicule ou est obligé d'appuyer sur l'entrée I2.

- I3 : Sortie S(S signifier sortie) lors d'une demande de sortie d'un véhicule ou est obligé d'appuyer sur l'entrée I3.

- I4, I5, I6, I7, I8 : correspond au clavier d'entrée clavier d'étage.

- I8 : correspond à la commande mémorisation de l'étage.

- I9 : capteur d'horloge vertical.

- IA, IB, IC, ID : clavier de chambre.

- IE : horloge de rotation.

- R : résistance pull drun.

Ⅲ.3. Les sorties de l'automate zelio :

Q1 : attaque le contacteur KM1 du moteur 1 pour faire monter l'ascenseur.

Q2 : attaque le contacteur KM2 du moteur 1 pour faire descendre l'ascenseur.

Q3 : attaque le vérin pour mettre le véhicule dans la chambre.

Q4 : attaque le vérin pour sortie le véhicule de chambre.

QB : attaque le KM3 pour faire tourner le moteur M2 dans le sens droit.

QC : attaque le KM4 pour faire tourner le moteur M2 dans le sens inverse.

Programme de monté et d'exécuter de véhicule SSss.

Ⅲ.4. Réalisations de clavier d'étage :

Ⅲ.4.1 Clavier d'étage

On utilise un clavier de 16 bouton (0 ,1 ,2 ,3,4,5,6,7,8,9,10,11,12,13,14,15) , pour chaque étage de parking on a un bouton et 4 sortie (I4, I5, I6, I7, I8).

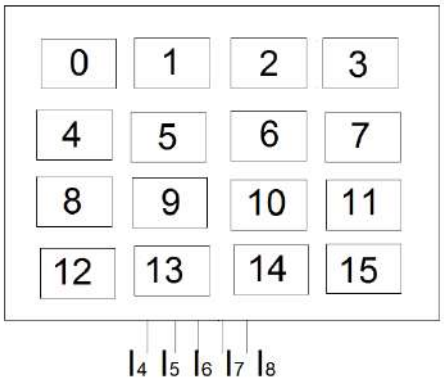
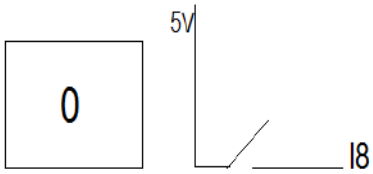
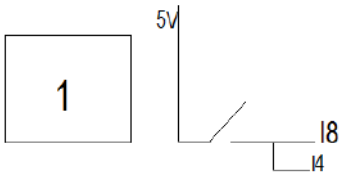


Fig (Ⅲ.1) : clavier d'étage.

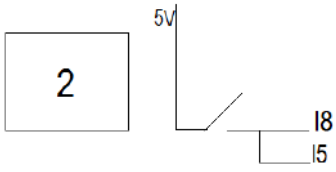
Chaque étage réalisé comme suite :



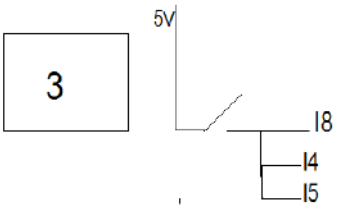
Étage 0 : 0000



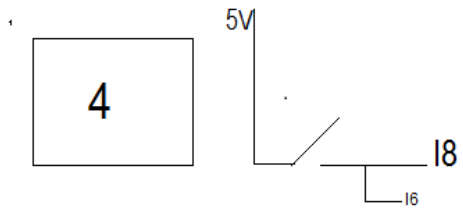
étage1 : 0001

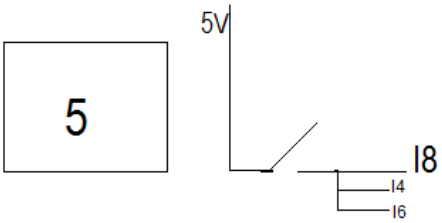


Etage2 : 0010

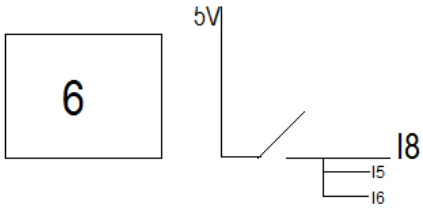


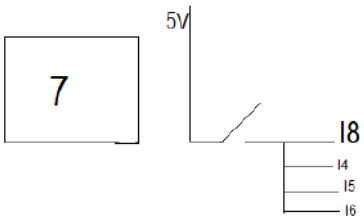
étage3 : 0011



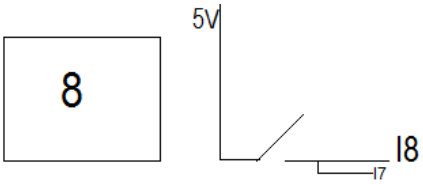


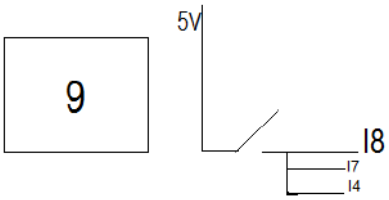
Etage 4 : 0100 étage5 : 0101



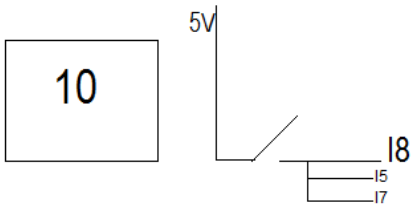


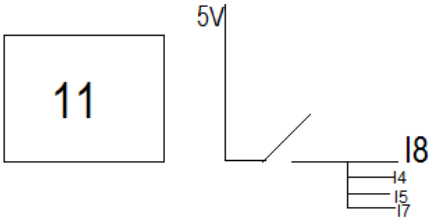
Etage 6 : 0110 étage 7 : 0111



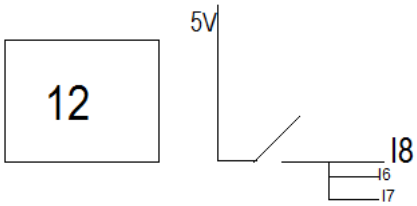


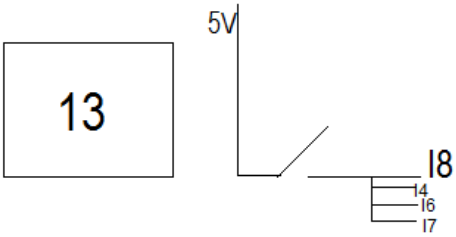
Etage 8 : 1000 étage 9 : 1001



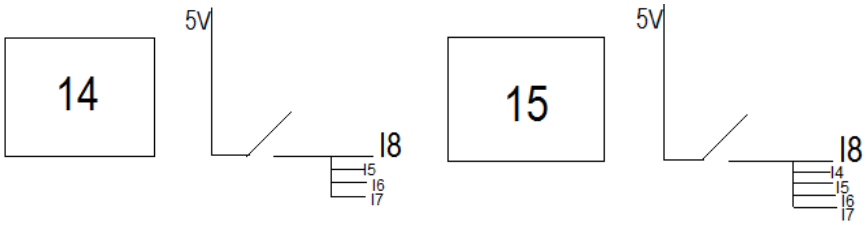


Etage 10 : 1010 étage 11 : 1011





Etage 12 : 1100 étage 13 : 1101



Etage 14 :1110 étage 15 : 1111

Ⅲ.4.2. Horloge d'étage :

_IQ : capteur d'horloge vertical.

-R : Résistance pull Drun.

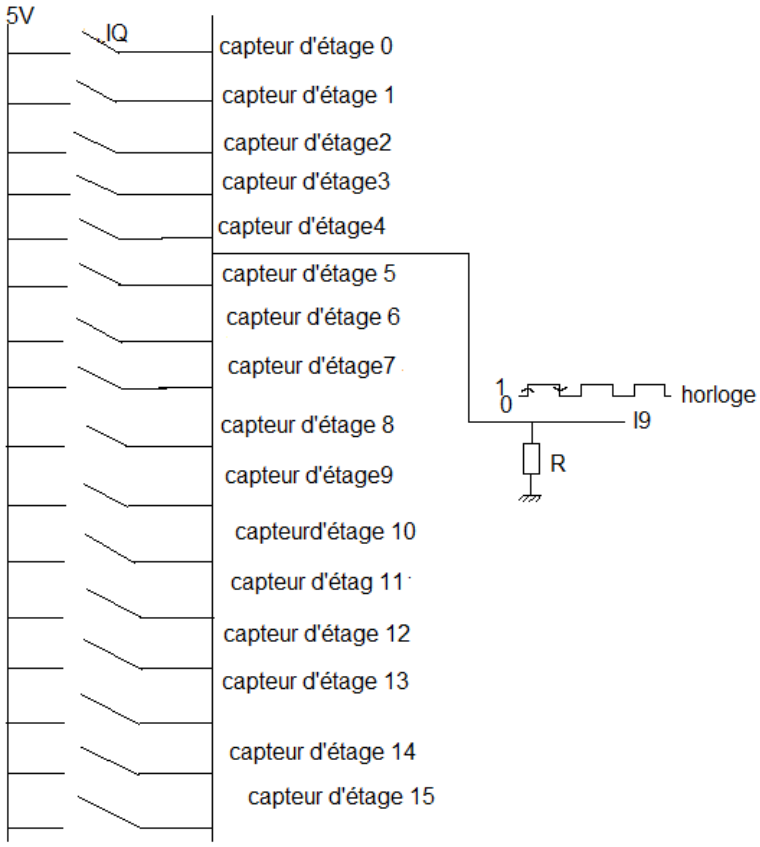


Fig (Ⅲ.2) : horloge d'étage.

Ⅲ.5. Réalisation de clavier de chambre :

Ⅲ.5.1 Clavier de chambre:

On utilise un clavier de 8 bouton (0, 1, 2, 3, 4, 5, 6,7), pour chaque chambre d'étage de parking on a un bouton et 4 sorte (IA, IB, IC, ID)

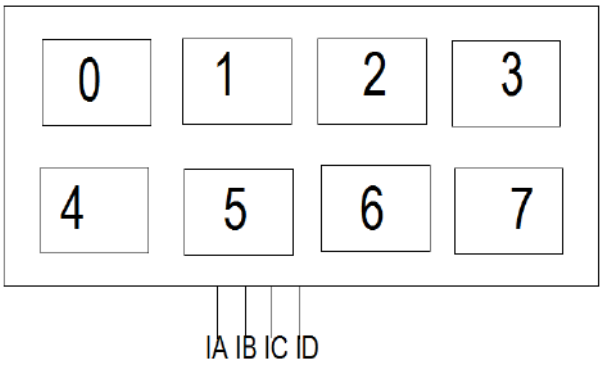
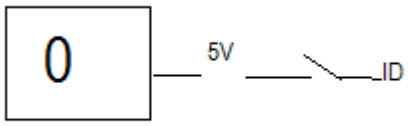
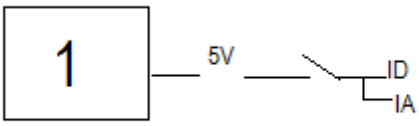


Fig (Ⅲ.3) : clavier de chambre.

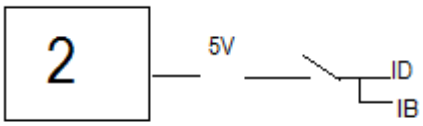
-chaalique chambre choisie poure le nombre entré dan le clavier :

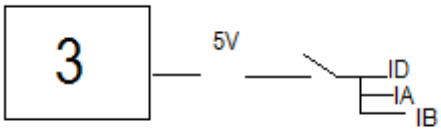




Chambre 0 : 0000

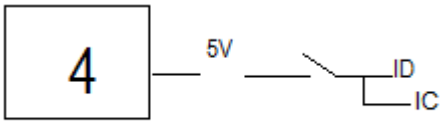
chambre 1 : 0001

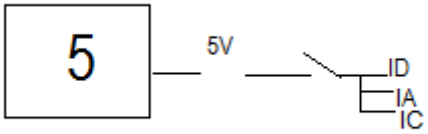




Chambre2 : 0010

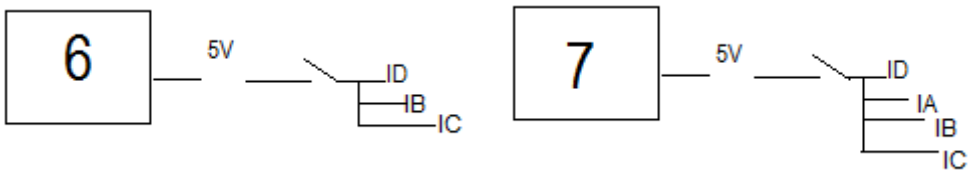
chambre3 : 0011





Chambre 4 : 0100

chambre5 : 0101



Chambre 6 : 0110 chambre7 : 0111

Ⅲ.5.2 horloge de chambre :

-IE : horloge de rotation

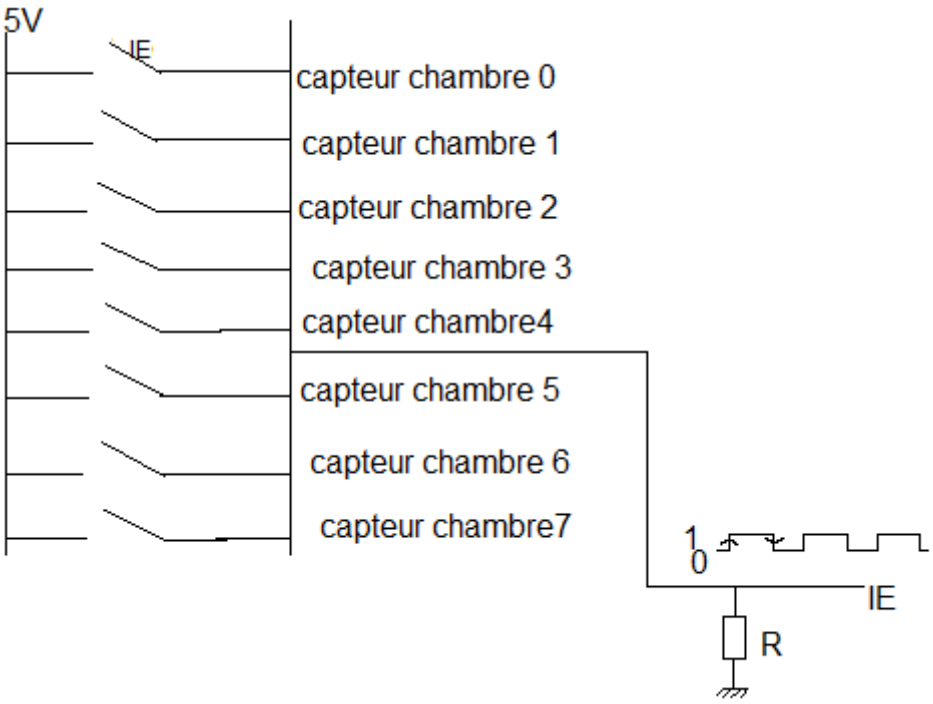


Fig (Ⅲ.4) : horloge des chambres.

Ⅲ .6. Circuit de commande et puissance pour

Moteur asynchrone :

On utilisé un circuit de puissance pour un moteur M1, avec deux sens de marche (droite et gauche)

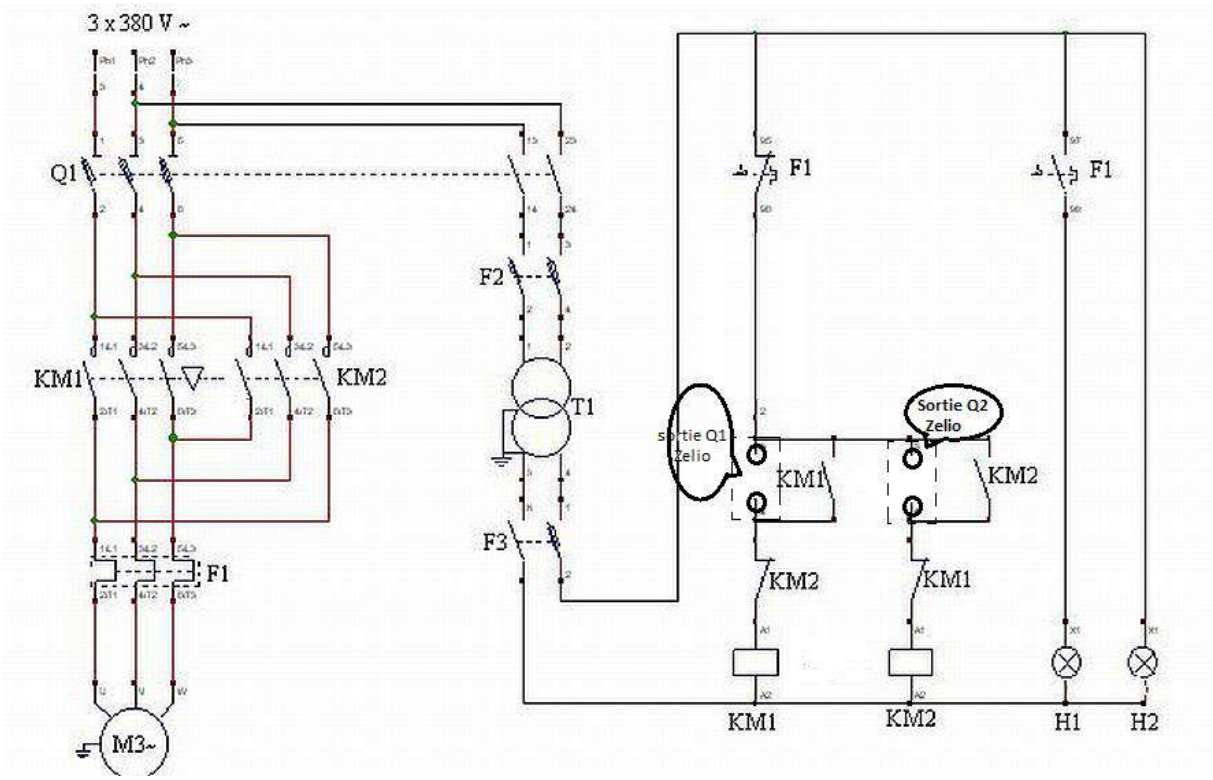


Fig (Ⅲ.5) : circuit de puissance de moteur M1 avec deux sens de marche.

NB : de la même manière en commande le moteur M2 sauf qu'on doit utiliser les sortie correspondante de l'automate zelio.

Ⅲ.7. Circuit de commande vérin :

Afin de faire rentrer et sortir le véhicule dans la chambre correspondante on utilise un vérin hydraulique figure….

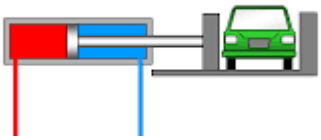


Fig (Ⅲ.6) : technique de rentrer et sortir le véhicule

On utilise les circuits hydrauliques car sont caractérisés par leurs puissances par rapport au circuit pneumatique.

Pour commander ce vérin double effet on utilise un distributeur 5/3 bistable a commande électrique et a centre fermé et a figure ….

On utilise le centre fermé pour des raisons de sécurité.

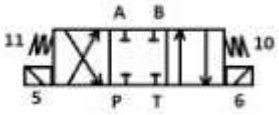


Fig (Ⅲ.7) : un distributeur 5/3 bistable à commande électrique et a centre fermé

NB : pour le détail de ce distributeur voire Annex 1

La figure Ⅲ.8 Montre le circuit de commande de vérin double effet par le distributeur 5/3 a centre fermé avec la commande Q3 et Q4 de module zelio et

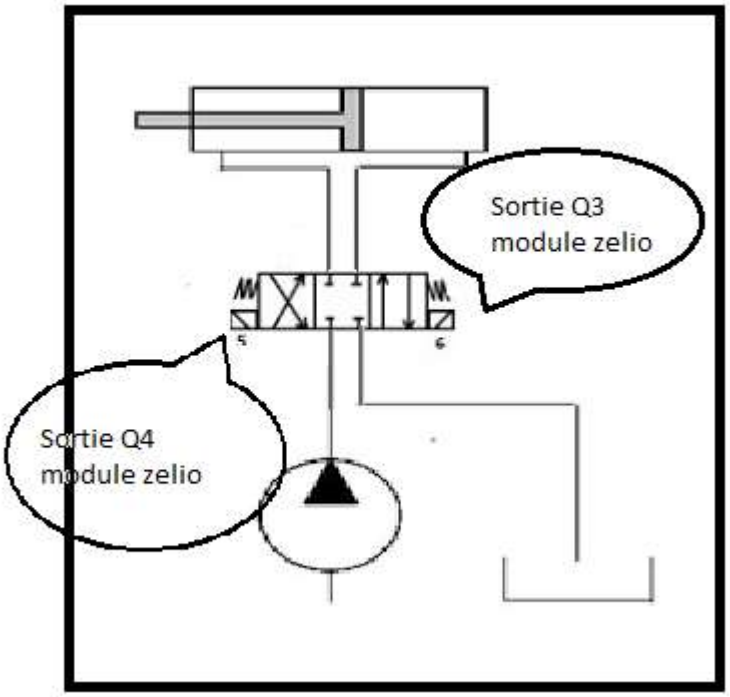


Fig (Ⅲ.8) : circuit de commande de vérin double effet par le distributeur 5/3 a centre fermé avec la commande Q3 et Q4 de module zelio

X) Fonctions Zelio

Pour la réalisation de ce programme on a utilisé quelque fonction zelio

X-1) la fonction CAM : voir Annex 2

La figure Ⅲ.9 : Montre la fonction CAM

Cette fonction sert à choisir un certain nombre de position programmé suivant le nombre de sortie de l'automate, ainsi suivant les entrés avancer et reculer on peut aller la position voulue

L'entre remise à zéro remis le module CAM a la position 1

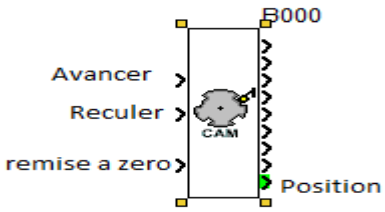


Fig (Ⅲ .9) : la fonction CAM

La figuⅢ.10 montre le choix de nombre de position

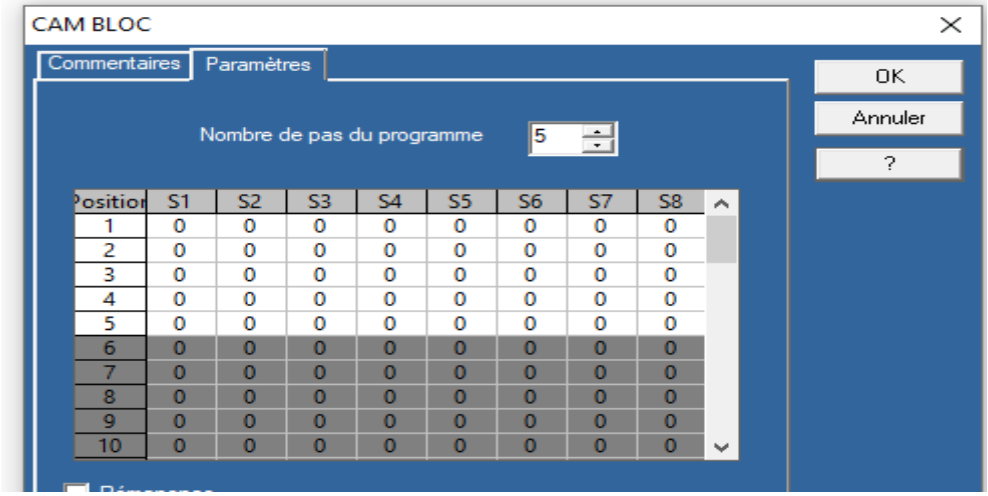


Fig (Ⅲ.10) : choix de nombre de position

X-2) la fonction Archive : voir Annexe 3

La fonction Archive permet de sauvegarder deux valeurs simultanément avec les informations relatives à leur datation figure…..

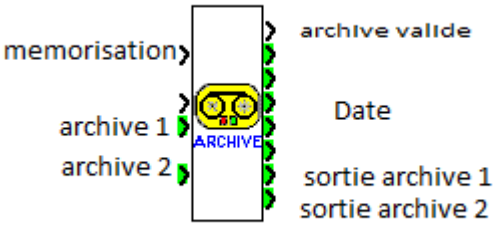


Fig (Ⅲ.11) : La fonction Archive

Ⅲ.8. Le programme de Parking :

Le programme de parking se divise en trois parties

Ⅲ.8.1 monté et décente verticalement (ascenseur):

Nous avons choisi un parking à 16 étages

Ⅲ.8.2 Programme de rotation:

On a choisi 08 chambres par étage

L'accès vers la chambre se fait par la rotation droite au gauche a l'aide de moteur M2 de rotation.

Le programme minimise le chemin de parcours vers la droite ou vers la gauche.

Le programme de parking se divise en trois parties

Ⅲ.8.3 Programme d'entrée et sortie le véhicule:

On choisit un vérin hydraulique commandé par un distributeur 5/3 à centre fermé

Déroulement du programme

- ✓ **Lors de l'entrer d'un véhicule :**
- À l'aide d'un clavier on donne l'étage de destination**.**
- À l'aide d'un second clavier on donne la chambre de destination**.**
- Le programme remonte à l'aide de moteur M1 vers l'étage 0
- Le programme fait sortir le vérin pour prendre le véhicule
- Le programme fait rentre le vérin.
- Le programme fait descendre le véhicule vers l'étage demandé ainsi et en parallèle il fait la rotation vers la chambre de destination.
- Le vérin sort de nouveau pour placer le véhicule dans la chambre
- Le vérin revient en position
- Le programme archive les positions verticales (étage) et la potion horizontale (la chambre).

- ✓ **Lors de sortie d'un véhicule :**
- Les claviers donnent les positions étage et chambre destinée
- Le programme optimise le sens de rotation de moteur M2 et tourne soit à droite ou à gauche vers la destination demandé et mémorise cette position.
- Le programme calcule suivant la position demandée et la position archivé de l'étage et fait monter ou descendre par le biais de moteur M1.
- Le vérin sort de nouveau pour tirer le véhicule de la chambre

- Le vérin revient en position du centre.
- Le moteur M1remonte le véhicule vers l'étage 0
- Le vérin sort de nouveau pour sortir le véhicule.
- Le vérin revient en position du centre.
- Le programme archive cette position

La fig Ⅲ.12 montre le programme de parking

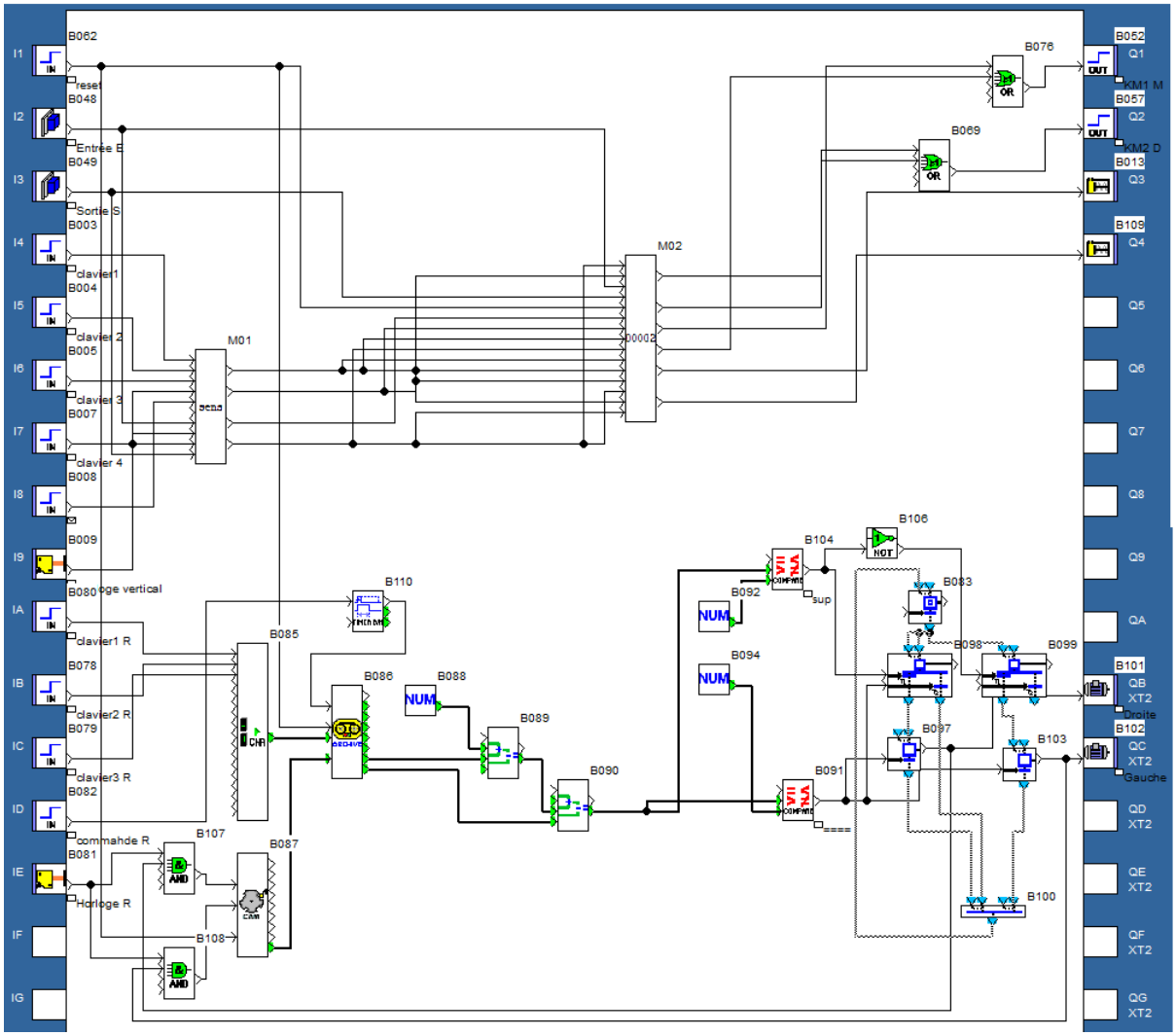


Fig (Ⅲ.12) : programme de parking

La figⅢ.13.montre le sous-programme Mo1

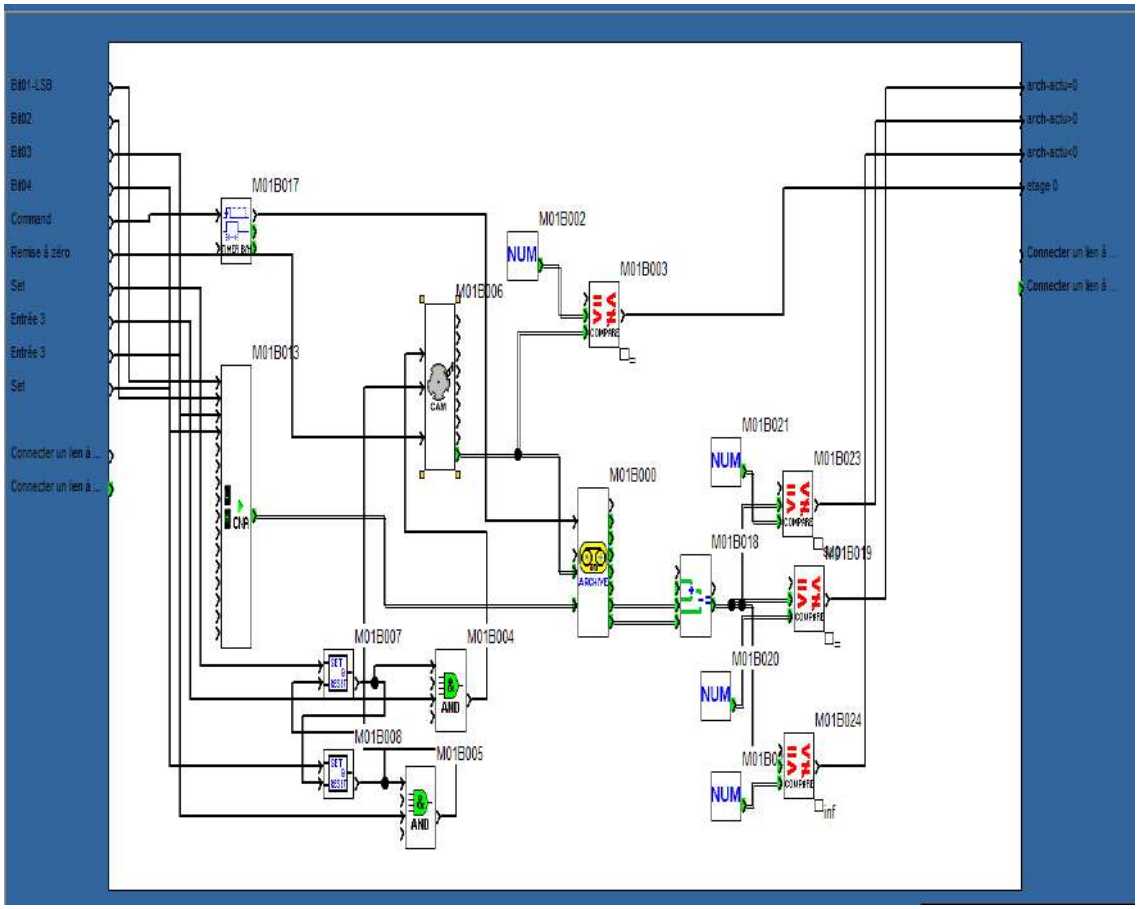


Fig (Ⅲ.13) : le sous-programme Mo1

La figⅢ.14 montre le sous-programme Mo2

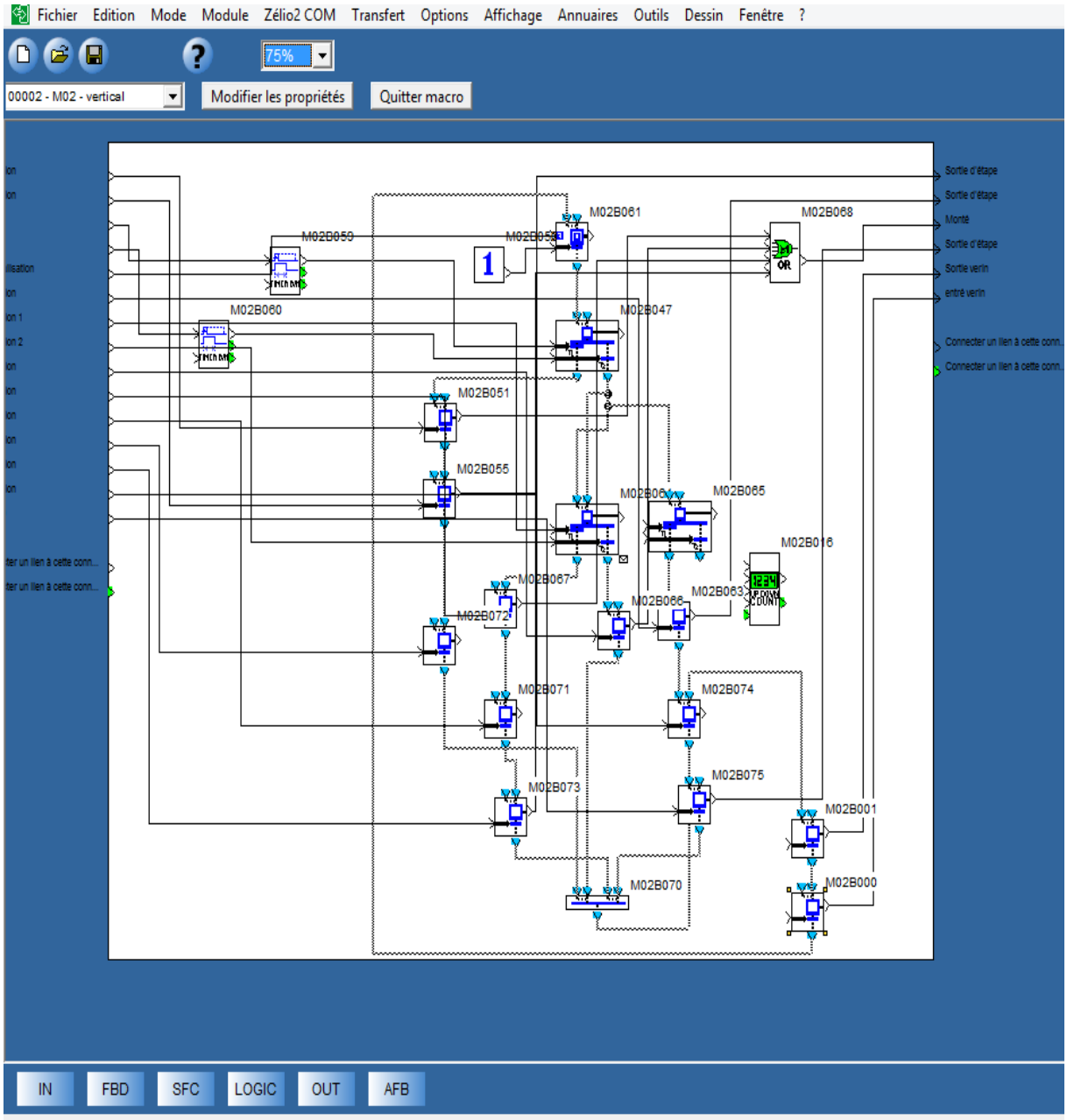


Fig (Ⅲ.14) : le sous-programme Mo2

Ⅲ.9 Conclusion:

Dans ce chapitre, Nous avons vue qu'à l'aide de langage FBD et les fonctions mathématiques de comparaison…etc. qui contient ce logiciel et surtout la fonction SFC (grafcet) utilisé pour manipuler chaque partie des mécanismes qui composent le parking.

L'aide de ce logiciel on a pu simuler le fonctionnement de parking.

CONCLUSION GENERALE

Notre projet consiste à trouver une solution aux problématiques de stationnement, le parking automatique est une alternative au garage classique, il tire un large profil de l'espace disponible.

L'automatisation répond d'une manière Idéale au système de stationnement, de point de vue précision, fiabilité, ainsi la gestion parfaite des entrées et de sortie.

Par conséquent, nous avons utilisé un automate Schneider (bloc logique) très performant ; ceci utilise un logiciel appelé zelio soft.

Parmi les points fort que possède ce logiciel est le langage FBD et dans ce langage en trouve plusieurs fonctions aptes de construire un système automatisé, ces fonctions sont logiques, séquentiels, mathématiques, électronique…etc.

Avec la combinaison de ces fonctions qu'on a automatisé notre système de parking, on a tracé un cahier de charge, avec une technologie limitée, ou la simulation a donné des bons résultats.

Ce travail possède beaucoup de perceptives qui seront peut-être amélioré par les futurs étudiants parmi eux on trouve :

- Un système de gestion de paiement.
- Un Visio (HMI).
- Un système intelligent afin de définir la disposition en temps réel de parking.

…etc.

Enfin ce projet de point de vue économique est très bénéfique, est rend service très remarquable surtout dans les zones à grandes agglomérations.

Finalement ce thème nous a permet de découvrir le monde de l'automatisme industriel, de découvrir cet automate de point de vue soft et hard, ainsi l'environnement mécanique, pneumatique ou hydraulique.

LISTE DE LA REFERENCE BIBLIOGRAPHIQUE

[1] http://parking.com/

[2]. Y Jog, A Sajeev et autre, Understanding smart and automated parking technology, International Journal of u- and e- Service, Science and Technology,

http : //dx.doi.org/10.14257/ijunesst.2015.8.2.25, Vol.8, No.2 (2015), pp.251-262

[3]. Clavel, E. Flambard et autre, Smart Parking, état de l'art, étude d'un exemple de système de mesure lié à ce sujet, Projet de physique p6 stpi/p6/2013 – 3, Institut national des sciences appliquées de rouen, France, 2013, pp. 3-5.

[4] https://www.parkingmap.fr/presentation/

[5] A. Albagul, K. Alsharef, Design and fabrication of an automated multi-level car parking system, Manufacturing Engineering, Automatic Control and Robotics, Libya

[6] http://google/photo-parking /

[7] http://google./photo -automate.etparking/

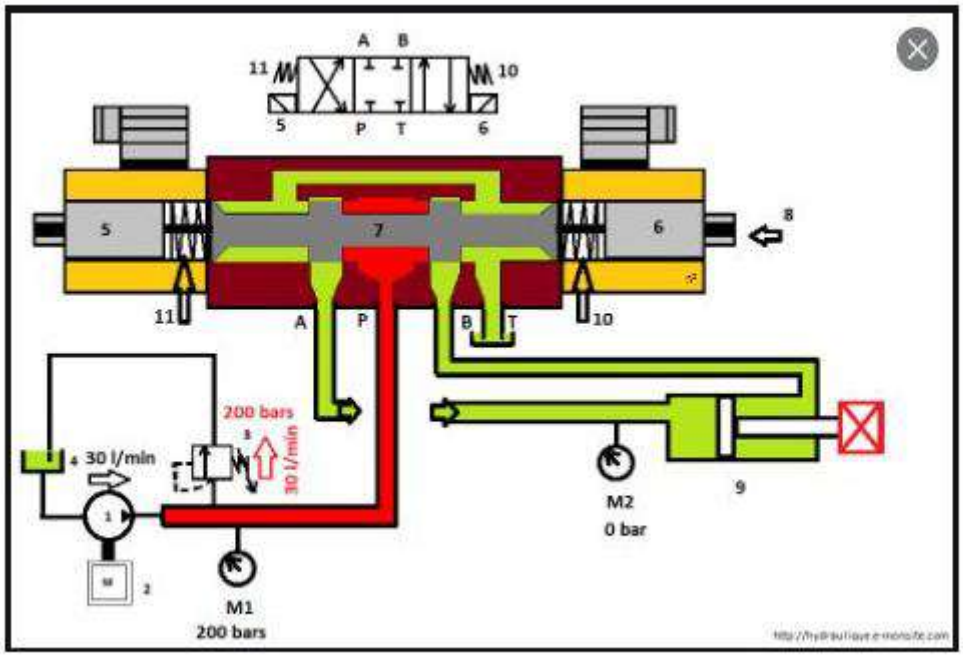
[8] www.technolguepres.com

[9] https://www-automation-science.com

[10] https://fasbeducation.net

ANNEX

Annex 1



Annex 2

Présentation

La fonction **CAM BLOC** contrôle un ensemble de 8 roues à cames solidaires.

La fonction présente sur ses 8 sorties (représentant les 8 roues) l'état correspondant à la position courante des roues de l'arbre.

La configuration de came est paramétrable ; pour chaque position, l'état des sorties est ajustable.

Une fois la valeur maximum atteinte, la came repart de sa position initiale (la sortie repasse à 0).

Accès

La fonction 

est accessible à partir de la barre de fonctions **FBD**.

Entrées/sorties

Description des entrées :

- **Avancer** : entrée qui commande l'avancement de la came ; celle ci avance d'un pas sur chaque front montant (passage d'inactif à actif).
- **Reculer** : entrée qui commande le retour en arrière de la came ; celle ci recule d'un pas sur chaque front montant (passage d'inactif à actif).

NOTE : l'entrée **Avancer** est prioritaire par rapport à l'entrée **Reculer**.

NOTE : si les entrées **Avancer** et **Reculer** ne sont pas connectées, elles sont définies sur l'état inactif.

- **Remise à zéro** (initialisation) : lorsque cette entrée est active, la came est remise à sa position initiale ; la sortie **Position** est forcée à 1.

NOTE : l'entrée **Remise à zéro** est prioritaire par rapport aux entrées **Avancer** et **Reculer**.

NOTE : si l'entrée **Remise à zéro** n'est pas connectée, elle est définie sur l'état inactif.

Description des sorties :

- **Sorties 1 à 8** : état correspondant à la position courante le l'arbre (représentant les 8 roues).
- **Position** : position de la came (1 à 50).

Paramètres

A partir de l'atelier de programmation

A partir de la fenêtre **Paramètres**, vous pouvez régler :

- **Le nombre de pas du programme** : de 1 à 50.
- **L'état des sorties 1 à 8** : pour chaque position de l'arbre.

La figure suivante illustre une partie de la fenêtre de paramétrage :

Nombre de pas du programme 3

Pas	S1	S2	S3	S4	S5	S6	S7	S8
1	0	0	1	1	1	0	0	0
2	1	1	0	0	0	0	1	1
3	0	1	0	1	0	1	0	1
4	0	0	0	0	0	0	0	0

Position de la came — Configuration des sorties pour chaque position de la came — Nombre de pas sélectionnés

Le paramètre `Latching` (Rémanence), s'il est sélectionné, permet de récupérer la valeur courante du compteur après une coupure secteur.

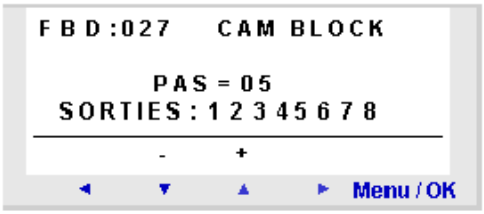
A partir de la face avant

A partir de la fenêtre PARAMETRES, il est possible de modifier bit à bit le contenu des pas du programmateur à cames, mais il n'est pas possible de modifier le nombre de pas.

Après avoir entré le numéro du bloc, il faut entrer :

- **Le numéro de pas** : valeur de 1 à 50.
- **L'état des sorties 1 à 8** : la valeur de chaque sortie peut être définie sur inactive (affichage normal du numéro) ou active (numéro en vidéo inverse).

Illustration :



Verrouillage des paramètres

Le verrouillage interdit la modification des paramètres du bloc fonction verrouillé à partir de la face avant du module logique par le menu **PARAMETRES**.

Annex 3

Présentation

La fonction **Archive** permet de sauvegarder deux valeurs simultanément avec les informations relatives à leur datation.

Accès

La fonction 

est accessible à partir de la barre de fonctions **FBD**.

Entrées/sorties

Description des entrées :

- **Mémorisation** : pour chaque entrée de commande de la fonction Archive (type TOR), sur chaque front montant (passage d'inactif à actif) l'entrée VALEUR est mémorisée.

NOTE : si l'entrée **Mémorisation** n'est pas connectée, elle est fixée à inactive.

- **Remise à zéro** : lorsque cette entrée (type TOR) est active, elle force la sortie **Archive valide** à inactive. les valeurs préalablement sauvegardées restent disponibles.

NOTE : si l'entrée **Remise à zéro** n'est pas connectée, elle est fixée à inactive.

- **Valeur 1** : première entrée qui est sauvegardée. La valeur présente sur cette entrée est sauvegardée avec des informations de datation : heure et date (ces informations sont disponibles sur les sorties).
- **Valeur 2** : seconde entrée sauvegardée.

NOTE : si une entrée **Valeur 1 ou 2** n'est pas connectée, elle est fixée à inactive.

Description des sorties :

- La sortie **Archive valide** (type TOR) indique la validité du stockage en cours :
  - Inactive : aucune donnée disponible
  - Active : données disponibles

NOTE : Les sorties suivantes sont des entiers.

- **Minute** : valeur de la minute dans la datation (0 à 59).
- **Heure** : valeur de l'heure (0 à 23).
- **Jour** : valeur du jour (1 à 31).
- **Mois** : valeur du mois (1 à 12).
- **Année** : valeur de l'année (0 à 99).
- **Archive 1** : valeur entière présente sur l'entrée **Valeur 1**.
- **Archive 2** : valeur entière présente sur l'entrée **Valeur 2**.

Paramètres

A partir de l'atelier de programmation

Le paramètre `Latching` (Rémanence), s'il est sélectionné, permet de récupérer la valeur courante du compteur après une coupure secteur.

Mécanisme de stockage

Si l'entrée **Mémorisation** est activée plusieurs fois, seules les données concernant la dernière activation sont mémorisées.

Affichage des valeurs sauvegardées

Il est possible de visualiser les valeurs sauvegardées ; pour cela, connectez les sorties de la fonction `Archive` aux blocs DISPLAY.
La fonction DISPLAY peut effectuer des modifications de la valeur affichée si le paramètre `Authorized modification` est coché.

NOTE : toute modification peut entraîner des incohérences dans les données archivées : Valeur/Date.