الجمهورية الجزائرية الديمقراطية الشعبية
République Algérienne démocratique et populaire

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

جامعة سعد دحلب البليدة
Université SAAD DAHLAB de BLIDA

كلية التكنولوجيا
Faculté de Technologie

قسم الآلية والإلكترو تقني
Département d'Automatique et d'Électrotechnique

Mémoire de Master

Filière : Automatique

Spécialité : Automatique et Systèmes

présenté par

BOUAMRA Yacine

Etude et Simulation sous LabVIEW de la Commande d'un Ascenseur avec un API

Proposé par : **Dr. CHENTIR Amina**

Année Universitaire 2022-2023

Remerciements

Je remercie tout d'abord ALLAH, de m'avoir donné la force, la volonté, le courage et la patience d'aller jusqu'au bout de ce modeste travail.

La première personne que je tiens à remercier est ma promotrice Madame ***CHENTIR Amina*** *pour avoir accepté de m'encadrer avec sa compréhension, son sérieux et son suivi continu tout le long de la réalisation de ce mémoire et qui n'a pas cessé de me donner des conseils.*

Je remercie chaleureusement les membres du Jury d'avoir accepté d'évaluer ce modeste travail.

Aussi, je remercie tous mes enseignants et membres de l'administration du département d'Automatique et d'Electrotechnique.

Finalement, j'adresse mes remerciements à tous ceux qui ont contribué de près ou de loin à la réalisation de ce travail.

Dédicaces

Je dédie mon travail :

A mes très chers parents pour m'avoir encouragé et poussé à atteindre mon but, et de m'avoir soutenu tout au long de mes études.

A mes frères Amine, Redha, Sofiane, Salim et mes belles sœurs Ahlem, Marwa, pour leur appui et leurs encouragements.

A mes amis qui ont été toujours présents à mes côtés avec leurs encouragements, critiques et conseils.

A tous ceux qui ont de près ou de loin, contribué d'une manière ou d'une autre pour que ce mémoire soit possible.

ملخص :

الهدف من هذا العمل هو دراسة مبدأ تشغيل مصعد من ثلاثة طوابق ثم محاكاته باستخدام برنامج LabVIEW من خلال تطوير واجهة رسومية للتحكم فيه باستخدام وحدة تحكم منطقية قابلة للبرمجة (PLC) من نوع ZELIO (SR3B261FU).

ثم لعمل نموذج لمصعد من ثلاثة طوابق يتم التحكم فيه بواسطة ZELIO LOGIC PLC (SR3B261FU) من شنايدر باستخدام برنامج Zeliosoft2.

الكلمات الرئيسية: مصعد ، LabVIEW، API، ZELIO، Zeliosoft2

Résumé :

Le but de ce travail est d'étudier le principe de fonctionnement d'un ascenseur à trois étages puis de le simuler avec le logiciel LabVIEW en développant une interface graphique pour le commander à l'aide d'un automate programmable industriel (API) de type ZELIO (SR3B261FU).

Puis de réaliser une maquette d'un ascenseur à trois étages commandée par un l'automate ZELIO LOGIC (SR3B261FU) de SCHNEIDER en utilisant le logiciel Zeliosoft2.

Mots clés : Ascenseur, LabVIEW, API, ZELIO, Zeliosoft2

Abstract :

The aim of this project is to study the operating principle of a three-floor escalator, then to simulate it using LABVIEW software by developing a graphical interface to control it using a ZELIO (SR3B261FU) programmable logic controller (PLC).

Then to construct a prototype of a three-floor escalator controlled by a SCHNEIDER ZELIO LOGIC (SR3B261FU) PLC using Zeliosoft2 software.

Keywords : Escalator, LabVIEW, PLC, ZELIO, Zeliosoft2

Table des

Matières

Chapitre 2 Les Automates Programmables Industriels

Chapitre 3 Logiciels de Programmation

Chapitre 4 Simulations et Tests

Liste des

Figures

Liste des Figures

Liste des

Tableaux

Liste des Tableaux

Liste

Des Abréviations

Liste des abréviations

API : Automate Programmable Industriel.

ASCII : American Standard Code for Information Interchange.

CAN : Convertisseur Analogique Numérique.

CNA : Convertisseur Numérique Analogique.

E/S : Entrée/Sortie.

FBD : Fonction Block Diagram (Logigramme).

HVAC : Heating Ventilation And Air-Conditioning.

IL : Instruction List (Liste d'instructions).

LD : Ladder Diagram (Diagramme échelle).

LOG : Logigramme.

LabVIEW : Laboratory Virtual Instrument Engineering Workbench.

Mm/In : Mili métre/Inch.

NF : Normalement Fermé.

NO : Normalement Ouvert.

NI : National Instrument.

NTC : Negative Temperature coefficient.

PC : Partie Commande.

PID : Proportionnel Intégral Dérivé.

PLC : Programmable Logic Controllers.

PO : Partie Opérative.

PWR : Power.

RAM : Random Access Memory (Mémoire vive).

ROM : Read Only Memory (Mémoire morte).

RTU : Remote Terminal Unit (Unité Terminale Distante).

SFC : Sequential Fonction (Grafcet).

ST : Structured Text (Texte Structuré).

TCP/IP : Transmission Control Protocol/Internet Protocol.

UC : Unité Centrale.

ZELIOSOFT2 : Logiciel de Programmation de l'API ZELIO.

Introduction

Générale

L'automatisation industrielle consiste tout simplement à utiliser des technologiques et des mécanismes capables d'exécuter des tâches mécaniques répétitives, afin de réduire l'intervention humaine et d'augmenter l'efficacité et la compétitivité de l'entreprise. Elle est devenue essentielle dans tous les secteurs de fabrication, ce qui nécessite de connaître ses fondamentaux et de suivre son évolution.

Tous les secteurs industriels font désormais appel à l'automatisme, pour tous les domaines d'activité : agroalimentaire, pétrochimie, pharmaceutique, semiconducteur, agriculture, hardware, etc. Parfois, ces systèmes automatisés sont si rapides et précis qu'ils réalisent des actions qui seraient impossibles pour un être humain. Ainsi, l'automatisation est synonyme de productivité et de sécurité.

L'histoire de l'ascenseur remonte aux temps des dispositifs primitifs de transport. C'est un appareil à déplacement longitudinal, utilisé intensivement au quotidien, pour gagner plus facilement les hauteurs d'un immeuble.

Cet appareil démocratisé remplace petit à petit les escaliers interminables des bâtiments et ne cesse de s'améliorer et c'est à partir de 1924 que l'ascenseur automatisé (sans liftier) a fait son apparition.

Actuellement, on assiste à des évolutions technologiques continuelles, en termes de fonctionnalités, de standing, de design et surtout de sécurité, auxquelles les grandes nations du monde prennent part.

Dans ce même contexte, l'objectif de notre étude est de simuler et deréaliser la commande d'un ascenseur à l'aide du logiciel LabVIEW et d'un automate programmable industriel (API) de type Zelio Logic (SR3B261FU).

Ce mémoire est organisé autour de quatre chapitres :

- Le premier chapitre : porte quelques généralités sur les ascenseurs, où on y présente les différentes types et catégories d'ascenseurs, les avantages et les inconvénients de chaque catégorie.

- Le deuxième chapitre : aborde la description des automates programmables industriels d'une façon générale et plus particulièrement l'automate Zelio Logic de SCHNEIDER que nous allons utiliser.

- Le troisième chapitre : permet de décrire les outils de programmation qu'on va utiliser tout au long de ce projet, à savoir le LabVIEW, Zeliosoft2 de SCHNEIDER et le module de communication Modbus de SCHNEIDER.

- Le quatrième chapitre : est réparti en deux parties. La première partie aborde les différentes étapes et tests réalisés pour la simulation sur LabVIEW de la commande d'un ascenseur à 3 étages. Ensuite, on passe à la deuxième partie qui consiste à réaliser une maquette qui simule ce fonctionnement en pratique.

- Enfin, une conclusion générale est alors présentée avec quelques perspectives.

Chapitre 1

Généralités sur les Ascenseurs

1.1 Introduction

Les premiers modèles d'ascenseurs utilisaient un système à vapeur et étaient utilisés dans des mines, mais ils étaient peu fiables et donc ont été rapidement abandonnés. Pour cela, l'américain Elisha Graves Otis décida de développer ce système en le dotant d'une protection contre toute chute : le frein parachute a été créé en 1854. Il permettait à l'appareil de ralentir et de s'arrêter en cas de dysfonctionnement ou de chute brutale.

L'ascenseur devient enfin sûr, et peut donc être installé dans des lieux publics. Otis met alors en place son invention dans un grand magasin de New York : elle permettait de transporter jusqu'à 450 kg, mais sa vitesse était limitée à 0,2 m/s.

Le terme ascenseur, quant à lui, fut inventé par le Français Léon Edoux, en 1864, qui proposa un système à base de vérins utilisant la pression provoquée par l'eau de la ville. Il rencontra par la suite les deux fils d'Elisha Graves Otis, ensemble, ils conçurent l'ascenseur de la tour Eiffel en 1889. Ce dernier était alors réputé pour sa hauteur et sa vitesse : 0,8 m/s.

Enfin, en 1924, un dispositif sans machiniste a été créé : entièrement électrique, et disposant de sécurités accrues, il se démocratisa dans la majorité des bâtiments [1].

1.2 Définition d'un ascenseur

Un ascenseur est un système de transport vertical qui transporte des personnes ou des objets entre les différents étages d'un bâtiment. Il se compose généralement d'une cabine fermée qui se déplace le long de rails ou de câbles, reliée à un mécanisme de levage et de contrôle.

Les ascenseurs peuvent être alimentés par l'électricité ou d'autres sources d'énergie et sont généralement contrôlés par un opérateur ou un système automatisé. Ils sont largement utilisés dans les immeubles de bureaux, les centres commerciaux, les hôtels et les résidences pour faciliter les déplacements entre les différents étages. La Figure 1.1 présente un aperçu des différentes parties d'un ascenseur.

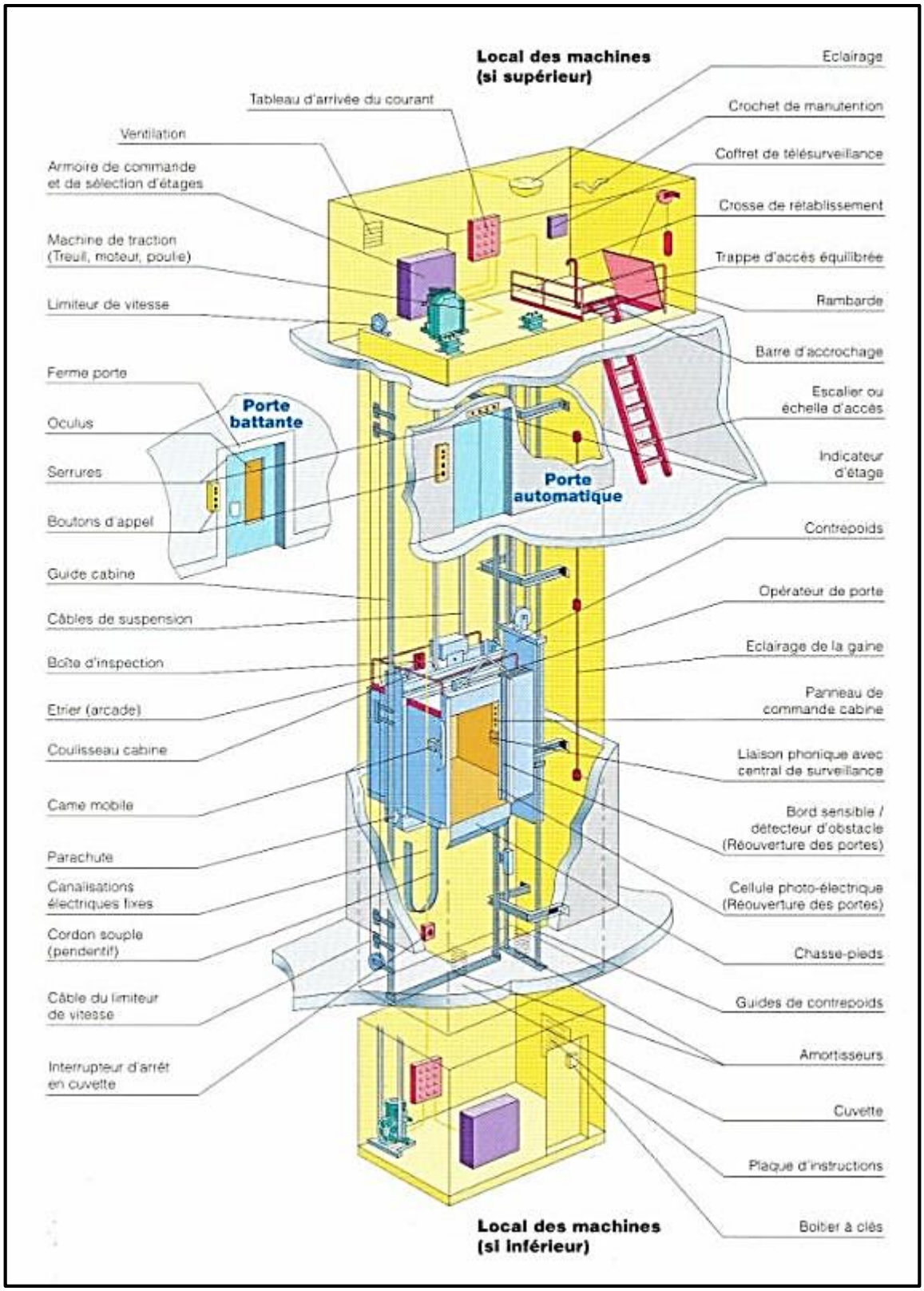

Figure 1.1 : Les différents parties de l'ascenseur [2]

1.3 Les différents types d'ascenseurs

En matière de transport vertical, il existe différents types d'ascenseurs qui répondent aux besoins spécifiques des utilisateurs.

Chaque type présente des caractéristiques particulières en termes de capacités, de vitesse et de technologie employée. Dans cette partie, nous allons passer en revue les principaux types d'ascenseurs et évoquer les situations dans lesquelles ils sont généralement employés, tout en présentant les différences entre eux.

1.3.1 Ascenseur pour personne

Ce type d'ascenseur est réservé aux personnes, il se déplace à l'aide d'une cabine à rails rigides verticaux. Il est simple d'utilisation et totalement sûr et se différencie des autres types d'ascenseurs par une belle cabine, un plus grand confort et une sécurité élevée [3].

1.3.2 Ascenseur de charge

Cet ascenseur est utilisé pour le transport des charges ou des équipements entre deux ou plusieurs niveaux définis en toute sécurité.

Robuste et simple d'utilisation, l'ascenseur de charge répond à la plupart des besoins pour le transport de marchandises : activités industrielles ou tertiaires, usines et entrepôts, hôtellerie, magasins, ateliers, etc… [3].

1.3.3 Ascenseur pour personnes à mobilité réduite

Également connu sous le nom d'ascenseur "handicapé", un ascenseur accessible aux personnes handicapées permet un accès facile aux personnes en fauteuil roulant.

Ce type d'ascenseur se caractérise par des dimensions spécifiques, notamment des ouvertures de porte d'au moins 0,8 m. Sa cabine mesure 1,4 mètre de haut et 1,10 mètre de large. De plus, il présente un panneau de commande bas, à une hauteur allant jusqu'à 1,3 m. Ce système peut également prendre la forme d'une cabine semi-ouverte, permettant d'accéder à l'étage souhaité si on appuie en permanence sur le bouton haut ou bas [3].

1.4 Les catégories d'ascenseurs

On distingue deux catégories d'ascenseur :

- Ascenseurs hydrauliques.
- Ascenseurs à traction à câble.

1.4.1 Les ascenseurs hydrauliques

Les élévateurs hydrauliques répondent aux besoins de transport d'objets lourds à faible hauteur et sont généralement utilisés pour des déplacements relativement courts de l'ordre de 15 à 18 m.

Ils sont utilisés dans plusieurs applications différentes, et les charges généralement soulevées par le système de levage sont élevées comme par exemple les automobiles.

Plusieurs modèles existent sur le marché, on citera les ascenseurs hydrauliques (Figure 1.2) [4] :

- À cylindre de surface.
- À cylindre enterré.
- Télescopiques à cylindre de surface.

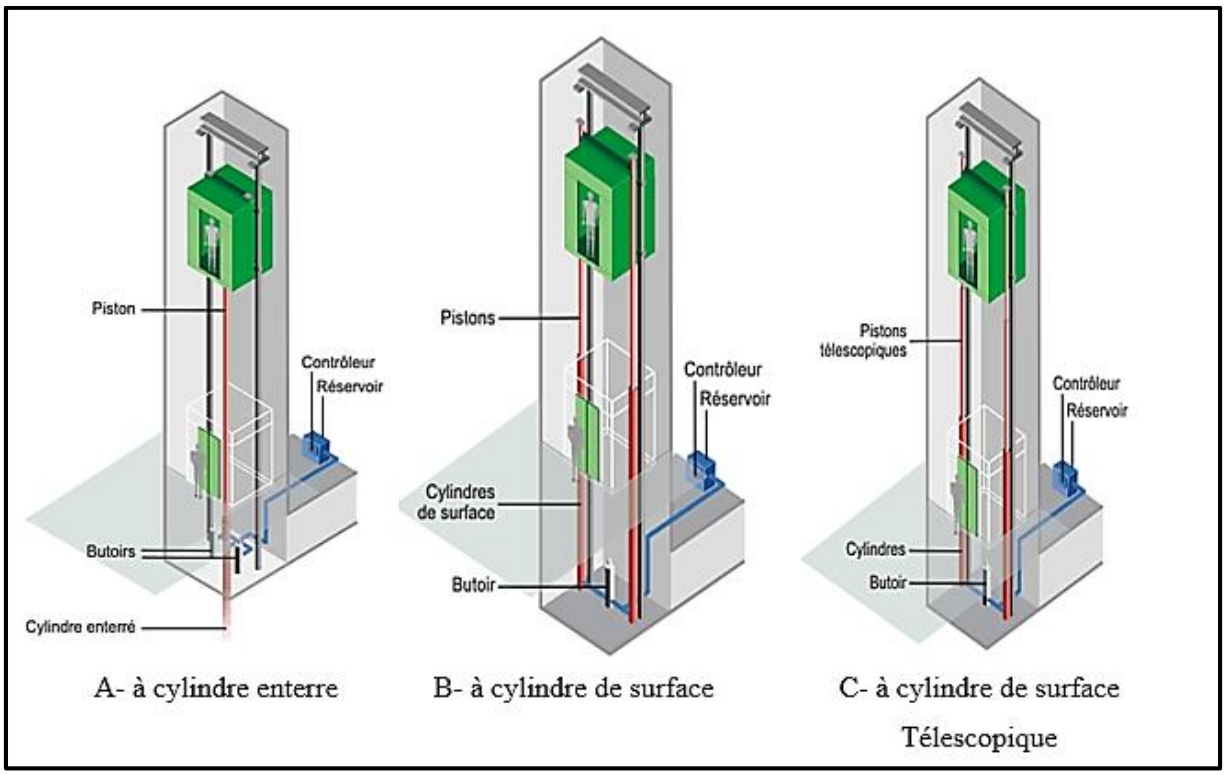

Figure 1.2 : Différents ascenseurs hydrauliques [4]

Les ascenseurs hydrauliques se composent principalement [4] :

- D'une cabine.
- De guides.
- D'un ensemble de pistons-cylindres hydrauliques placés sous la cabine de l'ascenseur.
- D'un réservoir d'huile.
- D'un moteur électrique accouplé à une pompe hydraulique.
- Et d'un contrôleur.

Les différents modèles rencontrés permettent de tenir en compte les critères suivants [4] :

- Place.
- Hauteur de l'immeuble à desservir.
- Stabilité de sol et du sous-sol.
- Risque de pollution par rapport au sol et plus spécifiquement aux nappes phréatiques.
- Esthétique.

a Les avantages

Parmi leurs avantages, on peut citer [4] :

- Réglage facile de la vitesse de déplacement.
- Implantation facile dans un immeuble existant.
- Précision au niveau de déplacement.
- Charge importante.
- Ne nécessite pas de local de machinerie.

b Les inconvénients

Parmi leurs inconvénients, on peut citer [4] :

- Risque de pollution des sous-sols.
- Consommation énergétique importante.
- Nécessite de renforcer la dalle de sol.
- Course verticale limitée à une hauteur entre 15 à 18 m.
- La sécurité incendie compliquée à cause de la quantité importante d'huile.
- Une vitesse réduite.

- L'absence de contrepoids provoque un surdimensionnement, des consommations et des appels de puissance importants (à charge et a vitesse égales, il faut de l'ordre de 3 fois plus de puissance).

1.4.2 Les ascenseurs à traction à câble

Les ascenseurs à traction à câbles sont les types d'ascenseurs que l'on rencontre le plus, notamment dans les bâtiments tertiaires.

Ils se différencient entre eux selon le type de motorisation (Figure 1.3) [5] :

- À moteur-treuil à vis sans fin.
- À moteur-treuil planétaire.
- À moteur à attaque directe (couramment appelé « Gearless » ou sans treuil).

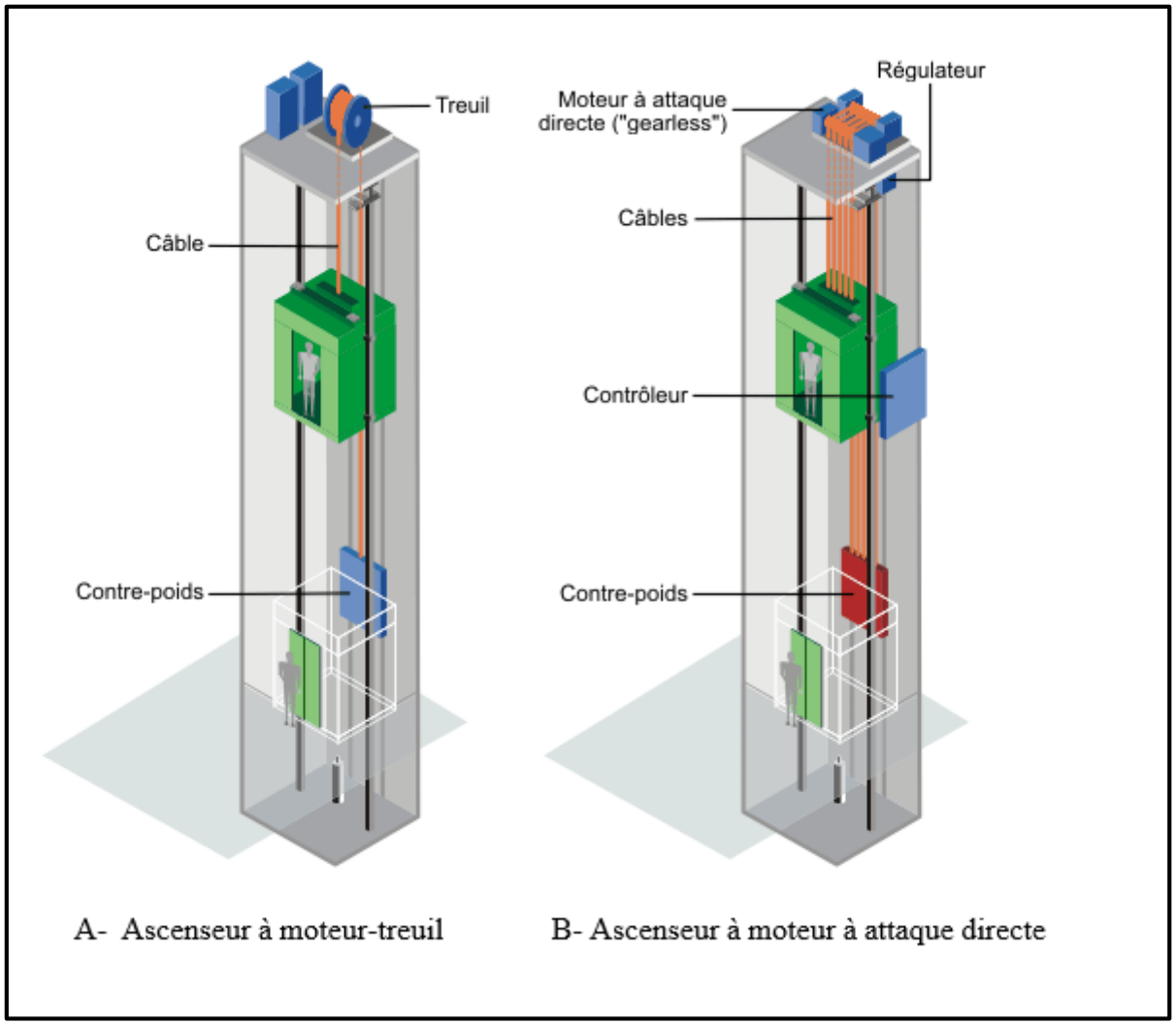

Figure 1.3 : Ascenseur à traction à câble [5]

a Les avantages

Parmi leurs avantages, on peut citer [5] :

- Course verticale pas vraiment limitée.
- Suivant le type de motorisation, précision au niveau de la vitesse et du déplacement.
- Rapidité de déplacement.
- Efficacité énergétique importante.
- Pas de souci de pollution.

b Les inconvénients

Parmi leurs inconvénients, on peut citer [5] :

- En version standard, nécessite un cabanon technique en toiture.
- Exigence très importante sur l'entretien.

1.5 Critère de choix de l'ascenseur

Lorsqu'il est demandé de réaliser ou d'installer un ascenseur, le critère de choix doit englober les informations suivantes [6] :

- Constructif : tel que le nombre d'étages du bâtiment, l'espace disponible sur les étages, la disponibilité de l'espace de la salle des machines au sommet de la gaine, la stabilité du sol, etc.
- Organisation : comme le type fonctionnel du bâtiment et son occupation, garantie de confort et de performance de circulation (rapport charge/vitesse).
- Sécurité : c'est le détail le plus important à prendre en compte.
- Énergétique : tout en considérant que la consommation électrique et les appels doivent être limités.

1.6 Système de motorisation

Les ascenseurs à traction étant plus courants que les ascenseurs hydrauliques, cette étude n'inclura pas ces derniers et se limitera à des comparaisons entre les différents types d'ascenseurs à treuil.

1.6.1 Moteur-treuil à vis sans fin à une ou deux vitesses

Un moteur de treuil avec un engrenage à vis sans fin (Figure 1.4) est moins utilisé qu'un moteur à entraînement direct (sans réducteur ou « sans engrenage »).

Dans ce type de motorisation, les vis sans fin provoquent de nombreuses pertes mécaniques, d'où une forte consommation électrique. Les moteurs électriques reliés aux treuils à vis sans fin sont généralement des moteurs à courant continu à excitation indépendante ou en parallèle, qui sont connus pour avoir la capacité de changer de vitesse très facilement.

Les moteurs à courant alternatif sont essentiellement des moteurs à deux vitesses où nous constatons [7] :

- Qu'au démarrage, la vitesse est plus lente.
- Et afin d'atteindre la vitesse optimale de déplacement, le moteur passe en deuxième vitesse en générant un léger choc d'accélération (passage de la petite vitesse à la grande vitesse).

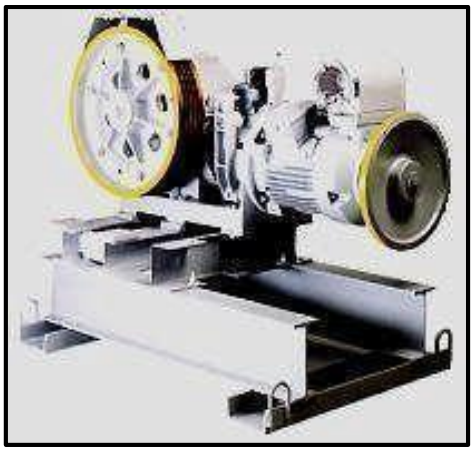

Figure 1.4 : Moteur-treuil à vis sans fin [7]

1.6.2 Les moteurs-treuils planétaires

Le moteur de treuil planétaire (Figure 1.5) adopte un système de réduction d'engrenage planétaire. Cela permet d'avoir un rapport de démultiplication confortable ainsi qu'une plage de vitesse adaptée au mouvement recherché.

Malgré les hautes performances des moteurs d'entraînement, le système garantit un rendement mécanique de 97 % à 98 %, garantissant ainsi un rendement énergétique total intéressant (environ 80 %) [7].

Les réducteurs planétaires peuvent être couplés à des moteurs électriques :

- Courant continu (large plage de vitesse).
- AC asynchrone à deux vitesses.
- Courant alternatif asynchrone contrôlé par un convertisseur de fréquence.

Figure 1.5 : Moteur-treuil planétaire [7]

1.6.3 Les moteurs à attaque direct (Gearless ou sans Treuil)

Avec l'avènement des variateurs de fréquence, sont apparus les moteurs à entraînement direct sans engrenage (Figure 1.6). L'installation est devenue si compacte qu'il est désormais possible de se passer d'une salle de machines sur le toit du bâtiment.

L'efficacité énergétique du système est élevée, principalement en raison de la présence de convertisseurs de fréquence qui optimisent la consommation d'énergie ; les pertes mécaniques sont réduites en raison de l'absence de réducteur, ce qui contribue également à une efficacité énergétique optimale.

Certains fabricants déclarent une efficacité énergétique d'environ 80 % [7].

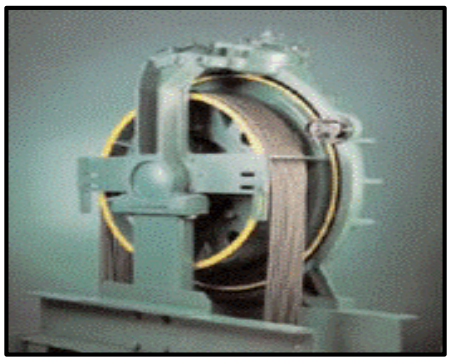

Figure 1.6 : Moteur-Gearless [7]

1.7 Critère de choix de la motorisation

Lors d'une installation, le critère de choix à suivre pour la motorisation est comme suit :

a **Efficacité globale :** Elle affecte la taille de l'installation et la consommation future. Lorsque les rendements globaux sont élevés, on réduit :
- Le surdimensionnement du moteur et de l'installation électrique.
- Les consommations énergétiques durant la vie de l'ascenseur.
- Les chutes de tension en ligne perturberont le réseau électrique interne voire externe.
  - Le rendement global d'une motorisation correspond à :

  ηglobal = η élec commande * ηélec moteur * η méca réducteur * η méca poulie

Une comparaison entre un moteur de traction avec et sans réducteur montre que pour une même puissance électrique absorbée, l'intermédiaire provoque des pertes et donc réduit la puissance mécanique disponible à la poulie de traction.

b **Performance énergétique** : Elle est déterminée par la gestion du démarrage et de l'arrêt de la gestion du variateur de vitesse. Les variateurs de vitesse garantissent réellement la performance énergétique en :
- Nivellement précis et démarrage en douceur pour le confort de l'utilisateur.
- Le contrôle permanent du couple et de la puissance en optimisant les courants de démarrage et les consommations.
- La possibilité de renvoyer de l'énergie sur le réseau électrique durant le freinage.

c **Le volume des équipements :** la limitation de l'espace nécessaire à la machinerie permet une réduction des couts d'investissement et un gain de place [6].

1.8 Conclusion

Nous avons présenté dans ce chapitre des informations générales sur les ascenseurs, les différents types et catégories, ainsi qu'une description de chaque catégorie et les systèmes de motorisations.

D'après les avantages et les inconvénients indiqués, on constate également que les ascenseurs à traction à câble sont plus efficaces que les ascenseurs hydrauliques.

Chapitre 2

Les Automates Programmables Industriels

2.1 Introduction

L'objectif de l'automatisation des systèmes est de produire, en ayant recours le moins possible à l'homme, des produits de qualité et ce pour un coût le plus faible possible.

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

2.2 Définition d'un API

L'Automate Programmable Industriel ou API (en anglais Programmable Logic Controller, PLC) est un dispositif similaire à un ordinateur ayant des entrées et des sorties et qui est utilisé pour automatiser des processus comme la commande des machines sur une ligne de montage dans une usine [11].

Autrement dit, c'est un appareil électronique numérique programmable conçu pour contrôler les processus industriels par traitement séquentiel. Il envoie des commandes au pré-actionneur (partie opérative ou PO côté actionneur) à partir de données d'entrée (capteurs) (partie commande ou PC côté capteur), d'instructions et de programmes informatiques [11].

2.3 Architecture des API

Les API comportent deux structures : externe et interne.

2.3.1 Structure externe de l'automate

Les automates peuvent être compacts ou modulaires [6] :

a Compact :

Ces automates contiennent une alimentation, un processeur, une interface E/S.

Selon les fabricants, il est même possible d'effectuer certaines fonctions supplémentaires comme le comptage rapide (Figure 2.1).

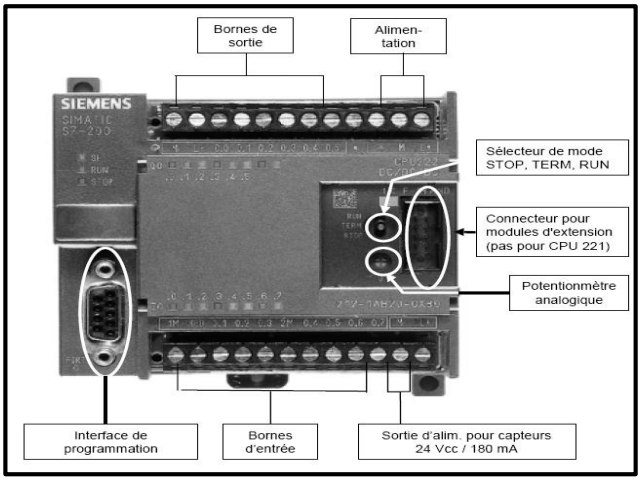

Figure 2.1 : Automate compact [12]

b Modulaire (Modicon) :

Ce type d'automate met l'alimentation, le processeur et les E/S dans des unités séparées. Ils sont fixés sur un ou plusieurs racks (Figure 2.2).

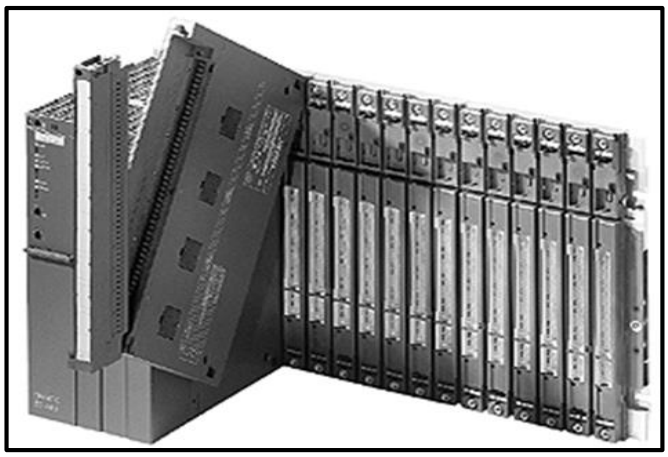

Figure 2.2 : Automate modulaire [12]

2.3.2 Structure interne de l'automate

Les API comportent trois parties principales (Figure 2.3) [12] :

a Une alimentation

Elle sert à fournir l'énergie nécessaire au bon fonctionnement des automates. La tension principale utilisée dans l'automate est de +12v et ±5v pour permettre le fonctionnement des cartes électroniques internes.

b Unité centrale

La structure interne d'un automate programmable industriel (API) est assez voisine de celle d'un système informatique simple.

L'unité centrale est le regroupement du processeur et de la mémoire centrale. Elle commande l'interprétation et l'exécution des instructions programme. Ces dernières, sont effectuées les unes après les autres, séquencées par une horloge.

Il y a deux types de mémoire qui cohabitent :

- La mémoire Programme où est stocké le langage de programmation (ROM : mémoire morte) est en général figée, c'est à dire en lecture seulement.
- La mémoire de données utilisable en lecture-écriture pendant le fonctionnement est la RAM (mémoire vive) qui fait partie du système entrées /sorties. Elle fige les valeurs (0 ou 1) présentes sur les lignes d'entrées, à chaque prise en compte cyclique de celle-ci, et mémorise les valeurs calculées à placer sur les sorties.

c Les interfaces d'Entrées/Sorties

- Interface d'entrées : L'interface d'entrée comporte des adresses d'entrée. Chaque capteur est relié à une de ces adresses.
- Interface de sorties : L'interface de sortie comporte de la même façon des adresses de sortie. Chaque pré actionneur est relié à une de ces adresses. Le nombre de ces entrées est sorties varie suivant le type d'automate.

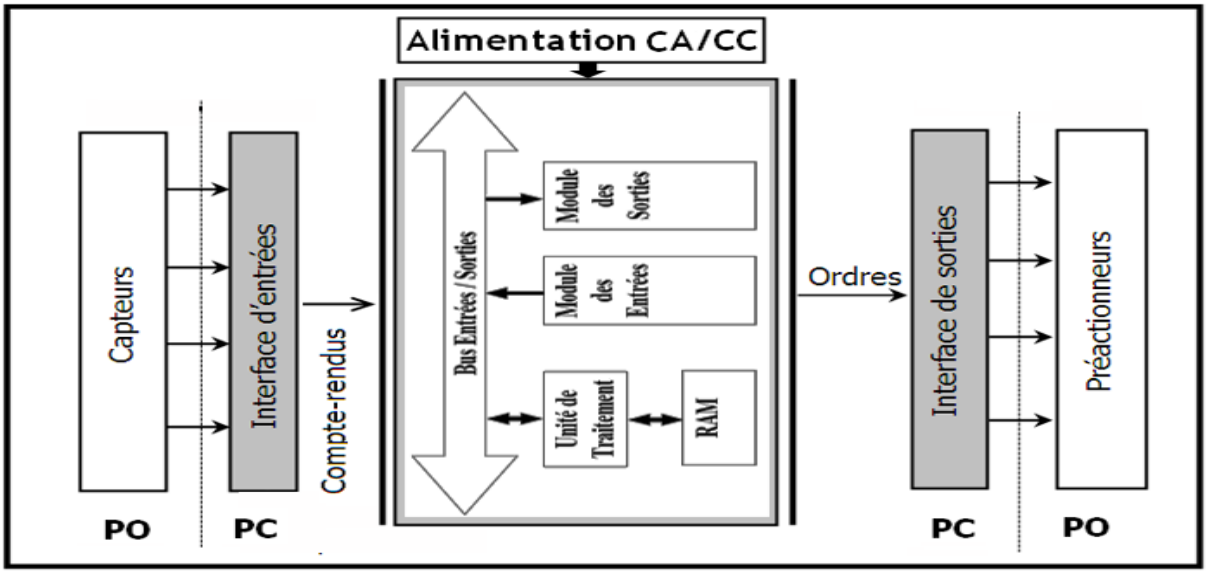

Figure 2.3 : Structure interne d'un automate programmable industriel [12]

2.4 Principe de fonctionnement d'un API

L'automate programmable fonctionne par déroulement cyclique du programme. Le cycle comporte trois opérations successives qui se répètent normalement comme suit (Figure 2.4) [14] :

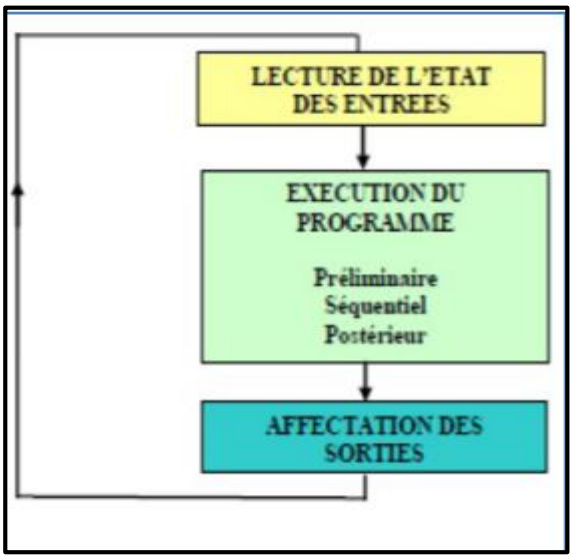

Figure 2.4 : Principe de fonctionnement d'un API [13]

a Phase 1 : Lecture

Durant cette phase, qui dure quelques microsecondes :

- L'entrée est photographiée et son état logique est stocké dans une région Spécifique au stockage de données.
- Le programme n'est pas scruté.
- La sortie n'est pas mise à jour.

b Phase 2 : Traitement

Durant cette phase qui dure quelques millisecondes :

- Les instructions de programme sont exécutées une à une. Si l'état d'une entrée doit être lu par le programme, c'est la valeur stockée dans la mémoire de données qui est utilisée.
- Le programme détermine l'état des sorties et stocke ces valeurs dans une zone de la mémoire de données réservée aux sorties.
- Les entrées ne sont pas scrutées.
- Les sorties ne sont pas mises à jour.

Notons que pendant cette phase, seules la mémoire de données et la mémoire programme sont mises à contribution. Si une entrée change d'état sur le module d'entrées, l'API ne voit pas ce changement.

c Phase 3 : Ecriture

Durant cette phase qui dure quelques microsecondes :

- Les états des sorties mémorisés précédemment dans la mémoire de données sont reportés sur le module de sorties.
- Les entrées ne sont pas scrutées.
- Le programme n'est pas exécuté.

2.5 Les avantages et les inconvénients

Les API présentent de nombreux avantages et inconvénients qui peuvent se résumer à [14] :

- Les avantages :

- Amélioration des conditions de travail en éliminant les travaux répétitifs.
- Amélioration de la productivité en augmentant la production.
- Amélioration de la qualité des produits ou réduction des coûts de production.
- Programmation facile avec un langage de programmation facile à comprendre (logique programmée) alors que la modification du programme est facile par rapport à la logique câblée.
- Simplification du câblage.
- Puissance et rapidité.
- Facilité de maintenance (l'API par lui-même est relativement fiable et peut aider l'homme dans sa recherche de défauts).
- Augmentation de la sécurité.
- Possibilités de communication avec l'extérieur (ordinateur, autres API).
- Énorme possibilité d'exploitation.
- Plus économiques.

- Les inconvénients :

- Plantage ou blocage du programme (très rare heureusement).
- Il y a trop de travail requis dans les fils de connexion.
- Besoin de formation.
- Son prix cher qui ne le met pas à la portée de toutes les bourses.

2.6 Langages de programmation

Le processeur d'un API peut exécuter un certain nombre d'opérations logiques ; l'ensemble des instructions booléennes des instructions complémentaires de gestion de programme (saut, mémorisation, adressage ...) constitue un jeu d'instructions.

Chaque automate possède son propre jeux d'instructions. Mais par contre, les constructeurs proposent tous une interface logicielle de programmation répondant à la norme CEI1131-3. Cette norme définit cinq langages de programmation utilisables, qui sont [12] :

- Les langages graphiques :
  LD : Ladder Diagram (Diagrammes échelle).
  FBD : Function Block Diagram (Logigrammes).
  SFC : Sequential Function Chart (Grafcet).
- Les langages textuels :
  IL : Instruction List (Liste d'instructions).
  ST : Structured Text (Texte structuré).

2.6.1 Les langages textuels

a Liste d'instructions (IL)

Le langage liste d'instructions est un langage textuel de bas niveau, il est particulièrement adapté aux applications de petite taille, il permet de transcrire sous forme de liste :

- Un schéma à contact.
- Un logigramme, équation booléennes.
- Un grafcet.

Il permet aussi de résoudre quelques calculs numériques, et la réalisation de fonction d'automatisme telles que la temporisation, le comptage…etc.

Chaque instruction est composée d'un code instruction et d'un opérande [9].

b Texte Structuré (ST)

Le texte structuré est un langage textuel de haut niveau qui est utilisé pour décrire des procédures complexes. Il est très proche du langage Pascal. Il utilise des expressions, un assemblage ordonné d'opérateurs, avec des priorités.

Ce langage facilite donc la mise en œuvre d'algorithmes complexes comportant beaucoup de traitement numérique. En contrepartie, il est moins commode pour la mise au point de fonctions booléennes [15].

2.6.2 Les langages graphiques

a LADDER (LD)

Le langage à contacts se base sur une approche visuelle évoquant des schémas électriques (avec symbole US). Dans le cas de traitement booléen, les éléments fondamentaux sont des contacts normalement fermés (NF) ou normalement ouverts (NO), et des bobines (Figure 2.5).

Ce langage est très efficace pour des systèmes combinatoires. Mais le Ladder n'est pas adapté aux programmes complexes impliquent un grand nombre de séquences, ni adapté aux calculs complexes [15].

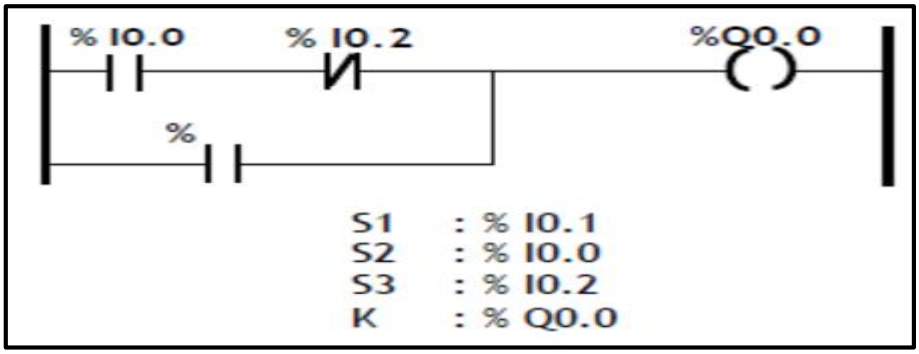

Figure 2.5 : Exemple de langage LADDER [13]

b Logigramme (FBD)

Un réseau LOG est composé d'une ou plusieurs boîtes d'opérations LOG. Au lieu d'utiliser des contacts, on affecte une ou plusieurs valeurs binaires comme entrées à une boîte d'opération LOG.

On utilise les sorties de l'opération pour connecter cette dernière à une opération consécutive ou pour achever le réseau. Ainsi, une seule opération LOG peut représenter la même fonction qu'un ensemble de contacts, bobines ou boîtes en schéma à contacts.

Le réseau est achevé lorsque on a procédé à l'affectation de tous les paramètres de l'opération ou qu'on les a connectés à une autre opération (Figure 2.6) [9].

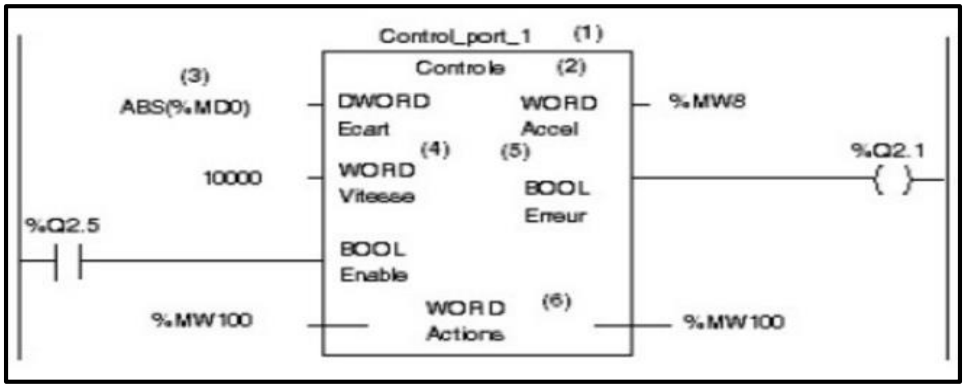

Figure 2.6 : Exemple de langage Logigramme [13]

c Les graphes de fonction séquentielle (SFC)

Ils sont issus du langage GRAFCET. Ce langage, de haut niveau, permet la programmation aisée de tous les procédés séquentiels, il est semblable au diagramme états-transitions.

Ils sont particulièrement adaptés à la commande de cycles opératoires. Graphiquement, on se trouve devant une alternance étape(s) — transition-étape(s), etc., avec des liaisons dirigées. À chaque transition est associée une condition de franchissement de cette transition ; à chaque étape peuvent être définies des actions à entreprendre (Figure 2.7) [15].

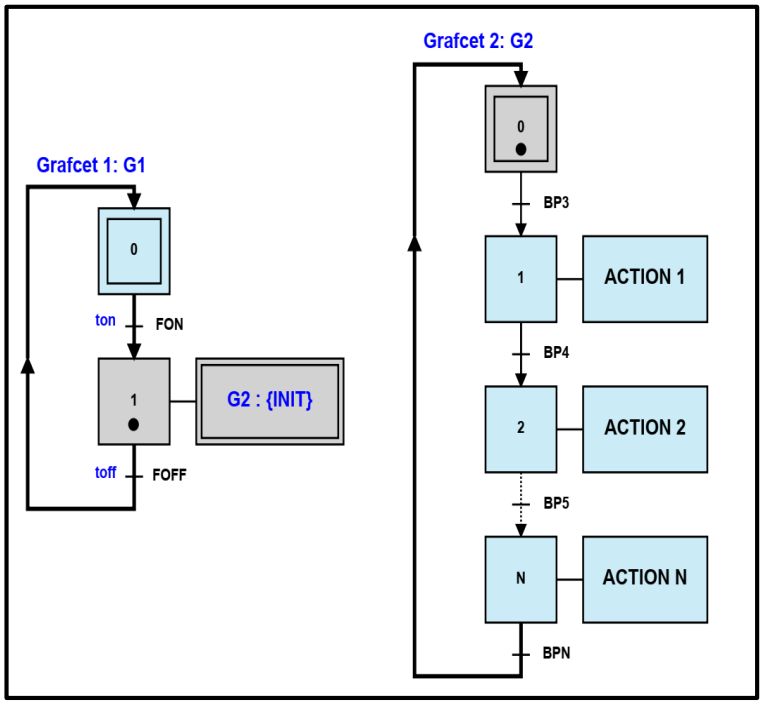

Figure 2.7 : Exemple d'un GRAFCET [16]

2.7 Critère de choix d'un automate

Le choix d'un automate programmable est basé sur plusieurs points, nous citons [13] :

- 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, Modbus ...).

2.8 L'automate Zelio Logic

Dans notre réalisation, nous allons faire appel à un automate Zelio Logic. Pour cela, une présentation générale de ce type d'automate est plus que nécessaire.

2.8.1 Définition

L'automate Zelio Logic (Figure 2.8) de Schneider Electric est un petit automate programmable spécialement conçu pour les applications en basse tension. Il offre une grande flexibilité et des performances élevées pour les applications de contrôle industriel simples. Il peut être utilisé pour contrôler des machines, des systèmes de production, des éclairages, des pompes, des moteurs et bien plus encore.

Il est doté d'une interface de programmation intuitive et facile à utiliser, qui permet aux utilisateurs de créer rapidement des programmes de contrôle personnalisés. Il est également équipé d'un grand nombre d'entrées et de sorties afin de gérer des systèmes plus complexes. Avec sa taille compacte et ses performances élevées, l'automate Zelio Logic est une solution idéale pour les petites et moyennes applications industrielles.

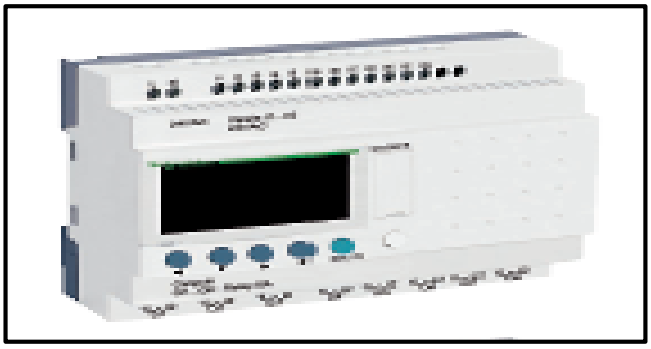

Figure 2.8 : Exemple d'un API Zelio Logic

2.8.2 Les avantages de l'API Zelio

L'API Zelio Logic offre de nombreux avantages [17] :

- Haute performance :
- Deux fois plus de mémoire de programmation et plus de blocs fonctionnels par simple mise à jour du micro logiciel.
- Plus de fonctionnalité :
- Fonction PID pour les applications HVAC et les modems 2G/3G.
- Entrées module 24V/DC compatibles avec les sondes de température NTC (programmables en langage FBD).
- Plus d'efficacité, moins de temps d'ingénierie :
- Logiciels et micro logiciels gratuits téléchargeables sur le site Web de Schneider Electric.
- Prise en main du logiciel en moins d'une heure, programmation simplifiée sans outil en langages à contacts, FBD et SFC pour les petites applications.
- Accès au programme et modification des paramètres sur afficheur intégré.
- Plus de flexibilité - Conception, maintenance et mise en service simplifiées :
- Gamme de relais intelligents compacts et modulaires et de modules d'extension.
- Logique programmable : une alternative intelligente à la logique câblée ou aux cartes dédiées.

2.8.3 Domaines d'application

Les relais intelligents Zelio Logic sont conçus pour être utilisés dans de petits systèmes automatisés. Ils sont utilisés dans les secteurs industriel et commercial [17] :

a Pour l'industrie :

- Automatisation de petites machines de finition, de production, d'assemblage ou de conditionnement.
- Petits automatismes fonctionnant sous 48 V (application levage...).
- Automatisation décentralisée des équipements annexes pour les grandes et moyennes machines (dans le textile, la plasturgie, etc.).
- Systèmes d'automatisation pour machines agricoles (irrigation, pompage, serres, etc.).

b Pour les secteurs tertiaire/bâtiment :

- Automatisation de barrières, volets roulants, contrôle d'accès.
- Automatisation des systèmes d'éclairage.
- Automatisation des compresseurs et des systèmes de climatisation.
- etc.

2.9 Conclusion

On a vu dans ce chapitre quelques généralités sur les automates, leur architecture, langage de programmation, puis on a mis en évidence leurs avantages et inconvénients et on a cité quelques critères de choix d'un API.

Ensuite, on a introduit brièvement l'automate Zelio Logic et on a exposé ses avantages et ses domaines d'applications.

Chapitre 3

Logiciels de

Programmation

3.1 Introduction

Les logiciels de programmation sont des outils utilisés pour écrire, développer, tester et déboguer des programmes relatifs aux systèmes automatisés à l'aide de différents langages de programmation.

Dans ce chapitre nous allons présenter les outils et les logiciels qu'on va utiliser dans notre projet.

3.2 Description du logiciel Zelio soft 2

Le logiciel de programmation Zelio Soft 2 est conçu pour programmer les modules logiques de la gamme Zelio Logic. Il permet de choisir entre les langages de programmation, d'afficher les données du programme et des paramètres, de charger et télécharger des applications, ainsi que d'imprimer la documentation de l'application.

Autrement dit, il permet [18] :

- La programmation en langage à contacts (LADDER) ou en langage à blocs fonctions (FBD), la simulation, le monitoring et la supervision, le chargement et le déchargement de programmes, l'édition de dossiers personnalisés, la compilation automatique de programmes et l'aide en ligne.
- De surveiller les applications grâce à son test de cohérence. À la moindre erreur de saisie, un indicateur passe au rouge. Il suffit d'un clic sur la souris pour localiser le problème.
- À tout moment de passer dans l'une des 6 langues (anglais, français, allemand, espagnol, italien, portugais) et d'éditer le dossier application dans cette langue.
- De configurer des blocs fonctions texte, affichables sur tous les modules LOGIC avec afficheur.

3.2.1 Langages Ladder et FBD

Le logiciel Zelio soft 2 permet deux langages de programmation (Figure 3.1) :

- En langage à contacts (Ladder).
- En diagramme à blocs fonctions (FBD).

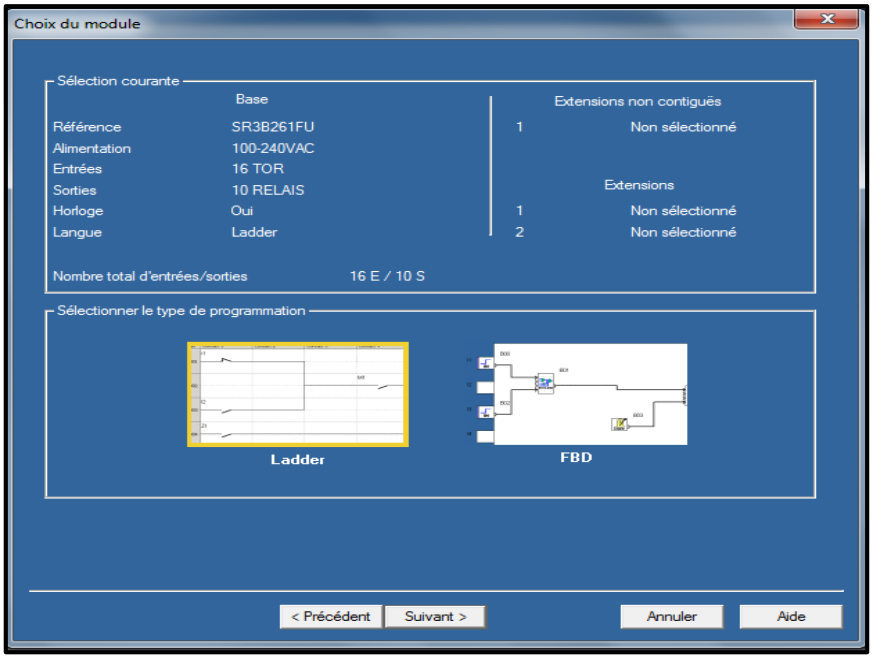

Figure 3.1 : Les modes de programmation pour zeliosoft2

a Langage FBD (Fonctionnel Block Diagramme)

Le langage FBD permet une programmation graphique basée sur l'utilisation de blocs fonctionnels prédéfinis (Figure 3.2).

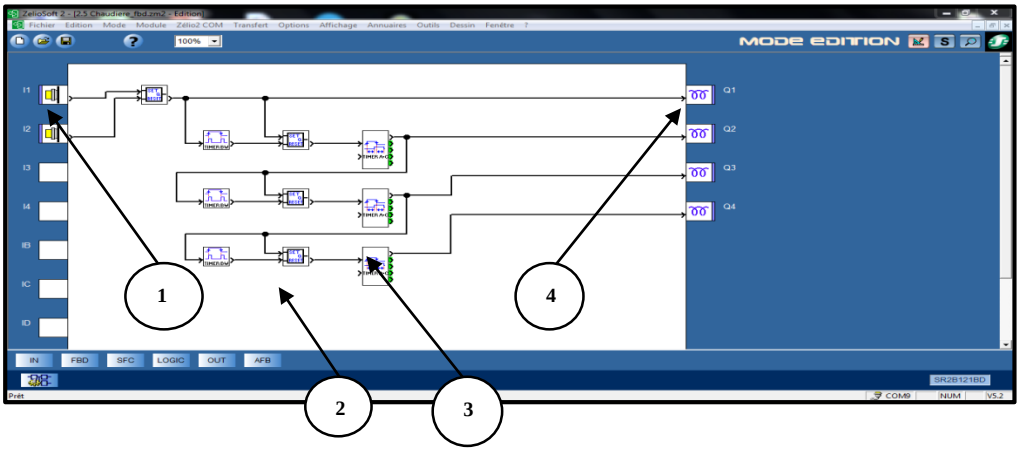

Figure 3.2 : Langage FBD

Le tableau 3.1 présente quelques éléments de la fenêtre d'édition.

REPERE	DESCRIPTION
1	Zone des blocs fonctions d'entrées
2	Zone de câblage
3	Block fonction
4	Zone des blocs fonctions des sorties

Tableau 3-1 : Éléments de la fenêtre d'édition du langage FBD

b Langage LD (Ladder Diagramme)

Le langage LADDER est un langage graphique. Il permet la transcription de schémas à contacts, adaptés au traitement combiné. La Figure 3.3 présente de différents éléments de ce langage.

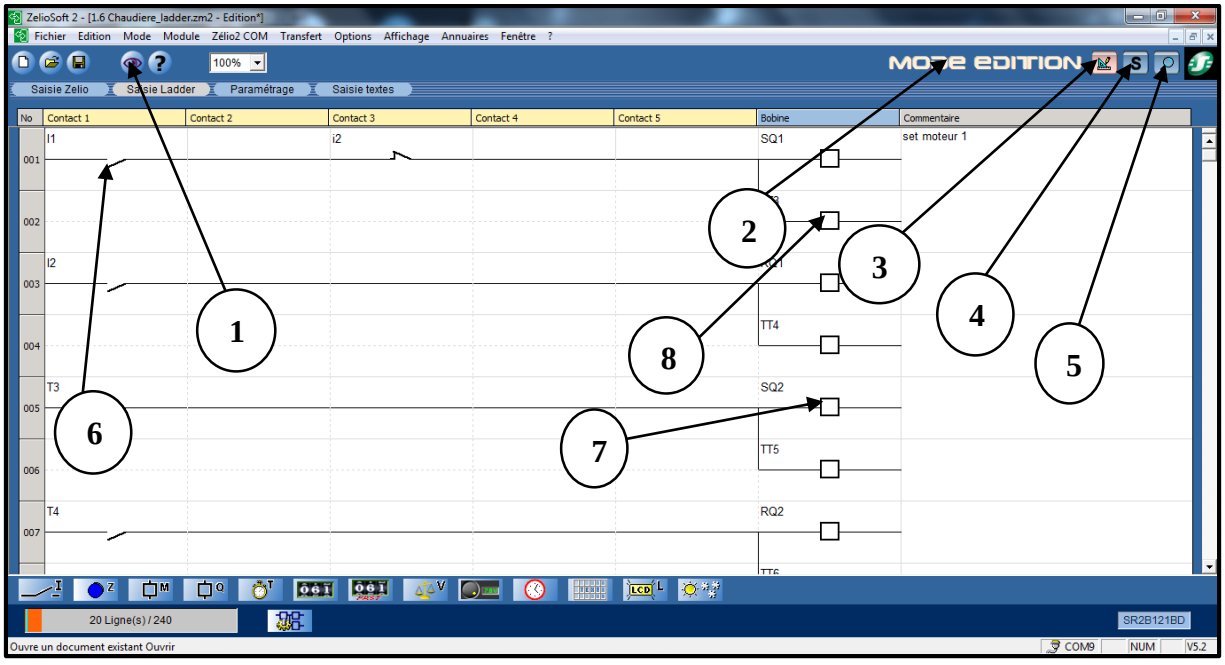

Figure 3.3 : Description des éléments de LADDER

Le tableau ci-dessous présente les différents éléments rencontrés dans un réseau de contacts.

REPERE	DESCRIPTION
1	Pour vérifier la cohérence du programme.
2	Ecran de programmation en Mode Edition.
3	Mode édition pour écrire le programme.
4	Mode simulation pour tester le programme.
5	Mode monitoring pour connecter l'automate.
6	Une entrée de l'automate.
7	Une sortie de l'automate.
8	Temporisateur ou tempo.

Tableau 3-2 : Les différents éléments d'un réseau de contacts

3.2.2 Différence entre les modes LADDER et FBD

Certains menus sont précis aux modes LD et FBD. Le tableau 3-3 explique la différence entre les deux modes [19] :

MENU	LD	FBD
PROGRAMMATION	✓	
MONITORING	✓	✓
PARAMETRES	✓	✓
RUN/STOP	✓	✓
CONFIGURATION	✓	✓
EFFACER PROG	✓	
TRANSFERT	✓	✓
VERSION	✓	✓
LANGUE	✓	✓
DEFAUT	✓	✓

Tableau 3-3 : La différence entre les deux modes LD et FBD

3.3 Description du logiciel LabVIEW

LabVIEW est un environnement de développement complet, graphique, compilé et particulièrement bien adapté au domaine de l'acquisition et de mesure. Son approche totalement graphique offre une souplesse et une dimension intuitive inégalée. Comparativement aux langages textuels, il offre la même puissance de programmation mais sans le coté abstrait et complexe lié à la syntaxe.

Ce langage s'adresse à tous les techniciens ou ingénieurs qui désirent développer une application de mesures, de tests, de supervision ou de contrôle /commande avec une interface utilisateur de très grande interactivité. Mais plus généralement à tous les développeurs qui désirent utiliser un langage, puissant et ouvert sur la totalité du génie logiciel [21].

3.3.1 Concepts de base de la programmation graphique

La programmation à l'aide d'un langage graphique date des années 1980. Les langages de programmation textuels conventionnels impliquent un mode de conception séquentiel contraint par l'architecture même de la machine. La représentation textuelle est moins performante puisqu'elle ne permet qu'une représentation séquentielle de l'information.

D'ailleurs, les efforts mis en œuvre pour rendre un texte plus rapidement compréhensible reposent sur des moyens graphiques. Lorsque des objets peuvent être manipulés graphiquement, la communication homme/machine est grandement facilitée [21].

3.3.2 Environnement de la programmation LabVIEW

Un programme ou VI, développé dans l'environnement LabVIEW, se compose principalement de deux éléments étroitement associés et regroupés sous le même nom « nom_application.vi » (l'extension .vi permet une reconnaissance immédiate par l'environnement LabVIEW).

Ainsi nous avons :

a La face avant (Front panel)

Qui représente le panneau de contrôle de l'instrument virtuel composé d'objets variés (boutons, indicateurs, graphes, etc.).

Cette fenêtre est l'interface utilisateur du programme au sens génie logiciel : définition des entrées/sorties de données accessibles par l'utilisateur du programme.

En imitant l'interface classique des appareils de mesures (générateurs, oscilloscope,), LabVIEW apporte une continuité pour les utilisateurs des appareils d'instrumentation. La

réalisation de cette interface peut être également effectuée de manière indépendante par rapport au programme (Figure3.4).

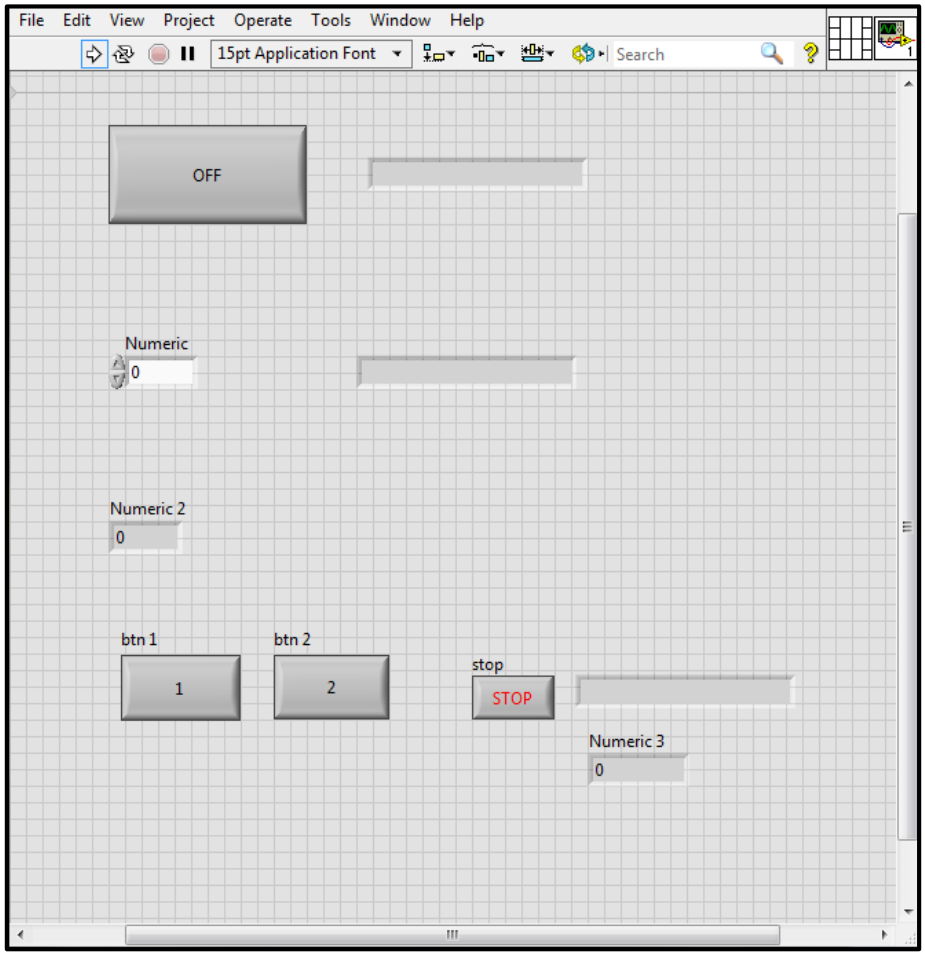

Figure 3.4 : Exemple de la face avant dans LabVIEW

b Le diagramme (Block diagram)

C'est le programme de l'application ou code source. Il est écrit sous la forme d'un diagramme flux de données en langage G : ensemble des icônes et des liaisons entre ces icones utilisées. Le diagramme contient les fonctions de l'instrument virtuel.

Contrairement à la procédure qui consistait pour le technicien ou le scientifique à dessiner le schéma d'une application et ensuite à le convertir en un code propre au langage choisi, pour LabVIEW, le diagramme est le programme. Représenté en image, le programme s'explique de lui-même et est donc facile à adapter et à comprendre lorsque la représentation graphique garde une taille acceptable. Cette partie de l'application est ce que l'on appelle le code source par opposition à l'interface utilisateur (Figure 3.5) [21].

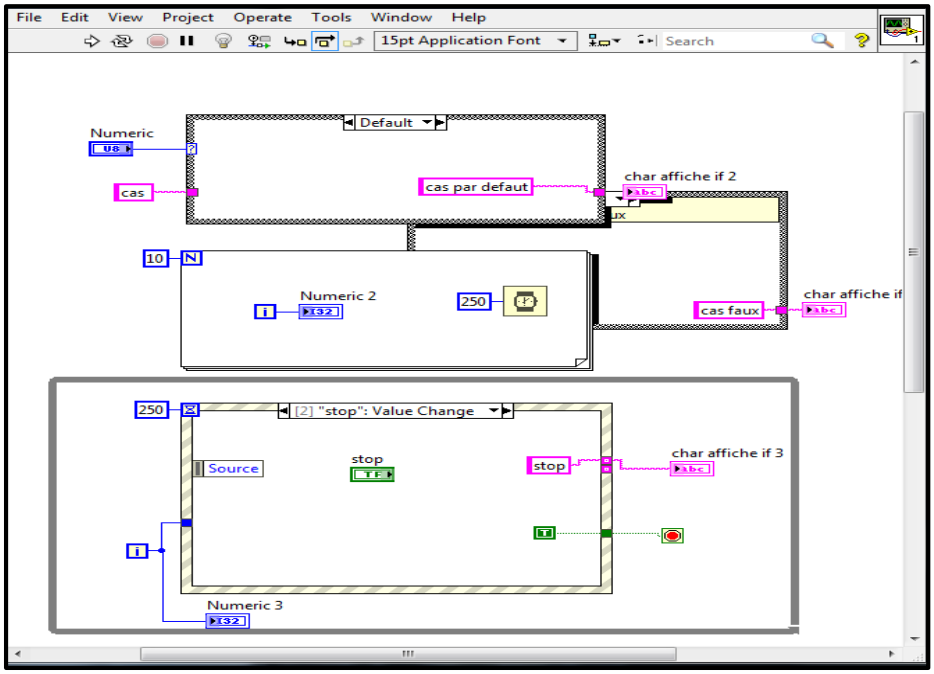

Figure 3.5 : Exemple d'un Bloc Diagramme

3.4 Protocole Modbus

Le protocole Modbus est un protocole de communication industriel développé dans les années 1970 par Modicon, maintenant connu sous le nom de Schneider Electric.

Il est utilisé pour établir une communication entre plusieurs équipements électroniques de terrain tels que des capteurs, des actionneurs et des automates programmables.

Le protocole Modbus est basé sur une structure de messages qui permettent à un maître (par exemple un automate) d'interroger et de contrôler des esclaves (par exemple des capteurs et des actionneurs). Les messages sont composés d'une adresse esclave, d'un code de fonction et d'une ou plusieurs données.

Il existe deux modes de communication dans le protocole Modbus : le mode ASCII et le mode RTU :

- Le mode ASCII représente les données en caractères ASCII, tandis que le mode RTU représente les données en bits.
- Le mode RTU est plus rapide et plus efficace que le mode ASCII, mais il est plus difficile à lire et à interpréter [22].

3.4.1 Principe de fonctionnement

Le protocole de communication Modbus est du type maître/esclave. Deux mécanismes d'échange sont possibles :

- Requête / réponse :

  - La requête du maître est adressée à un esclave donné.

  - La réponse est attendue en retour de la part de l'esclave interrogé.

- Diffusion :

- Le maître diffuse une requête à toutes les stations esclaves du bus. Ces dernières exécutent l'ordre sans émettre de réponse.

Les modules Zelio Logic modulaires se connectent au réseau Modbus via l'extension de communication réseau Modbus esclave. Cette extension est un esclave non isolé électriquement L'extension de communication réseau Modbus esclave doit être connecté à un module logique modulaire SR3B***BD, (alimenté en 24 Vcc) [17].

3.4.2 Types de données Modbus

On y trouve [22] :

- Les bobines qui sont simplement des bits. Ces bits peuvent être activés (1) ou désactivés (0). Certaines bobines représentent des entrées, ce qui signifie qu'elles contiennent le statut de certaines entrées physiques discrètes (tout ou rien) ou bien elles représentent les sorties, ce qui signifie qu'elles conservent l'état d'un signal de sortie physique discret.
- Les registres qui sont simplement des données de registre non signées de 16 bits. Les registres peuvent avoir une valeur de 0 à 65535 (hexadécimal de 0 à FFFF). Ils sont regroupés en registres d'entrée et en registres de stockage.
- L'intention initiale d'un registre d'entrée est de refléter la valeur d'une entrée analogique. C'est une représentation numérique d'un signal analogique comme une tension ou un courant.

3.4.3 Modules d'extension de communication

Les modules d'extension de communication réseau Liaison série Modbus et Ethernet Modbus/TCP permettent une connexion aux équipements d'automatismes tels que des afficheurs ou des automates programmables.

a Extension de communication réseau Liaison série Modbus

Les modules Zelio Logic modulaires se connectent au réseau Modbus via l'extension de communication réseau Modbus esclave. Cette extension est un esclave non isolé électriquement.

Cette extension de communication réseau Modbus esclave doit être connectée à un module logique modulaire SR3B***BD, alimenté en 24 Vcc (Figure 3.6).

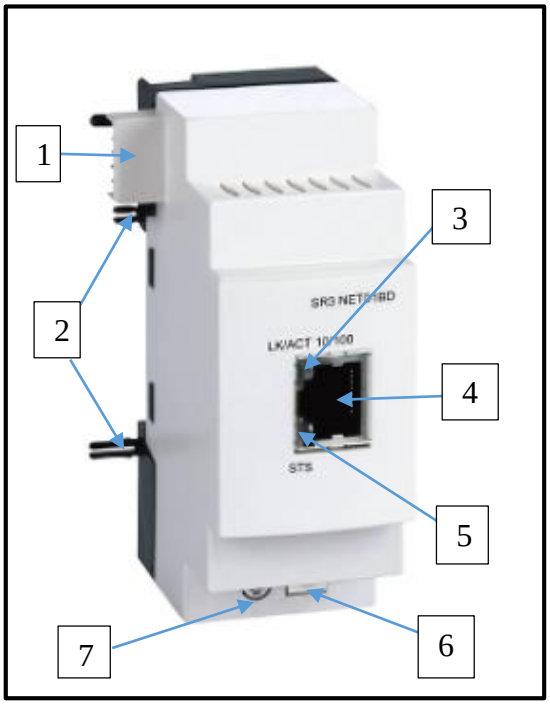

Figure 3.6 : Exemple d'un module de communication esclave réseau Modbus 24Vcc de Zelio Logic [23]

Le tableau 3-4 représente les différents éléments de l'extension de communication réseau Modbus esclave SR3MBU01BD :

REPERE	DESCRIPTION
1	Un connecteur pour raccordement au module Zelio Logic (alimentation fournie par le module Zelio Logic).
2	Des pions de détrompage.
3	Une DEL de visualisation pour la communication (COM).
4	Une connexion réseau Modbus (connecteur blindé RJ45 femelle).
5	Une DEL de visualisation de l'alimentation (PWR).
6	Un bornier à vis pour la connexion à la terre de protection.
7	Un ressort de clipsage pour le montage sur profiler de 35 mm/1,38 in.

Tableau 3-4 : Éléments de L'extension de communication SR3MBU01BD [17]

b Extension de communication réseau Ethernet (serveur)

L'extension SR3NET01BD permet de communiquer sur le réseau Ethernet selon le protocole Modbus/TCP en mode serveur, elle doit être connectée à un module logique modulaire SR3B***BD, alimenté en 24 Vcc [17].

3.4.4 Adressage des échanges Modbus

Dans l'adressage des échanges Modbus l'extension de communication réseau Ethernet/TCP accepte la conversion du logiciel zeliosoft2 par le mode FBD et l'extension de communication réseau liaison série Modbus accepte les échanges par les deux modes FBD et LD [17] :

a Programmation en langage à contacts (LADDER)

En mode LADDER (langage à contacts), les 4 mots (16 bits) de données à échanger ne sont pas accessibles par l'application. Les transferts avec le maître sont implicites et s'opèrent de manière complètement transparente (Tableau 3-5).

ECHANGES MODBUS	CODE	NOMBRE DE MOTS
IMAGE DES E/S DU MODULE LOGIQUE	Lecture 03	4
MOTS D'HORLOGE	Lecture/Ecriture 16, 06 ou 03	4
MOTS DE STATUS	Lecture 03	1

Tableau 3-5 : Adressage des échanges Modbus avec le mode LADDER [17]

b Programmation par blocs fonctions (FBD)

En mode FBD, les 4 mots (16 bits) de données en entrée (de J1XT1 à J4XT1) et les 4 mots de données en sortie (de O1XT1 à O4XT1) sont accessibles par l'application. Les blocs fonctions de conversion permettent :

- De décomposer une entrée de type entier (16 bits) en 16 sorties de type "bit" : fonction CAN (conversion analogique numérique).

- De composer une sortie de type entier (16 bits) à partir de 16 entrées de type “bit” : fonction CNA (conversion numérique analogique).

Le tableau 3-6 représente l’adressage des échanges Modbus avec le mode FBD.

ECHANGES MODBUS	CODE	NOMBRE DE MOTS
MOTS D’ENTREE	Lecture/Ecriture 16, 06 ou 03	4
MOTS DE SORTIE	Lecture 03	4
MOTS D’HORLOGE	Lecture/Ecriture 16, 06 ou 03	4
MOTS DE STATUS	Lecture 03	1

Tableau 3-6 : Adressage des échanges Modbus avec le mode FBD [17]

3.4.5 NI Modbus serveur

Le logiciel NI LabVIEW peut communiquer avec un contrôleur logique programmable (PLC) de différentes manières. Modbus est un protocole de communication série édité par Modicon en 1979 pour communiquer avec les automates, puis étendu au protocole TCP.

Modbus est devenu l'un des protocoles de communication standard de facto dans l'industrie en raison de sa disponibilité [24].

3.5 Conclusion

Dans ce chapitre, nous avons donné un panorama de notions essentielles pour les logiciels utilisés dans la programmation de la gamme Zelio, puis nous avons présenté les caractéristiques de chacun d’eux.

Une description rapide de l’environnement LabVIEW a aussi été abordée suivie par la présentation du protocole de communication Modbus, son principe de fonctionnement et ses types de données. Enfin, on a cité les Extensions de communication réseau Liaison série Modbus et Ethernet Modbus/TCP.

Chapitre 4

Simulations et Tests

4.1 Introduction

Après avoir expliqué le but de notre projet et la procédure suivie pour le mettre en œuvre dans les précédents chapitres, ce chapitre présentera le principe de fonctionnement choisi de l'ascenseur.

Une fois défini, les programmes nécessaires seront développés avec les deux logiciels LabVIEW et Zeliosoft2 de Schneider (sous LADDER).

À la fin, on va faire une réalisation pratique à l'aide de l'automate programmable Zelio Logic de Schneider.

4.2 Principe de Fonctionnement

Comme mentionné auparavant, notre choix s'est fixé à un ascenseur à 3 étages, soient :

- Le rez-de-chaussée (RDC).
- Le premier étage (ET1).
- Et le deuxième étage (ET2).

Le déroulement du fonctionnement de notre ascenseur a été choisi et défini comme suit :

- Le passager provoque la mise en route de la cabine par appui sur un bouton d'envoi en cabine (RDCC, ET1C, ET2C) ou un bouton d'appel hors de la cabine (un bouton d'appel dans chaque étage : RDC, ET1, ET2).
- La cabine monte ou descend selon sa dernière position (son dernier arrêt).
- Une fois que la cabine est arrivée à l'étage, le capteur définit la position de la cabine (CP-RDC, CP-ET1, CP-ET2) qui s'active et enclenche l'arrêt de la cabine si elle est arrivée à destination.
- Ce même capteur de position active une temporisation (T0) qui va durer 2 secondes à partir de l'arrêt de la cabine pour ensuite ouvrir la porte cabine.
- Une fois que la porte est ouverte, le capteur « porte ouverte » active une autre temporisation (T1) qui va durer 3 secondes, cette temporisation permettra aux usagers de sortir ou de pénétrer à l'intérieur de la cabine en toute sécurité.
- Une fois le temps écoulé la porte cabine se ferme.

  A noter que :

- La priorité est pour la commande en cabine.
- Et s'il y'a plusieurs appels hors cabine, la priorité est pour l'appel le plus proche de la position de la cabine.
- Aussi, la priorité est toujours pour les appels en cabine.
- Pour la sécurité des personnes, l'ascenseur reste bloqué et n'accepte aucune commande si les conditions de démarrage (CID) ne sont pas satisfaites.
- Ces conditions CID sont : Bouton d'arrêt d'urgence actionné et Porte cabine ouverte.

4.3 Grafcet niveau 2

La Figure 4.1 présente le Grafcet niveau 2 de notre ascenseur, où :

CID : Condition initial de démarrage.

CIDS : Condition initial de démarrage satisfaite.

ET : Appel d'étage.

ETC : Appel d'étage en cabine.

E : Etage.

M : La montée.

D : La descente.

CP : Capteur de position.

T0 : Temporisation.

OU : Ouverture de la porte.

FE : Fermeture de la porte.

CPO : Capteur porte ouverte.

CPF : Capteur porte fermée.

M6 : Priorité d'appel en cabine pour étage 2.

M5 : Priorité d'appel en cabine pour étage 1.

M4 : Priorité d'appel en cabine pour RDC.

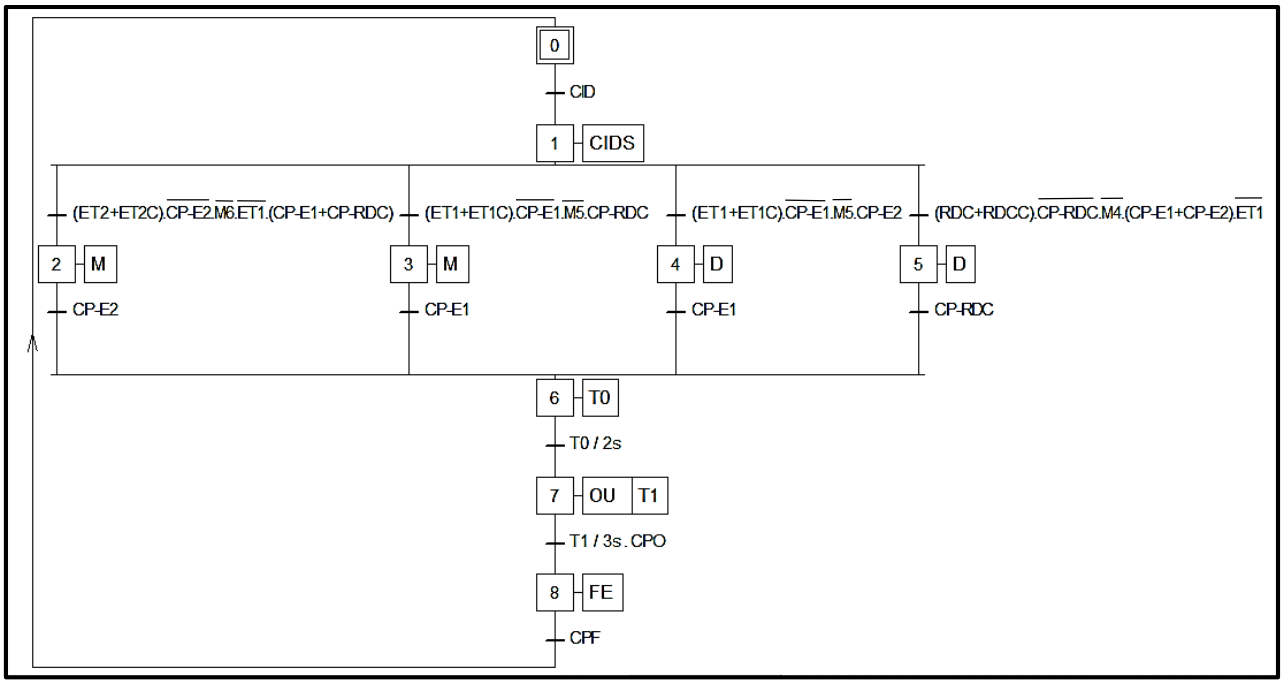

Figure 4.1 : Grafcet niveau 2 de l'ascenseur

4.4 Partie 1 : Simulation sur LabVIEW

Afin de s'assurer du bon fonctionnement de notre ascenseur, nous avons procéder à sa simulation sur le logiciel LabVIEW.

Cette opération est menée à terme grâce au panneau avant servant d'interface avec l'utilisateur et le diagramme contenant le programme source en langage graphique G, en faisant appel à des palettes indépendantes d'outils et d'objets permettant d'éditer les deux fenêtres du programme et de tester son fonctionnement.

4.4.1 Réalisation de l'interface graphique

La Figure 4.2 présente la face avant que nous avons programmé. C'est l'interface de notre ascenseur où on affiche l'état de chaque étage grâce à des voyants, des afficheurs de temporisation, des détecteurs, des boutons d'appels, etc.

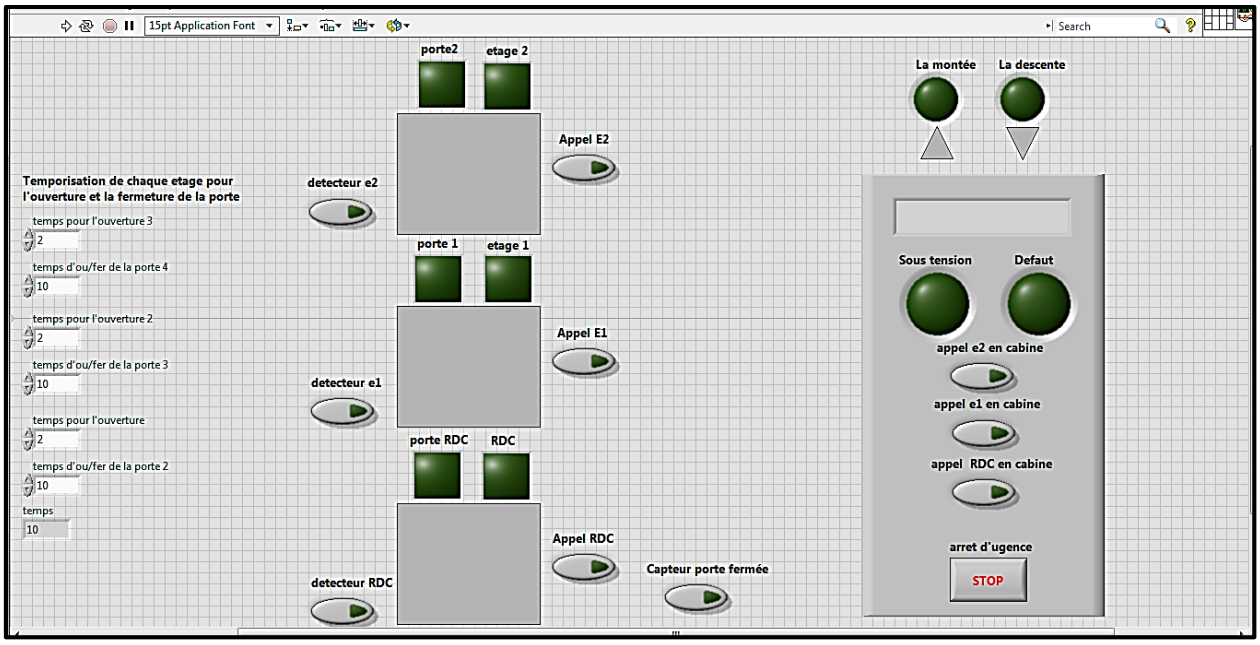

Figure 4.2 : La face avant du programme

4.4.2 Réalisation du diagramme de programme

La partie programmation sur LabVIEW se traduit par la réalisation du block diagramme. Les différentes parties de notre programme sont décrites ci-dessous.

Note : Avant chaque demande d'étage, la priorité est toujours pour les appels en cabine (Voir Annexe A).

a Appel RDC

Pour demander la cabine en rez-de-chaussée RDC, on clique soit sur le bouton d'appel RDC, soit sur le bouton d'appel RDC en cabine (Figure 4.3).

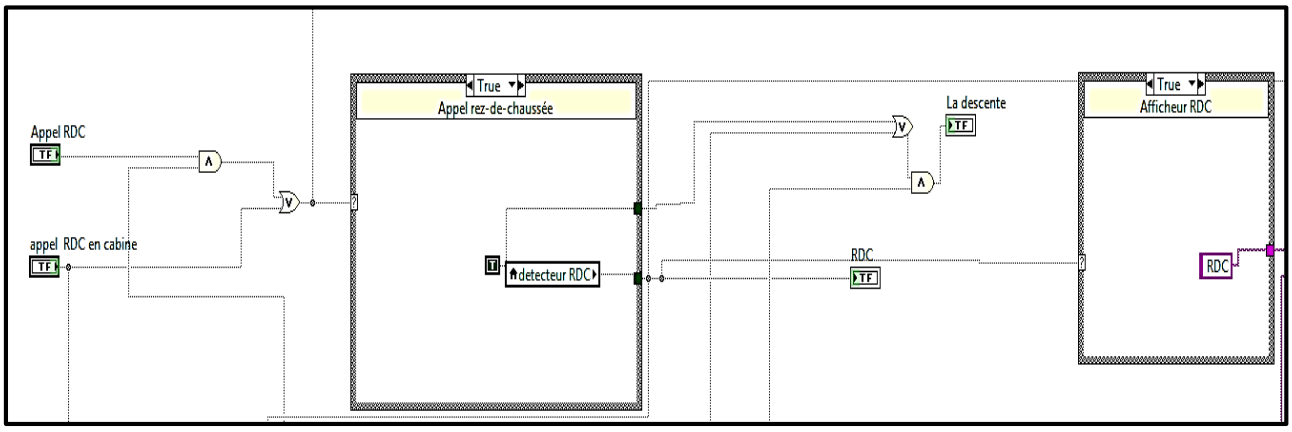

Figure 4.3 : Appel RDC

b Appel Étage 1

Pour appeler l'étage 1, on clique sur le bouton d'appel E1 à l'extérieur, ou sur le bouton d'appel E1 en cabine.

- Si la cabine est au RDC, elle doit monter pour atteindre l'étage 1.
- En revanche si la cabine est à l'étage 2, elle doit descendre vers l'étage 1.

La Figure 4.4 montre l'appel de l'étage 1.

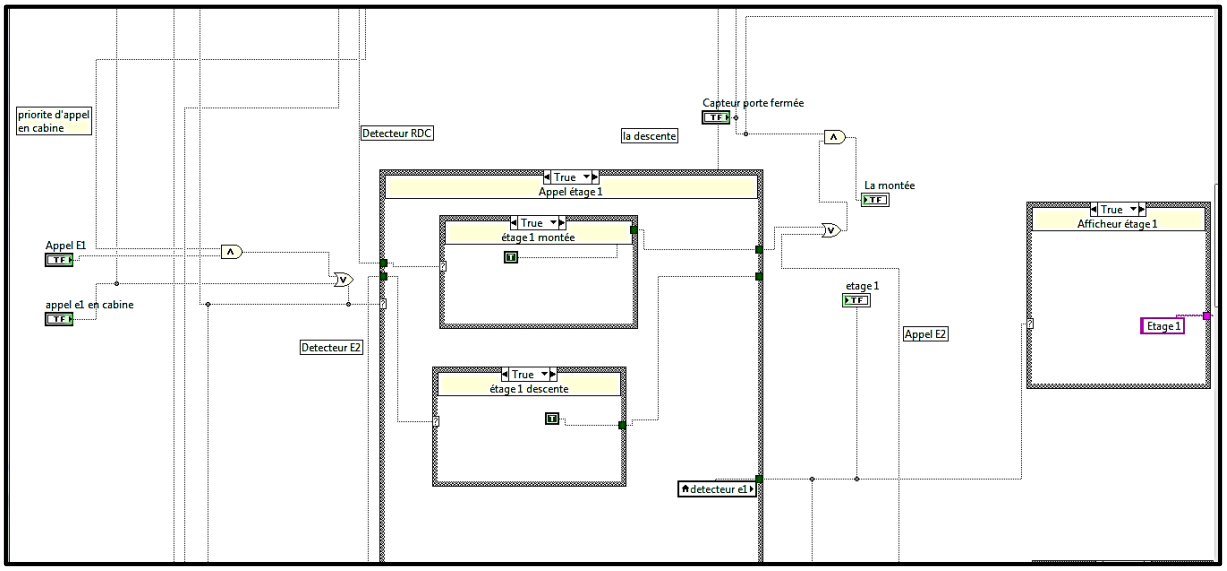

Figure 4.4 : Appel Étage 1

c Appel Étage 2

Pour appeler l'étage 2, on clique soit sur le bouton d'appel E2 à l'extérieur, soit sur le bouton d'appel E2 en cabine.

La Figure 4.5 présente l'appel de l'étage 2.

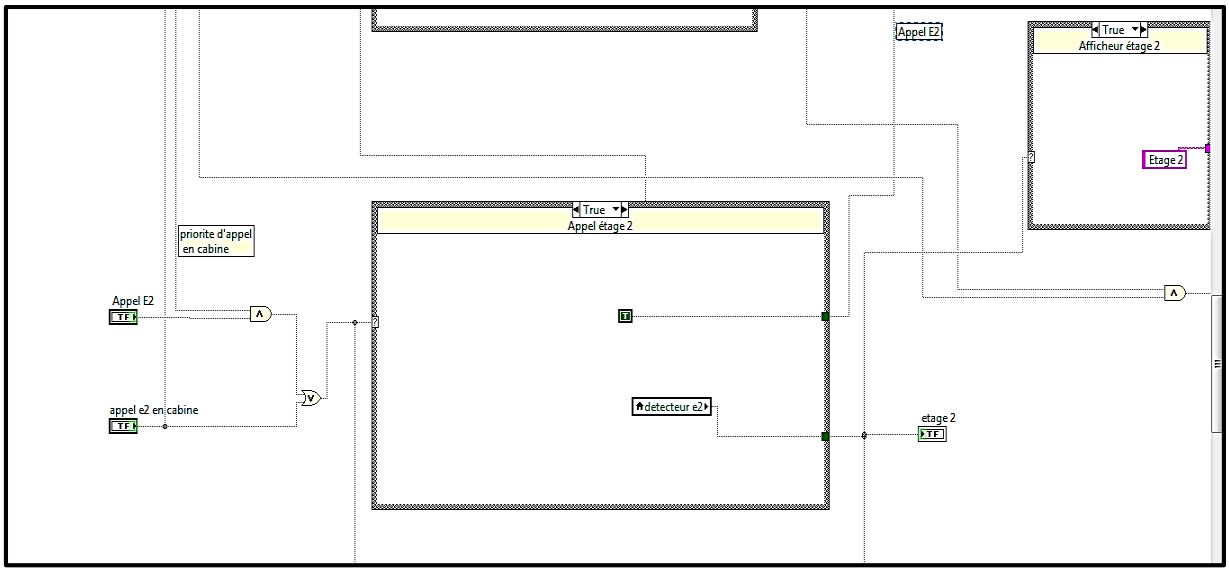

Figure 4.5 : Appel Étage 2

d Ouverture / Fermeture de la porte au RDC

Si on est à l'étage 1 ou 2 et que c'est le RDC qui est demandé, alors après avoir atteint le RDC, nous passons à la deuxième commande, qui est l'ouverture de la porte.

Après 10 secondes, une troisième commande est activée pour la fermer.

La Figure 4.6 présente l'ouverture et la fermeture de la porte au RDC.

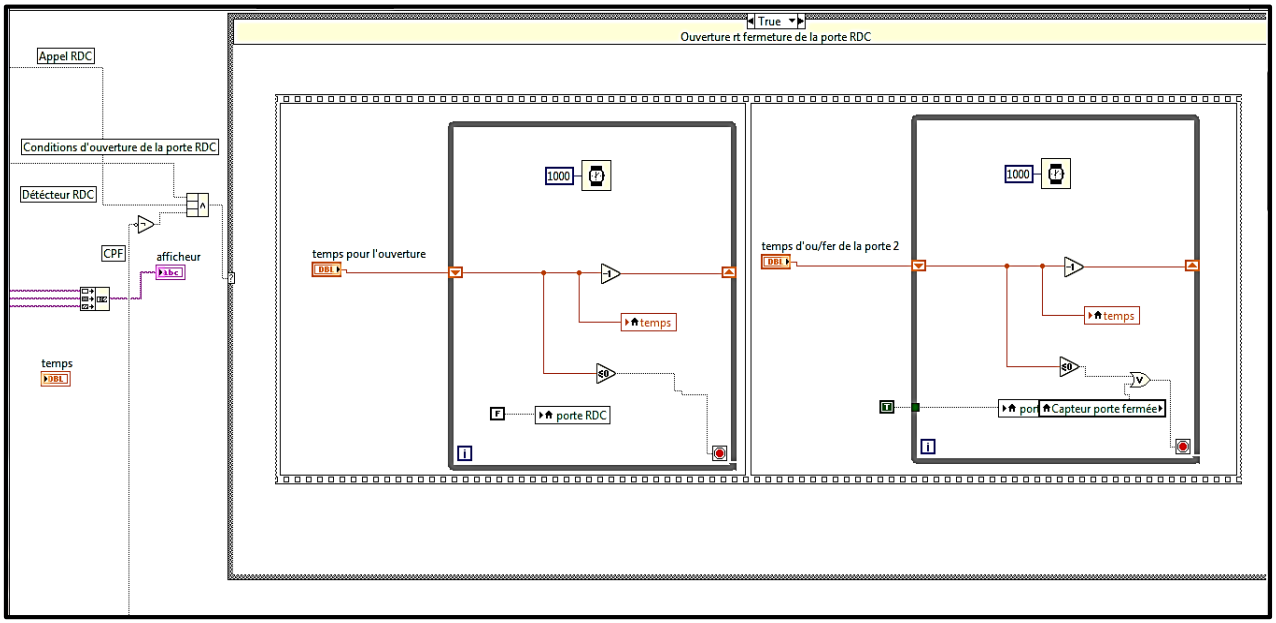

Figure 4.6 : Ouverture et fermeture de la porte au RDC

e Ouverture / Fermeture de la porte à l'Étage 1

Si on est à l'étage 2 ou RDC et que c'est l'étage 1 qui est demandé, alors après avoir atteint en E1, nous passons à la deuxième commande, qui est l'ouverture de la porte.

Après 10 secondes, une troisième commande est activée pour la fermer.

La Figure 4.7 présente l'ouverture et la fermeture de la porte à l'Étage 1.

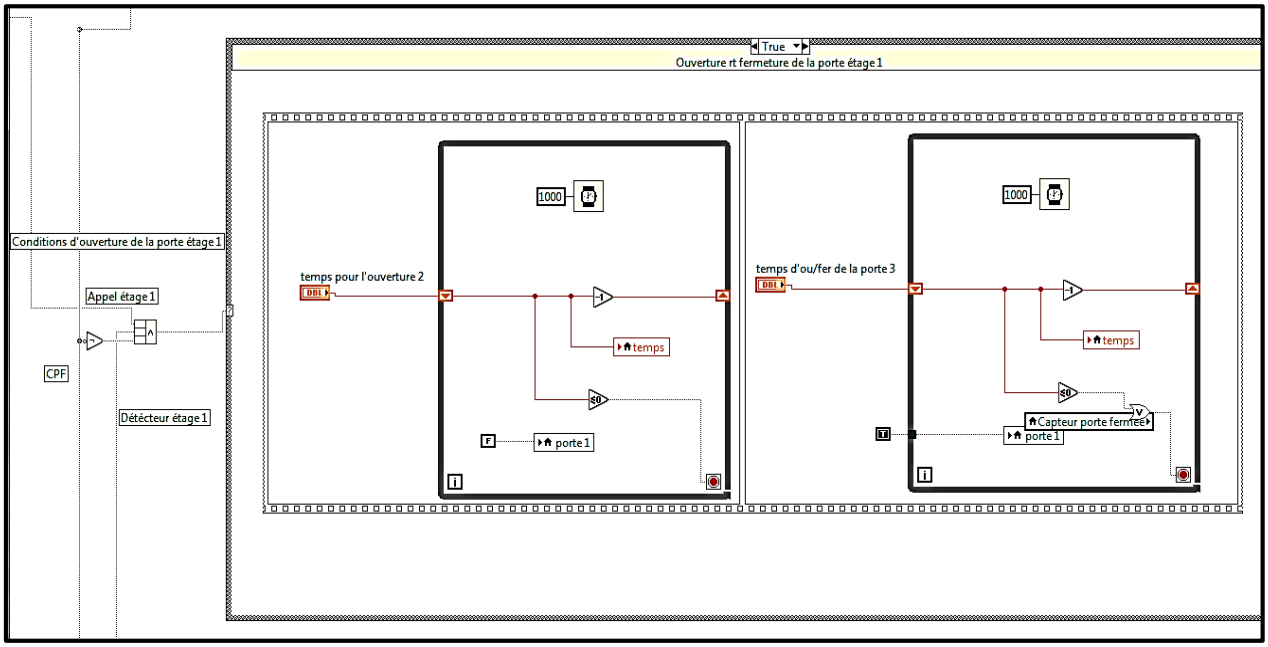

Figure 4.7 : Ouverture et fermeture de la porte à l'Étage 1

f Ouverture / Fermeture de la porte à l'Étage 2

Si on est à l'étage 1 ou RDC et que c'est l'étage 2 qui est demandé, alors après avoir atteint l'étage 2, nous passons à la deuxième commande, qui est l'ouverture de la porte.

Après 10 secondes, une troisième commande est activée pour la fermer.

La Figure 4.8 représente l'ouverture et la fermeture de la porte à l'Étage 2.

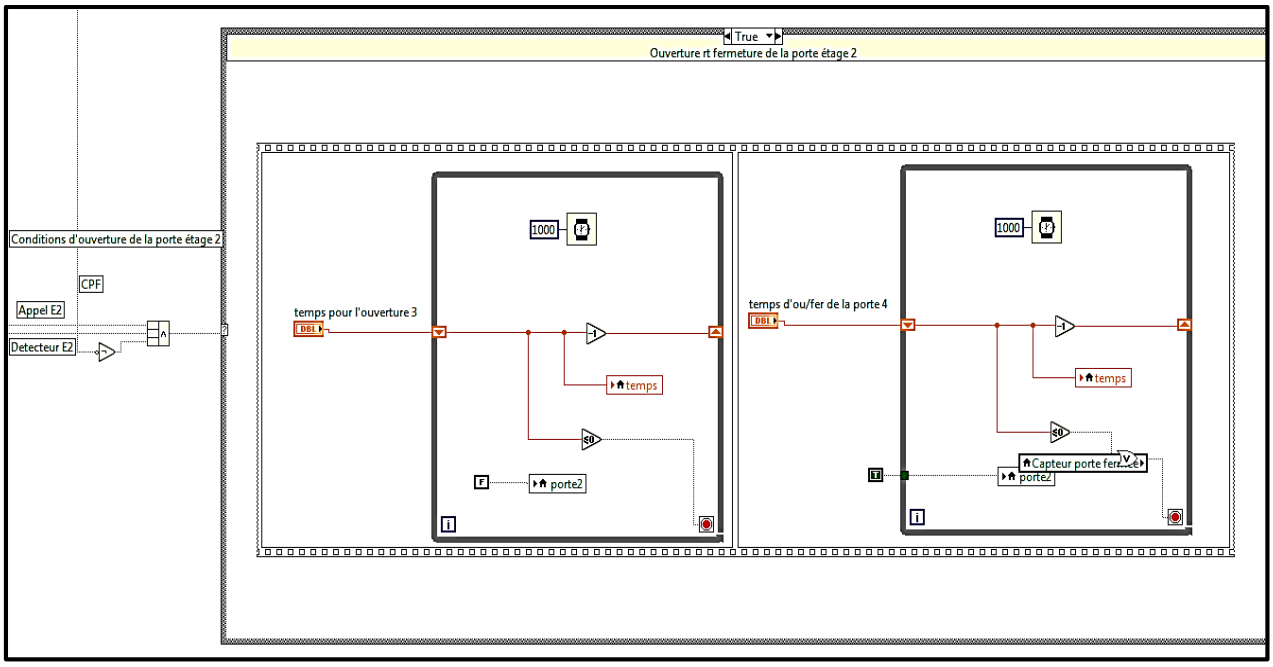

Figure 4.8 : Ouverture et fermeture de la porte à l'Étage 2

4.5 Partie 2 : Réalisation pratique avec Zelio Logic

Dans cette partie, on va utiliser le logiciel Zeliosoft2 pour programmer l'automate Zelio Logic (SR3B261FU) qui va commander la maquette de l'ascenseur réalisée.

4.5.1 Description de l'équipement

L'ensemble de l'ascenseur est composé de deux parties :

a La partie opérative

C'est la partie qui contient une maquette d'ascenseur montée sur un support. Elle est munie de plusieurs boutons et voyants qui correspondent dans la réalité à des actionneurs, des capteurs, des boutons et des voyants nécessaires au fonctionnement d'un ascenseur réel.

b La partie commande

Qui contient l'automate Zelio Logic de type (SR3B261FU) de Schneider Electric contenant un programme en LADDER que nous avons développé pour le fonctionnement de la maquette d'ascenseur. L'alimentation est à 220 Vcc.

4.5.2 Description des Entrées

La Figure 4.9 montre les différentes entrées utilisées de type TOR dans notre maquette d'ascenseur :

No	Bloc	Commentaire
Entrées TOR		
01	I1	appel RDC
02	I2	appel e1
03	I3	appel e2
04	I4	appel RDC en cabine
05	I5	appel e1 en cabine
06	I6	appel e2 en cabine
07	I7	detecteur RDC
08	I8	detecteur e1
09	I9	detecteur e2
10	IA	detecteur du poid
11	IB	capteur porte ouverte
12	IC	arret d'urgence
13	ID	capteur porte fermer

Figure 4.9 : Les entrées présentes sur la maquette

Où nous avons :

- Les boutons d'appels : Il y a un bouton poussoir à chaque étage pour appeler l'ascenseur.
- Les boutons d'envois : qui correspondent aux boutons situés à l'intérieur de la cabine d'ascenseur.
- Arrêt d'urgence : Un bouton poussoir (bouton coup de poing) qui permet de bloquer l'ascenseur en cas de danger.
- Détecteur de l'étage : Un bouton poussoir qui détecte la position de la cabine.
- Capteur porte ouverte/ Capteur porte fermée : Deux boutons poussoirs qui assurent l'ouverture et la fermeture de la porte en toute sécurité.

Le Tableau 4.1 résume les différents paramètres expliqués précédemment :

ADRESSE	FONCTION	MNEMONIQUE
%I.1	Bouton poussoir	Appel RDC
%I.2	Bouton poussoir	Appel Étage 1
%I.3	Bouton poussoir	Appel Étage 2

%I.4	Bouton poussoir	Appel RDC en cabine
%I.5	Bouton poussoir	Appel E1 en cabine
%I.6	Bouton poussoir	Appel E2 en cabine
%I.7	Bouton poussoir	Détecteur RDC
%I.8	Bouton poussoir	Détecteur E1
%I.9	Bouton poussoir	Détecteur E2
%I.B	Bouton poussoir	Capteur porte ouvert
%I.D	Bouton poussoir	Capteur porte fermé
%I.C	Bouton poussoir(bouton coup de poing)	Arrêt d'urgence

Tableau 4-1 : Fonction et mnémonique des Entrées utilisées

4.5.3 Les Sorties

La Figure 4.10 montre les différentes sorties rencontrées dans la maquette de notre ascenseur :

No	Bloc	Commentaire
Sorties TOR		
01	Q1	monter la cabine
02	Q2	descent la cabine
03	Q3	sous tension
04	Q4	ou/ferm la porte e1
05	Q5	ou/ferm la porte e2
06	Q6	lampe RDC
07	Q7	lampe e1
08	Q8	lampe e2
09	Q9	ouv/ferm de la porte RDC
10	QA	Dèfaut

Figure 4.10 : Les sorties associées aux programme

Où nous avons :

- Des voyants : où chaque voyant donne un signal qui sert à une action qui est expliquée dans le Tableau 4.2.

ADRESSE	FONCTION	MNEMONIQUE
%Q1	Voyant vert	Montée la cabine
%Q2	Voyant rouge	Descente la cabine
%Q3	Voyant vert	Sous tension
%Q6	Voyant vert	Voyant d'RDC
%Q7	Voyant vert	Voyant d'étage 1
%Q8	Voyant vert	Voyant d'étage 2
%Q9	Voyant rouge	Ouverture / fermeture de la porte RDC
%Q4	Voyant rouge	Ouverture/ fermeture de la porte E1
%Q5	Voyant rouge	Ouverture/ fermeture de la porte E2
%QA	Voyant rouge	Arrêt d'urgence

Tableau 4-2 : Fonction et mnémonique des sorties utilisées

4.5.4 Description du programme

Le programme implémenté dans notre API ZELIO a été écrit en langage ladder grâce au logiciel Zeliosoft2. Il est composé de parties comme suit :

a Demande RDC

La Figure 4.11 présente l'appel de RDC.

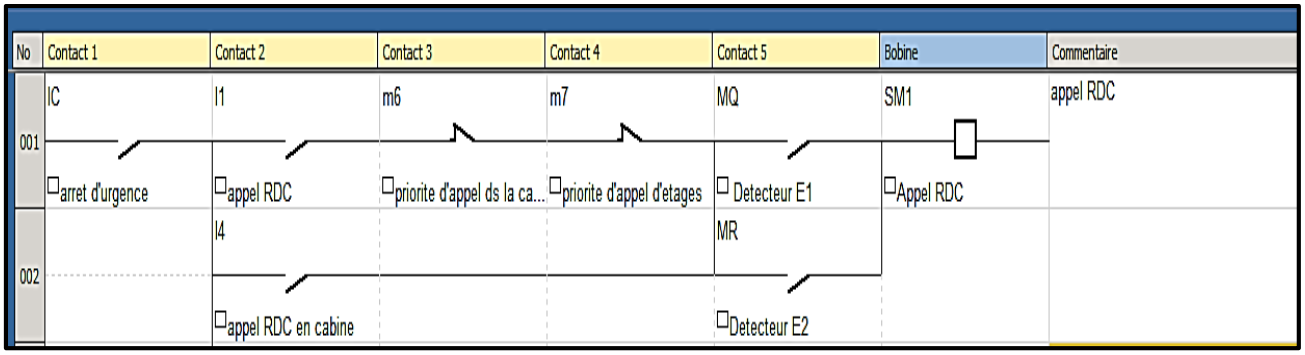

Figure 4.11: Appel RDC

b Demande Étage1

L'appel de l'étage 1 contient deux actions :

- Si la cabine est à l'étage 2 donc activer l'action de la descente (elle descend).
- Par contre si la cabine est au RDC donc activer l'action de la montée (elle monte).

La Figure 4.12 présente l'appel de l'étage 1.

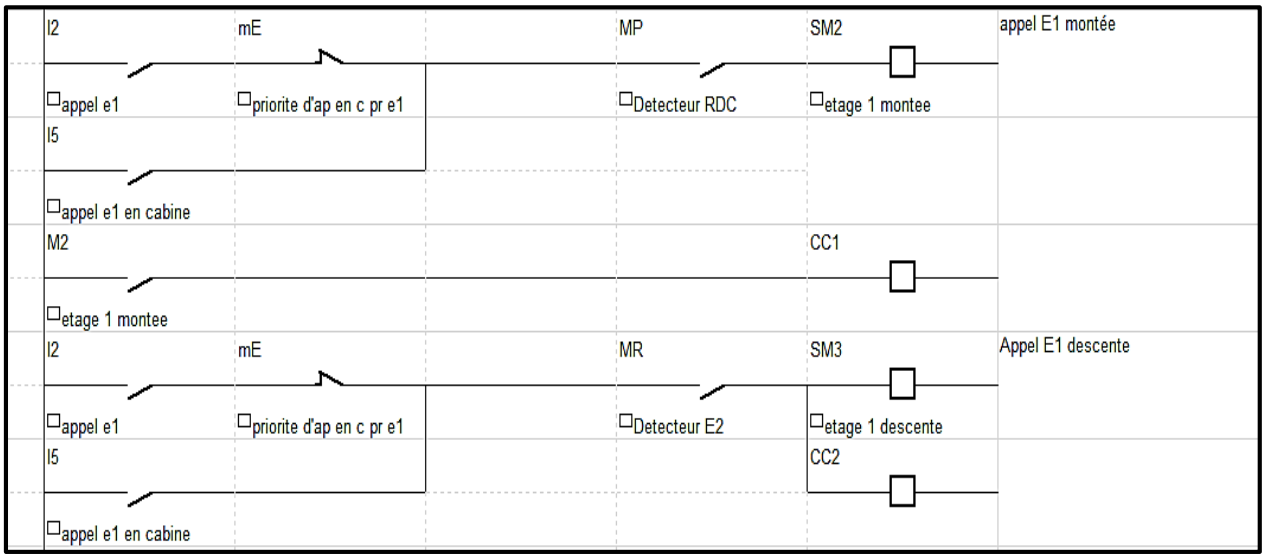

Figure 4.12 : Appel Étage 1

c Demande Étage2

En suivant le même raisonnement, la Figure 4.13 représente l'appel de l'étage 2.

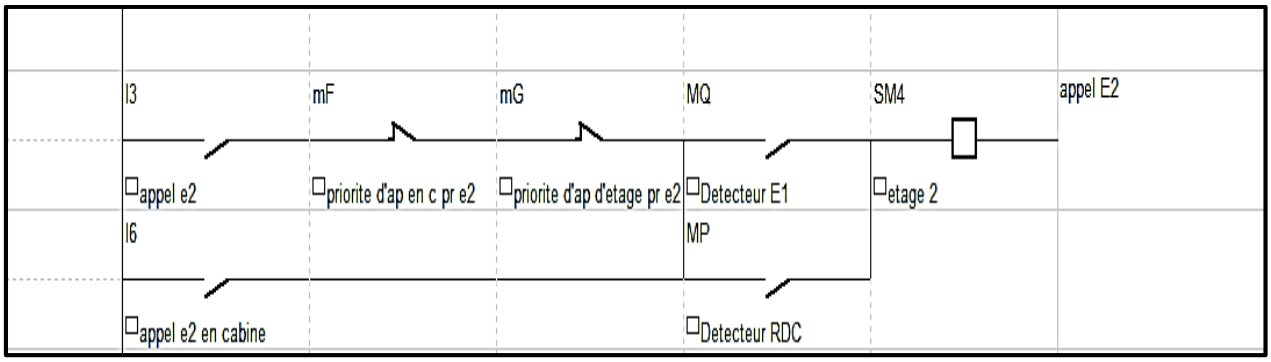

Figure 4.13 : Appel Étage 2

d La phase montée

La Figure 4.14 présente la phase montée et les conditions d'arrêt.

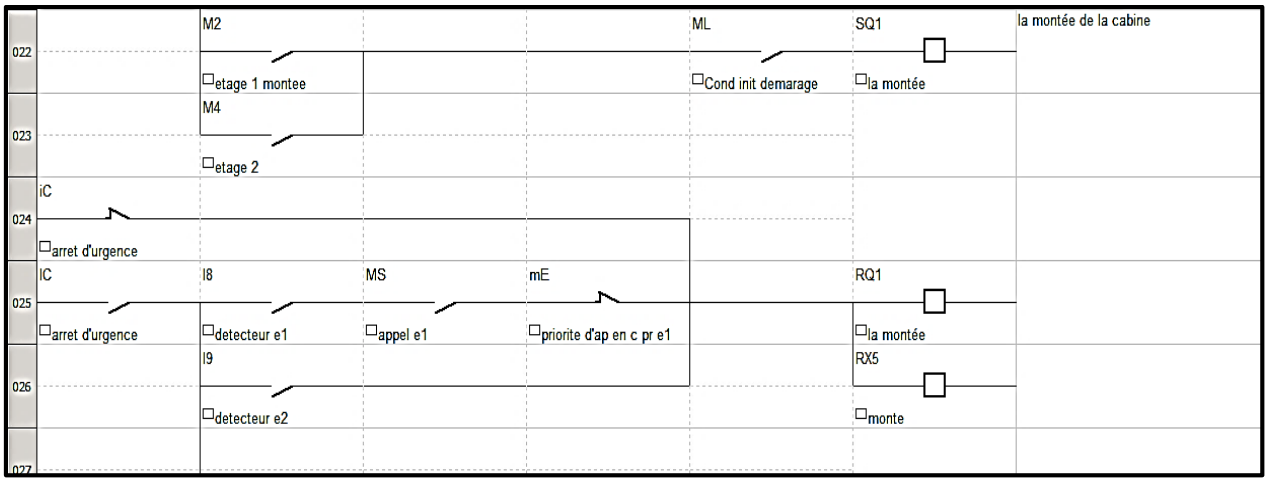

Figure 4.14 : La montée

e La phase descente

La Figure 4.15 présente la phase de descente et les conditions d'arrêt.

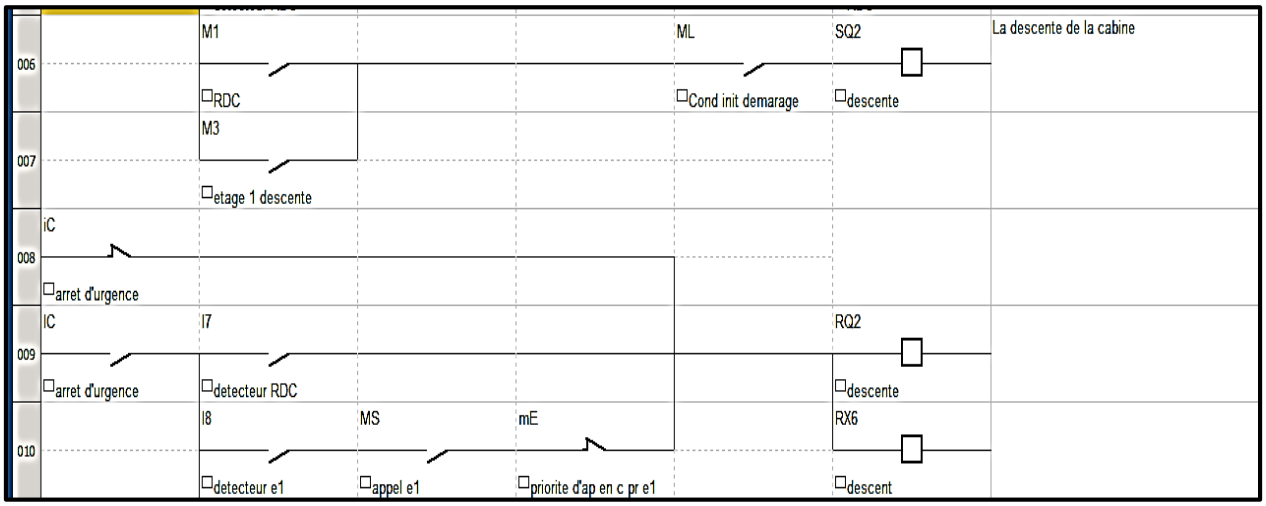

Figure 4.15 : La descente

f Ouverture de la porte

A l'arrêt, à chaque étage :

- La porte s'ouvre automatiquement après une attente de deux secondes de protection.
- Elle reste ouverte pendant trois secondes.
- Puis elle se referme automatiquement.

La Figure 4.16 montre la temporisation de 2s.

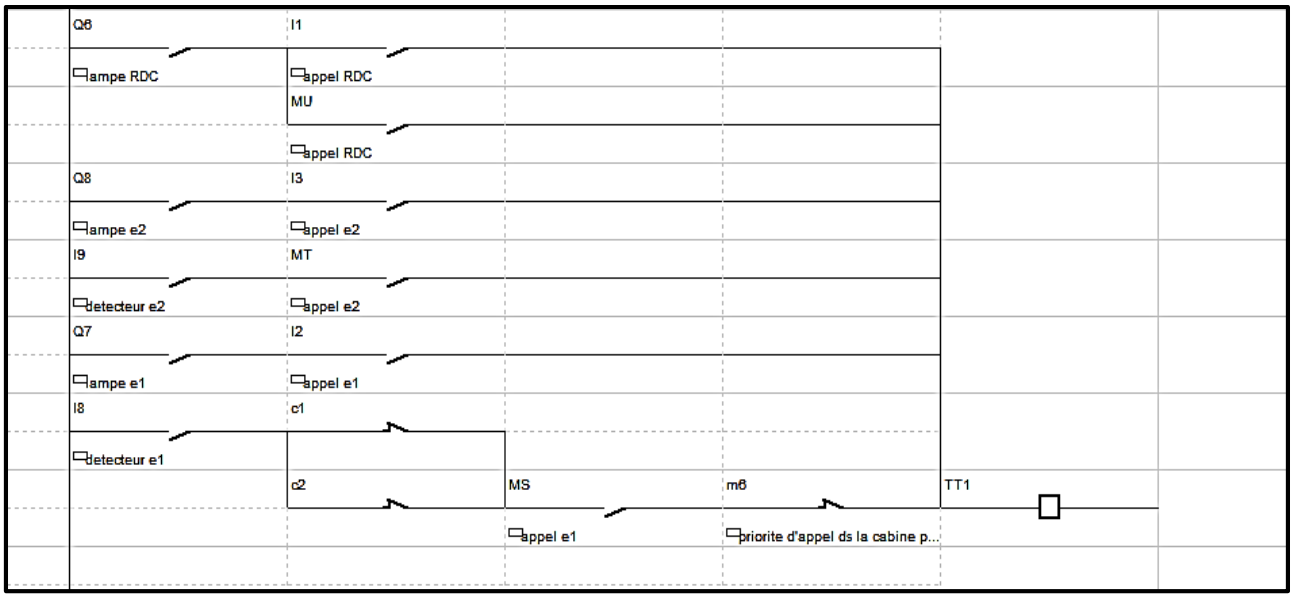

Figure 4.16 : Temporisation 2s

La Figure 4.17 présente l'ouverture de la porte :

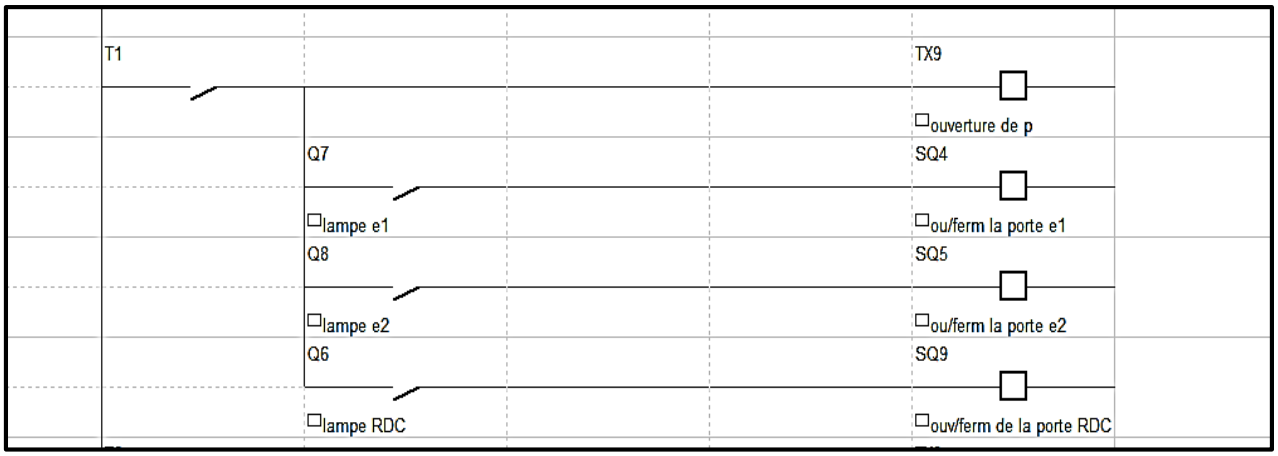

Figure 4.17 : Ouverture de la porte

g Fermeture de la porte

La Figure 4.18 représente la fermeture de la porte.

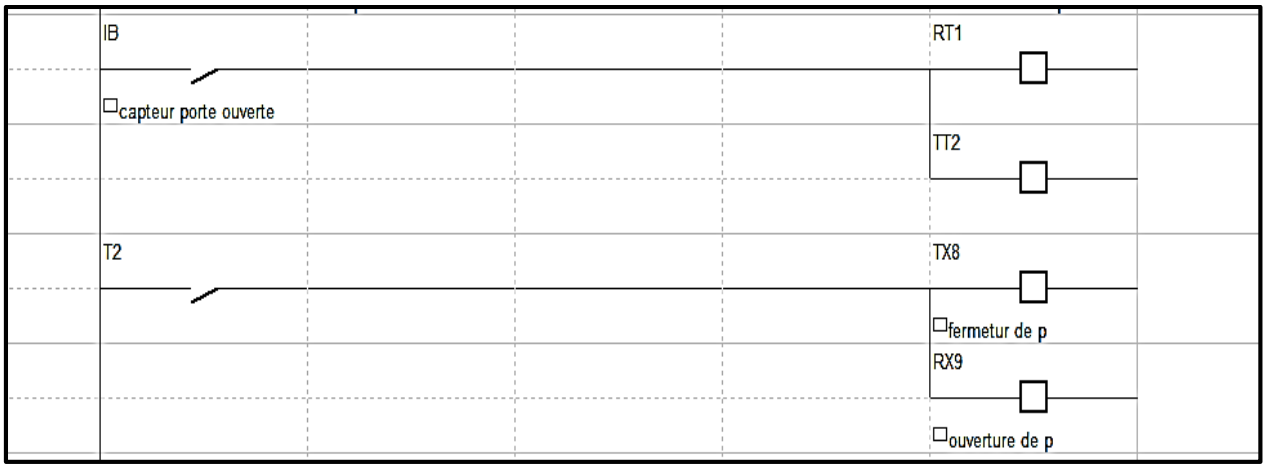

Figure 4.18 : Fermeture de la porte

Note : Pour la suite du programme, voir Annexe B.

4.5.5 Description finale de la maquette

Notre réalisation pratique (maquette d'un ascenseur à trois étages) est montrée sur la figure 4.19.

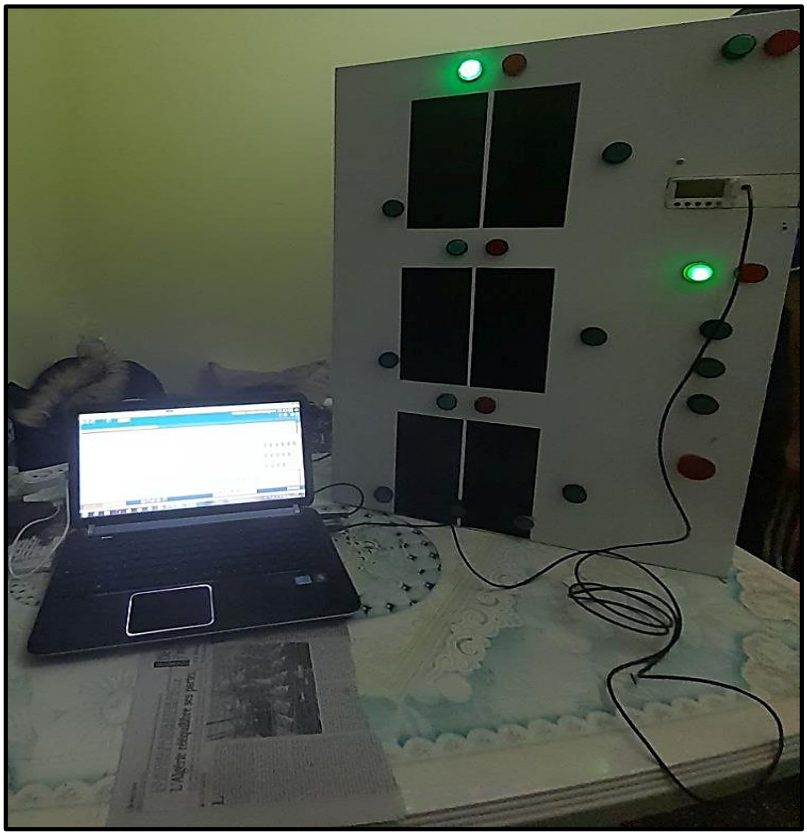

Figure 4.19 : Aperçu final de la maquette

La Figure 4.20 présente la face arrière de notre maquette qui montre la liaison entre l'automate et la partie opérative.

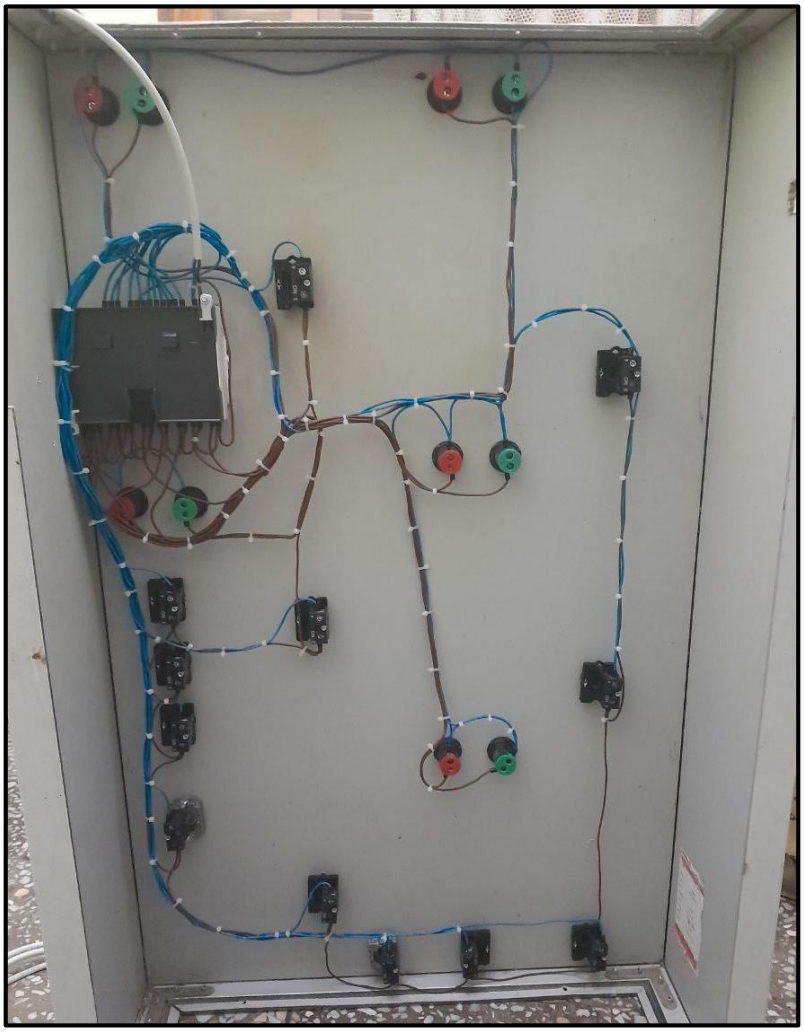

Figure 4.20 : Branchement des E/S de l'automate avec la partie opérative

4.6 Conclusion

Dans ce chapitre, on a présenté tout d'abord le principe de fonctionnement de notre ascenseur.

A partir de ce dernier, on a réalisé alors une simulation correspondante en utilisant le logiciel LabVIEW. Enfin, on a terminé par une réalisation pratique d'une maquette fonctionnant à l'aide de l'automate Zelio Logic ce qui nous a permis de mieux comprendre le fonctionnement de cet ascenseur.

Conclusion

Générale

Le but de notre travail était de simuler et de réaliser la commande d'un ascenseur à l'aide du logiciel LabVIEW et d'un automate ZELIO LOGIC de type (SR3B261FU) de SCHNEIDER.

Dans ce mémoire, nous avons parlé de l'ascenseur en général, les différentes types et catégories, la description des API d'une façon générale et plus particulièrement l'automate ZELIO LOGIC de SCHNEIDER avec lequel nous avons travaillé.

La simulation de notre projet a été mise en œuvre par le logiciel LabVIEW et le logiciel Zeliosoft2 (le langage LADDER).

On a terminé notre travail par une réalisation pratique d'une maquette à base de l'automate programmable ZELIO LOGIC de type (SR3B261FU) de Schneider dont le choix a été basé sur deux critères importants et qui sont le coût et la performance.

Le choix de ce système s'est avéré bénéfique et très intéressant, car sa mise en place a nécessité l'application de multiples disciplines techniques. De plus, étant un moyen de déplacement largement utilisé et en constante expansion, il nous a permis de comprendre son fonctionnement, d'explorer et de maîtriser de nouveaux logiciels de contrôle et d'automatisation, contribuant ainsi à notre acquisition de connaissances.

Cependant, la non disponibilité et le prix cher du module de communication Modbus de SCHNEIDER, ne nous a pas permis de mettre en réalisation l'interfaçage direct entre les deux logiciels LabVIEW et Zeliosoft2.

Toutefois, nous sommes optimistes quant au bénéfice que peut apporter ce travail à d'autres étudiants dans le futur en leur fournissant des informations techniques précieuses sur les ascenseurs et la simulation de leur fonctionnement sur LabVIEW.

Comme perspectives possibles pour le futur, nous pouvons citer :

- La simulation d'un ascenseur avec plus d'étages.

- La réalisation pratique en utilisant l'interfaçage entre les deux logiciels de programmation.
- L'augmentation des moyens de protection de l'ascenseur (montée, descente, arrêt).
- L'amélioration et le développement d'un programme plus complet afin d'améliorer le fonctionnement de l'ascenseur.
- L'évaluation des besoins futurs en matière d'ascenseurs dans différentes zones urbaines et proposition de solutions pour répondre à ces besoins de manière efficace et durable.

En conclusion, ce mémoire de fin d'étude a été une expérience extrêmement enrichissante pour moi. J'espère que mon travail servira de référence et de base pour ma future carrière professionnelle et qu'il sera bénéfique également aux générations futures.

Références

Bibliographiques

[1] « Ascenseur : qu'est-ce que c'est ? » par FUTURA-SCIENCES, 2001-2022, https://www.futura-sciences.com/tech/definitions/technologie-ascenseur-11102 / (consulté le 18.02.2023).

[2] « Ascenseur », 2009, https://pompiereddy.skyrock.com/3.html (consulté le 18.02.2023)

[3] T. Boutaghane, I. Kebsa, « Commande d'un Ascenseur Trois Étages par Automate Programmable », Mémoire de Master en Électronique, Université Mohamed Seddik Benyahia, Jijel, Algérie, 2021.

[4] « Les Ascenseurs Hydrauliques », AFEM - Ascenseur Fabrication Entretien Montage, http://www.afem.com/services/ascenseur-2/ (consulté le 23.02.2023).

[5] « Les Ascenseurs à Traction à Câbles », AFEM - Ascenseur Fabrication Entretien Montage, http://www.afem.com/services/ascenseur-3/ (consulté le 02/04/2023)

[6] M. Djidel, « Commande d'un Ascenseur par API », Mémoire d'Ingéniorat en Automatique, ENP, Algérie 2021.

[7] « Systèmes de Motorisation », 2007, https://energieplus-lesite.be/techniques/ascenseurs7/systemes-de-motorisation/ (consulté le 22.02.2023).

[8] L. Boualam, N. Hachiche , «Conception et Réalisation d'une Carte de Commande d'une Maquette d'Ascenseur à base d'une Carte Arduino Mega 2560 », Mémoire de Master Professionnel en Automatique, Université Mouloud Mammeri, Tizi- Ouzou, Algérie, 2016.

[9] Y. Rigui, A. Cherirat, « Commande d'un Ascenseur à l'aide d'un Automate Programmable Industriel » Mémoire de master en Électrotechnique Industrielle, Université Kasdi Merbah Ouargla, Algérie, 2022.

[10] « Législation Ascenseur : Quels sont les Dispositifs de Sécurité Obligatoires ? », par maison Drieux-Combaluzier,2018, https://www.drieux-combaluzier.com/legislation-ascenseur-dispositifs. , (consulté le 22.02.2023).

[11] J. Luc, « L'automate programmable ? », Info technique, 2010, http://www.projeltestautomatisation.com/info-technique/lautomate-programmable , (consulté le 01.03.2023).

[12] « Les Automates Programmables Industriels (API) », Cours par technologuePro, 2017, https://www.technologuepro.com/cours-automate-programmable-industriel/Les-automates-programmables-industriels-API.htm ,(consulté le 01.03.2023).

[13] H. Ayad, « PLC», Cours Master1 Automatiques et Systèmes, 2021-2023,

Université Saad Dahlab Blida 1, Blida, Algérie.

[14] E. Maalem, I. Taouadji , « Les Langages de Programmation de l'Automate Programmable Industriel : Application Pilotage d'un Ascenseur) », Mémoire de Master en Commande des Machines électriques, Université d'Adrar , Algérie 2017.

[15] Lucas, « Langages de Programmation Automate (API) », par ScieTech, 2021, https://scietech.fr/langages-de-programmation-automate-api-scietech/ (consulté le 03.03.2023).

[16] « GRAFCET | AUTOMATISME #9: Forçage et Figeage », Cours par électronique-mixte, https://www.electronique-mixte.fr/grafcet-automatisme/grafcet-automatisme-9-forcage-et-figeage/ , (consulté le 03.03.2023).

[17] « Zelio Logic Smart Relais » de Schneider Electric 2017, https://www.tme.com/Document/7ca3b3e6a9c18c5aa34ef7c1d04ea045/ZELIO%20LOGIC%20-%20EN.pdf , (consulté le 05.03.2023).

[18] « Modules logiques Zelio Logic », https://docs.rs-online.com/123a/0900766b804bae56.pdf (consulté le 05.04.2023).

[19] « Zelio Logic 2 Module Logique Manuel utilisateur » de Schneider Electric 2007, https://docs.rs-online.com/90e8/0900766b8152ee6a.pdf (consulté le 07.04.2023).

[20] « Zelio Logic Programming Guide » de Schneider Electric 2017, Zelio Logic Programming Guide (schneider-electric.com) (consulté le 10.04.2023).

[21] « LabVIEW Programmation et applications Introduction à LabVIEW NXG » de F.Cottet, M.Pinard, L.Desruelle 2015 https://fr.readkong.com/page/fullscreen/labview-programmation-et-applications-introduction-1248062 (consulté le 18/04/2023).

[22] « Modbus » https://fac.umc.edu.dz/ista/pdf/cours/L3-GIM_R%C3%A9seau%20automates-cours5_modbus.pdf (consulté le 22/04/2023).

[23] « Zelio Logic - module de communication esclave réseau Modbus 24Vcc SR3MBU01BD » de Schneider Electric https://www.se.com/fr/fr/product/SR3MBU01BD/zelio-logic-module-de-communication-esclave-r%C3%A9seau-modbus-24vcc/ (consulté le 25/04/2023).

[23] « Connect LabVIEW to Any PLC With Modbus » 2022 https://knowledge.ni.com/KnowledgeArticleDetails?id=kA03q000000x0QgCAI&l=en-DZ (consulté le 25/04/2023).

Annexes

A.1 Priorité d'appel en cabine

Pour la demande de la cabine, la priorité (ou l'avantage) est pour les demandes en cabine. La Figure A.1, présente la priorité d'appel de la commande en cabine.

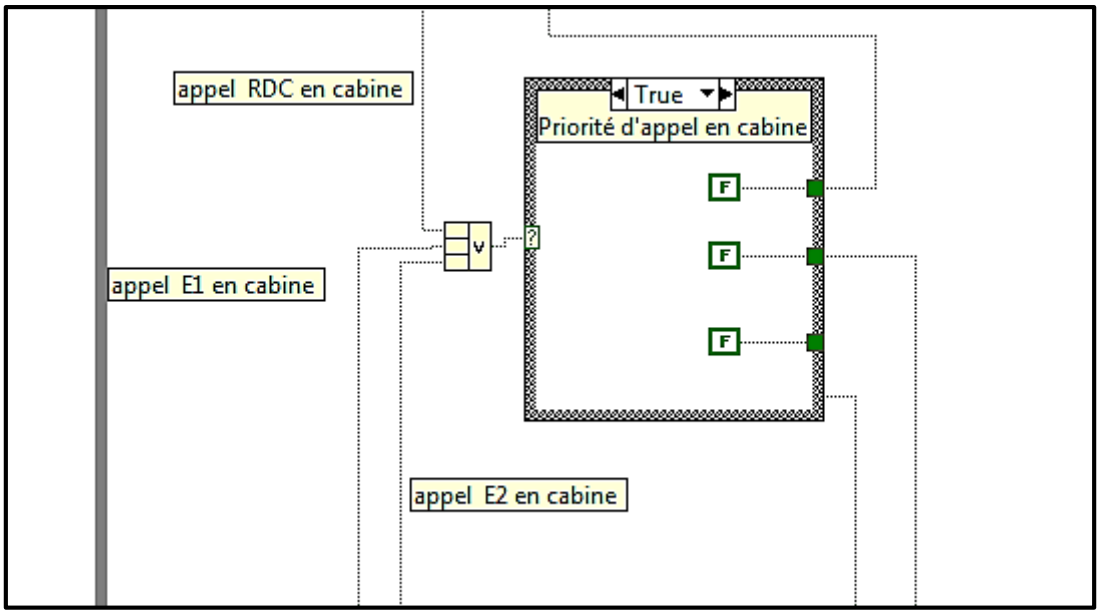

Figure A.1 : Priorité d'appel en cabine pour les trois étages

A.2 Exemple de simulation sur la face avant

La Figure A.2 présente un exemple de simulation sur la face avant. Quand le capteur indique l'arrivage de la cabine à l'étage 2 :

- Ouverture de la porte et le voyant rouge s'allume durant 10sec (On a utilisé un seul capteur pour l'ouverture et la fermeture de la porte c'est CPF).
- Allumage du voyant de l'étage 2 (voyant vert) automatiquement.

.

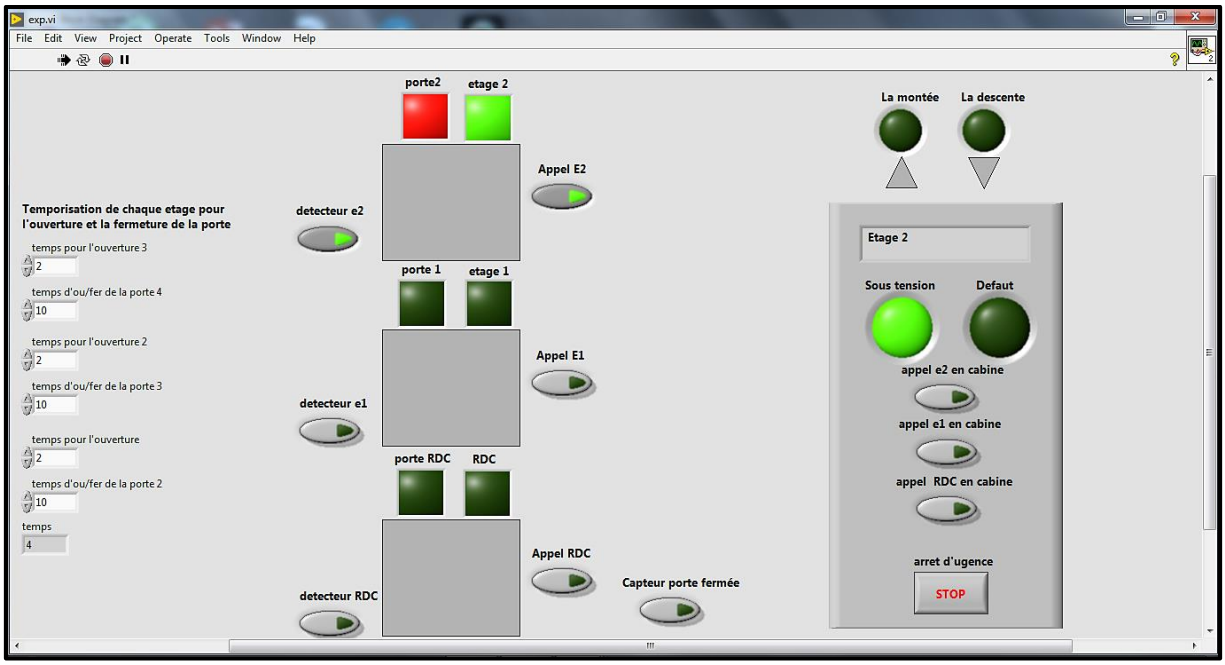

Figure A.2 : Vision générale d'un cas de simulation du programme

B.1 Conditions d'arrêt pour RDC

La Figure B.1 explique les conditions d'arrêt de la demande du RDC où les conditions suivantes doivent être vérifiées selon l'ordre de priorité :

- On a le détecteur de la cabine en RDC,
- Aussi on a la priorité d'appel en cabine (Appel E1 en cabine ou Appel E2 en cabine).
- Et on a l'appel de l'étage 1 parce que si la cabine est en étage 2 donc l'appel de l'étage 1 est prioritaire (dans le cas où l'appel de RDC et E1 sont appuyés en même temps).

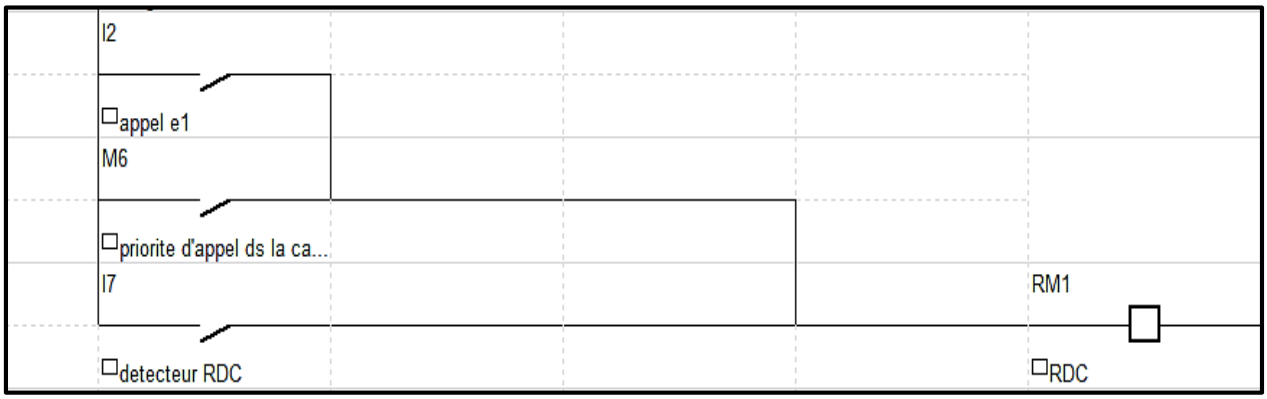

Figure B.1 : Conditions d'arrêt pour RDC

B.2 Conditions d'arrêt pour l'étage 1

Pour l'arrêt de la demande de l'étage 1 (Figure B.2), on a le détecteur de la cabine en Etage 1, après on a deux conditions pour l'arrêt :

- La première c'est l'appel de l'étage 1 (soit en cabine ou en étage).
- Et la deuxième condition est la priorité d'appel en cabine (Appel RDC en cabine ou Appel E1 en cabine).

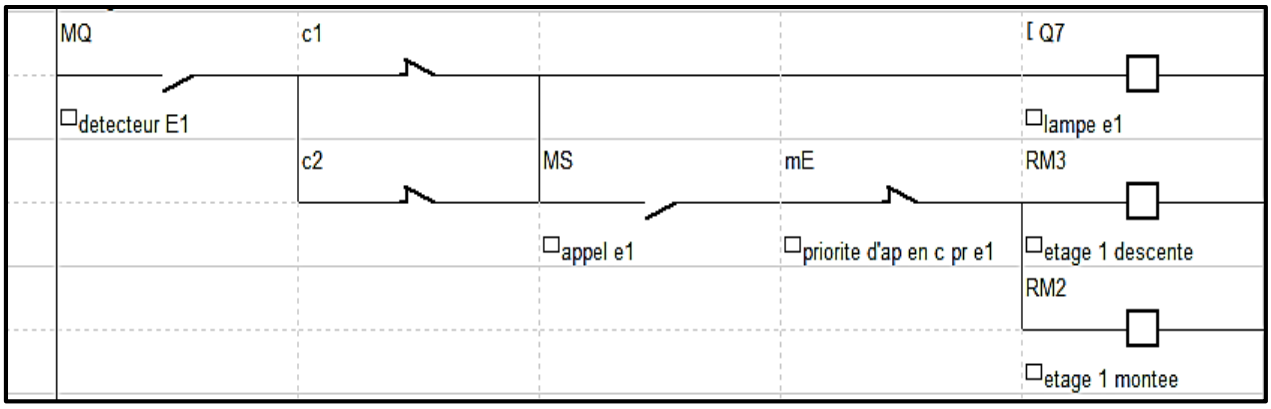

Figure B.2 : Conditions d'arrêt pour E1

B.3 Conditions d'arrêt pour l'étage 2

Pour l'arrêt de la demande de l'étage 2 (Figure B.3), on a le détecteur de la cabine en Etage 2, puis :

- Aussi on a la priorité d'appel en cabine (Appel E1 en cabine ou Appel RDC en cabine).
- Et on a l'appel de l'étage 1, parce que si la cabine est en RDC donc l'appel de l'étage 1 est prioritaire (dans le cas où l'appel de E2 et E1 sont appuyés en même temps).

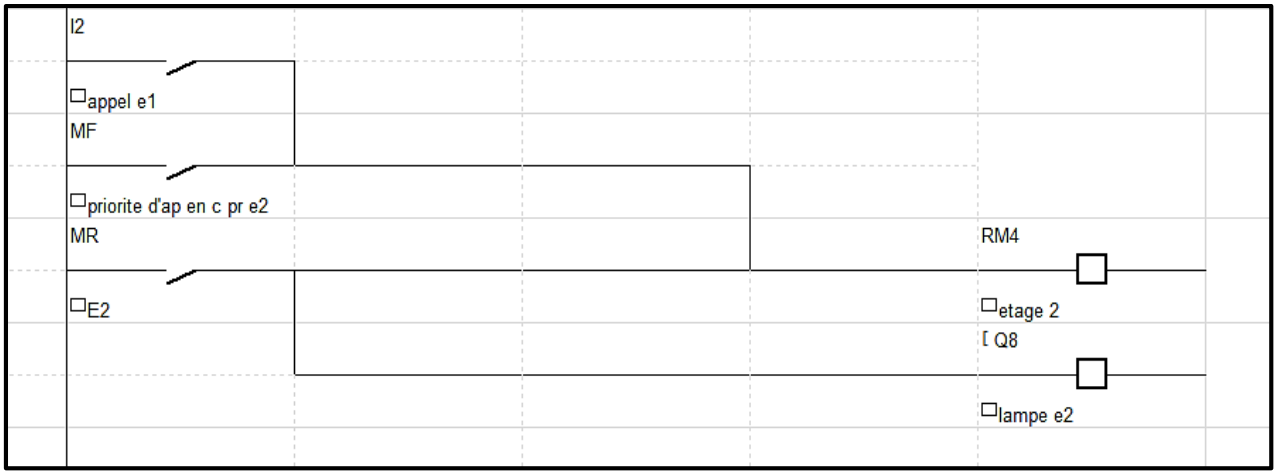

Figure B.3: Conditions d'arrêt pour E2

B.4 Mise sous tension

Le voyant de sous tension sera activé lorsque le programme est en cours (Figure B.4).

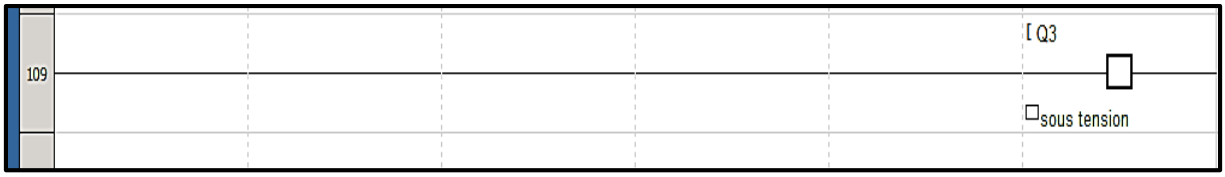

Figure B.4 : Sous tension

B.5 Arrêt d'urgence

Dans le cas où le passager à l'intérieur de l'ascenseur ressent un danger, il appuie sur le bouton d'arrêt d'urgence (Figure B.5).

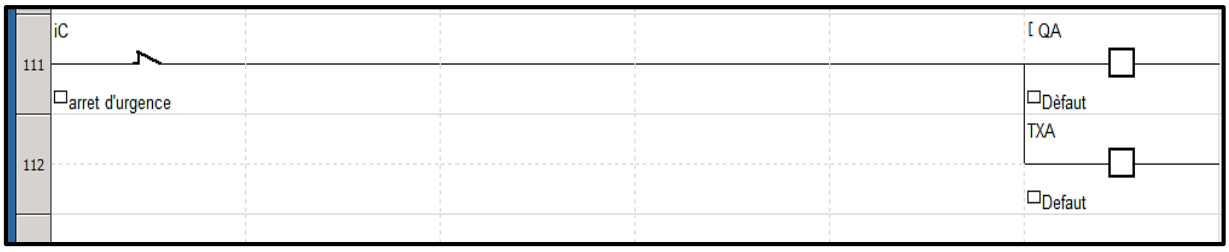

Figure B.5 : Arrêt d'urgence

B.6 Conditions initiales de démarrage

Pour le démarrage de la cabine, on a deux conditions initiales (Figure B.6) :

- La première c'est l'arrêt d'urgence.
- Et la deuxième est le capteur porte fermée.

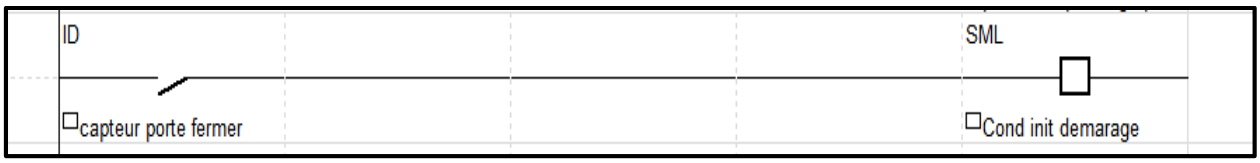

Figure B.6 : Conditions initiales de démarrage

B.7 Priorité d'appel en cabine

Lorsqu'on appuie sur l'un des bouton d'appels à l'extérieur (E1, E2, RDC), l'une des raisons de ne pas avoir la priorité est d'avoir aussi une demande de l'intérieur de la cabine. Pour cela on a :

- Priorité d'appel en cabine pour RDC (Figure B.7).

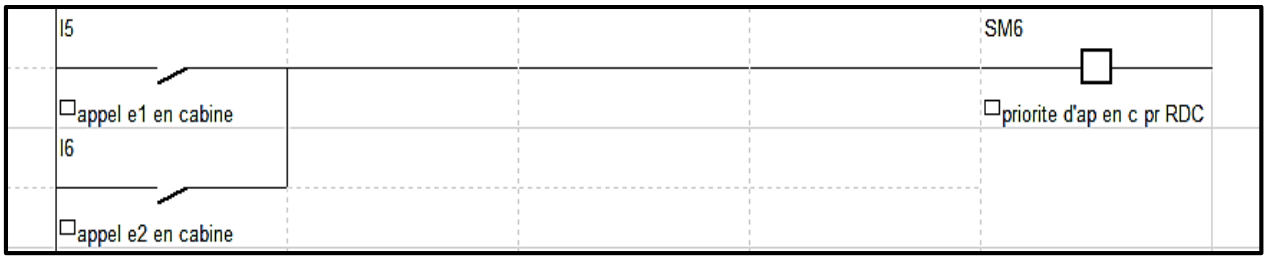

Figure B.7 : Priorité d'appel en cabine pour RDC

- Priorité d'appel en cabine pour l'étage 1 (Figure B.8).

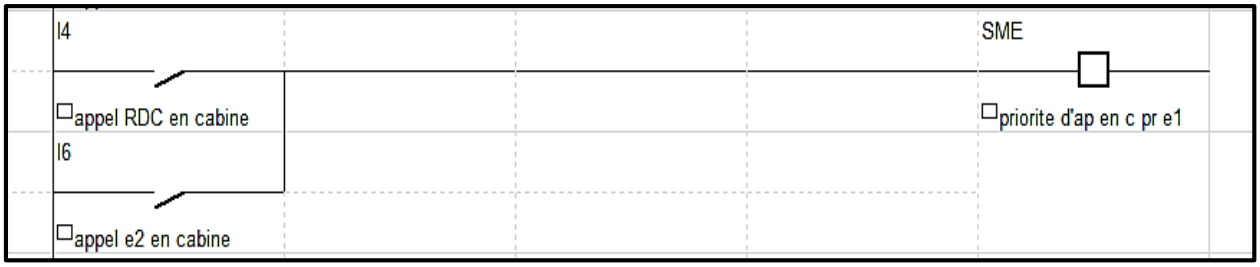

Figure B.8 : Priorité d'appel en cabine pour E1

- Priorité d'appel en cabine pour l'étage 2 (Figure B.9).

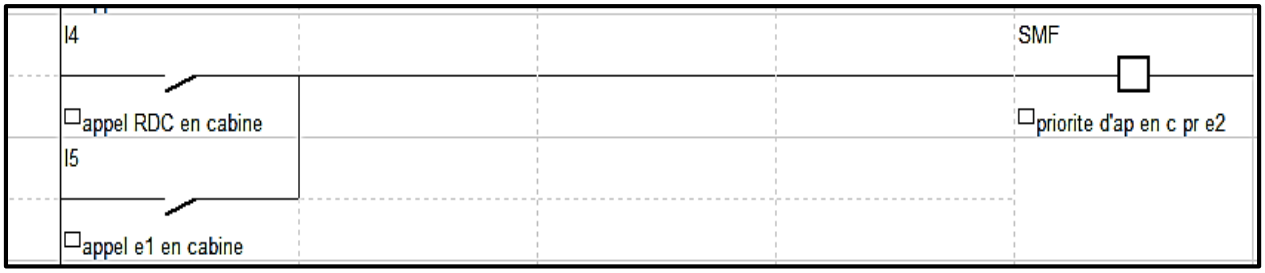

Figure B.9 : Priorité d'appel en cabine pour E2

B.8 Les blocs textes

Pour afficher les actions, nous avons utilisés les blocs textes définis sur la Figure B.10, pour mieux comprendre le fonctionnement de l'ascenseur.

Blocs textes		
01	X1	
02	X2	RDC
03	X3	Etage 1
04	X4	Etage 2
05	X5	La montée de la cabine
06	X6	La descente de la cabine
07	X7	
08	X8	fermetur de la porte
09	X9	ouverture de porte
10	XA	Defaut

Figure B.10 : Blocs textes