



Université d'Oran 2
Institut de Maintenance et de sécurité Industrielle

MEMOIRE / Master 2

Pour l'obtention du diplôme de Master 2
En Ingénierie de maintenance en instrumentation

**Réalisation D'une Barrière Automatique
A Base D'Arduino**

Présentée et soutenue publiquement par :

**YAHYA Achraf Abdelmadjid
MESSAOUDI Abdel ilah Said**

Devant le jury composé de :

Mr. Khalid Hachemi	Professeur	Université d'Oran 2	Président
Mr. Adda Neggaz Samir	M.A.A	Université d'Oran 2	Encadreur
Mr. Mekki Ibrahim El khalil	M.C.A	Université d'Oran 2	Examineur

Année universitaire 2022-2023

Barrière levante automatique

Résumé :

Les barrières sont modelées pour autorisées en toute sécurité le passage aux véhicules dans les parkings et les rues. Au fil du temps, le nombre de véhicules dans les villes urbaines ne cesse d'augmenter, créant une crise pour le flux de circulation et pour tous les systèmes manuels utilisées. Pour cela, les barrières automatiques sont apparues pour traiter ce problème de la gestion de la circulation et pour améliorer la sécurité dans les parkings aux utilisateurs. Le fonctionnement de ce genre de barrière peut être commandé par un bouton poussoir, une clé magnétique, une télécommande à distance ou avec reconnaissance des matricules. Elle peut être alimentée par plusieurs sources.

Mots clés : *Système automatisé, barrière levant, badge RFID, système de commande, ouverture et fermeture automatique.*

Automatic rising barrier

Abstract:

The barriers are shaped to safely allow vehicles to pass through parking lots and streets. Over time, the number of vehicles in urban cities continues to increase, creating a crisis for the flow of traffic and for any manual systems used. For this, automatic barriers have appeared to deal with this problem of traffic management and to improve safety in car parks for users. The operation of this type of barrier can be controlled by a push button, a magnetic key, a remote control or with recognition of registration numbers. It can be supplied by several sources.

Key words: *Automated system, rising barrier, RFID badge, control system, automatic opening and closing.*

حاجز صاعد أوتوماتيكي

المخلص:

يتم تشكيل الحواجز للسماح للمركبات بالمرور بأمان عبر مواقف السيارات والشوارع. بمرور الوقت، يستمر عدد المركبات في المدن في الازدياد، مما يخلق أزمة لتدفق حركة المرور وأي أنظمة يدوية مستخدمة. لهذا، ظهرت حواجز آلية للتعامل مع مشكلة إدارة حركة المرور هذه ولتحسين السلامة في مواقف السيارات للمستخدمين. يمكن التحكم في تشغيل هذا النوع من الحواجز بواسطة زر ضغط أو مفتاح مغناطيسي أو جهاز تحكم عن بعد أو مع التعرف على أرقام التسجيل. يمكن توفيره من قبل عدة مصادر.

كلمات مفتاحية: نظام الي، حاجز صاعد، إشارة RFID، نظام تحكم، فتح وإغلاق أوتوماتيكي.

Remerciements

Nous remercions Dieu, le tout puissant, de nous avoir accordé le courage, de nous donner la santé et la volonté d'entamer et de terminer ce mémoire.

Nous tenons à estimer nos remerciements avec une profonde reconnaissance et gratitude à notre encadreur de recherche, Mr ADDA NEGGAZ SAMIR pour ses précieux conseils, son soutien et sa générosité sans égale.

Notre plus vif remerciement aux membres de jury qui nous fait l'honneur d'examiner ce travail.

Nos remerciements vont également à tous les enseignants et les responsables de la département GI qui ont contribué à notre formation.

Sommaire

REMERCIEMENTS	i
SOMMAIRE	ii
LISTE DES FIGURES.....	iv
LISTE DES TABLEAUX.....	vii
LISTE DES ABREVIATIONS.....	viii
INTRODUCTION GÉNÉRALE.....	1
Chapitre 1: Les systèmes automatisés	3
1.1 Historique	3
1.2 Développement Du Système Automatisé	3
1.3 Définitions	5
1.3.1 Définition d'un système	5
1.3.2 Définition d'un système automatisé	5
1.4 Structure D'un Système Automatisé	5
1.4.1 La Partie Relation (en abrégé P.R)	6
1.4.2 La Partie Commande (en abrégé P.C)	7
1.4.3 La Partie Opérative (en abrégé P.O)	7
1.5 Les Interfaces Homme Machine (IHM)	8
1.5.1 Utilisateurs des IHM	9
1.5.2 Organes de dialogue	9
1.6 Chaîne Fonctionnelle	10
1.6.1 Chaîne d'information / La partie commande (PC)	10
1.6.2 Chaîne d'énergie / La partie opérative (PO)	11
1.7 Représentation Des Fonctions d'Automatisation	14
1.8 Objectifs d'Automatisation	14
1.9 Surveillance	15
1.10 Différentes Types De Commande	15
1.11 Conclusion	17
Chapitre 2: Les barrières levantes	18
2.1 Définition	18
2.1.1 Définition d'une barrière levante	18
2.1.2 Définition d'une barrière de parking automatique	18
2.2 Les Types Des Barrières	19
2.2.1 Barrière manuelle	20
2.2.2 Barrière automatique	21
2.3 Sources D'alimentation	29
2.4 La Structure De Barrière Levante	30
2.5 Mécanisme Des Barrières Levantes	32
2.5.1 Principe de Fonctionnement mécanique	32
2.5.2 Schéma bloc de la chaîne d'énergie	34

2.5.3	Le diagramme fonctionnel de barrière levante	35
2.6	Les Composants, Dispositifs De Sécurité Et Dispositifs De Commande	36
2.6.1	Les composants de la barrière levante	36
2.6.2	Dispositifs de sécurité	38
2.6.3	Dispositifs de commande	39
2.7	Conclusion	39
Chapitre 3: Arduino et périphériques		40
3.1	Présentation Général	40
3.2	Description De La Partie Matérielle	40
3.2.1	Définition d'une carte Arduino	41
3.2.2	Les différents types des cartes Arduino	41
3.2.3	Présentation de la carte Arduino MEGA 2560	44
3.2.4	Périphériques	47
3.3	Description De La Partie Logicielle	55
3.3.1	Logiciel IDE	55
3.3.2	Logiciel Proteus V8	60
3.3.3	Logiciel Micrografx Flowcharter 7.....	61
3.4	Conclusion	63
Chapitre 4: Simulation et Application.....		64
4.1	Différentes Composants Du Projet Réalisé	64
4.2	Organigramme Et Code Des Composants Avec l'Arduino Mega	65
4.2.1	Organigramme de gestion d'Entrée	65
4.2.2	Le code d'Entrée avec l'Arduino (composants d'Entrée)	67
4.2.3	Le fonctionnement de la gestion d'Entrée	69
4.2.4	Organigramme de gestion de Sortie	69
4.2.5	Le code de Sortie avec l'Arduino (composants de sortie)	71
4.2.6	Le fonctionnement de la gestion de Sortie	73
4.2.7	Organigramme de gestion de Stationnement	74
4.2.8	Le code de Stationnement avec l'Arduino (composants de stationnement)	76
4.2.9	Le fonctionnement de la gestion de Stationnement	77
4.3	Le Rôle Principale Des Composants Dans Notre Maquette	78
4.4	Schéma Fonctionnel Du Gestionnaire De Parking (PROTEUS)	79
4.5	Les Différentes Etapes De La Gestion De Parking	83
CONCLUSION GENERALE ET PERSPECTIVES		86
Bibliographie.....		87
Annexes.....		91

Liste des Figures

Figure 1.1 : Développement de store [22]	5
Figure 1.2 Structure d'un système automatisé	6
Figure 1.3 La Partie Relation (pupitre).....	6
Figure 1.4 La Partie Commande (automate programmable)	7
Figure 1.5 La Partie Opérative (actionneur, effecteur).....	7
Figure 1.6 Architecture générale de système automatisé	8
Figure 1.7 La relation homme machine [9]	9
Figure 1.8 Utilisation d'interface homme machine	10
Figure 1.9 Chaîne d'information d'un système automatisé [10]	11
Figure 1.10 Chaîne d'énergie d'un système automatisé [10]	12
Figure 1.11 Schéma d'un système automatisé [10].....	12
Figure 1.12 Chaîne fonctionnelle d'un système automatisé.....	13
Figure 1.13 Fonction globale d'un système automatisé	14
Figure 1.14 Système à logique combinatoire à plusieurs entrées (a) plusieurs sorties (b) une sortie	16
Figure 1.15 Système à logique séquentiel (a) asynchrone (b) synchrone	16
Figure 2.1 Système de Barrière Levante	19
Figure 2.2 Barrière Levante.....	19
Figure 2.3 Barrière Manuelle Levante (BML)	20
Figure 2.4 : Barrière Manuelle Pivotante	21
Figure 2.5 : Barrière Levante FB.....	22
Figure 2.6 Barrière Levante FBC	23
Figure 2.7 Barrière Levante FBX.....	24
Figure 2.8 Barrière Levante HB	25
Figure 2.9 Barrière Automatique TERRA Compact	26
Figure 2.10 Barrière Automatique TERRA ULTIMATE	27
Figure 2.11 Barrière à poutre coulissante (BC).....	28
Figure 2.12 Chaîne d'acquisition des données et Chaîne d'action [22]	30
Figure 2.13 Schéma pour le système de la barrière levante [23].....	31
Figure 2.14 Le mécanisme de la tête de barrière levante [23].....	32
Figure 2.15 Ressort.....	33
Figure 2.16 Schéma cinématique des mécanismes de barrière levante [24]	34

Figure 2.17 Schéma bloc de la chaîne d'énergie [24]	34
Figure 2.18 Le diagramme fonctionnel de barrière levante [22].....	35
Figure 2.19 Les composants de la barrière levante automatique à l'extérieur	36
Figure 2.20 Les composants du groupe mécanique la barrière à l'intérieur selon la première façade.....	37
Figure 2.21 Les composants du groupe mécanique la barrière à l'intérieur selon la deuxième façade [24]	38
Figure 3.1 La carte Arduino UNO.....	42
Figure 3.2 La carte Arduino MEGA.....	42
Figure 3.3 La carte Arduino NANO	43
Figure 3.4 La carte Arduino LEONARDO	43
Figure 3.5 La carte Arduino DUE	44
Figure 3.6 Présentation de la carte Arduino MEGA	45
Figure 3.7 Les composants de la carte Arduino MEGA	45
Figure 3.8 Le microcontrôleur ATmega2560	47
Figure 3.9 Servomoteur	48
Figure 3.10 Servomoteur SG90.....	48
Figure 3.11 Capteur de proximité Infrarouge – FC 51	50
Figure 3.12 Ecran LCD	51
Figure 3.13 Bouton Poussoir 4 Pattes.....	52
Figure 3.14 Buzzer actif	53
Figure 3.15 LED diffusante 5mm.....	54
Figure 3.16 Ecran principal au lancement de l'IDE.....	55
Figure 3.17 L'interface de l'IDE Arduino.....	56
Figure 3.18 Les fonctions d'un programme Arduino dans l'IDE	57
Figure 3.19 Les parties d'un programme Arduino [54]	58
Figure 3.20 Paramétrage de la carte étape 1	59
Figure 3.21 Paramétrage de la carte étape 2.....	59
Figure 3.22 L'écran principal au lancement de Proteus [56]	60
Figure 3.23 Interface principale de FlowCharter 7.....	62
Figure 4.1 Organigramme de déclaration et initialisation des composants(entrée) ...	66
Figure 4.2 Organigramme fonctionnel d'entrée du parking	67
Figure 4.3 Organigramme de déclaration et initialisation des composants(sortie)	70
Figure 4.4 Organigramme fonctionnel de sortie du parking	71
Figure 4.5 Organigramme de déclaration et initialisation des composants.....	74
Figure 4.6 Organigramme fonctionnel de stationnement(couvert)	75

Figure 4.7 Schéma fonctionnel Proteus de parking.....	80
Figure 4.8 Schéma fonctionnel Proteus de la gestion d'entrée et sortie.....	81
Figure 4.9 Schéma fonctionnel Proteus de la gestion de stationnement	82
Figure 4.10 Images pour notre maquette	85

Liste des Tableaux

Tableau 2.1 Caractéristiques des Barrières manuelles BML et BMP [19] [20]	21
Tableau 2.2 Caractéristiques de Barrière Levante FB [21]	22
Tableau 2.3 Caractéristiques des Barrières Levantes FBC, FBX et HB [21].....	26
Tableau 2.4 Caractéristiques des Barrières Automatiques TERRA Compact et TERRA ULTIMATE [21]	28
Tableau 2.5 Caractéristiques de Barrière à poutre coulissante (BC) [21]	29
Tableau 3.1 Caractéristiques de la carte Arduino MEGA 2560 [33]	46
Tableau 3-2 Caractéristiques du servomoteur SG90 [37]	49
Tableau 3.3 Caractéristiques du Capteur de proximité Infrarouge – FC 51 [43]	50
Tableau 3.4 Caractéristiques de l’Ecran LCD	51
Tableau 3.5 Caractéristiques du Buzzer actif [51]	53
Tableau 3.6 Caractéristiques du LED 5mm [52].....	54
Tableau 4-1 Rôle des composants d’entrée	78
Tableau 4-2 Rôle des composants de stationnement	78
Tableau 4-3 Rôle des composants de sortie.....	79

Liste des Abréviations

IA : Intelligence Artificielle

IdO : Internet des objets

SAP : Système Automatisés de Production

P.R : Partie Relation

P.C : Partie Commande

P.O : Partie Opérative

IHM : Interface homme machine

BML : Barrière manuelle levant

BMP : Barrière manuelle pivotante

HB : Barrière hydraulique

BC : Barrière coulissante

HID: Human Interface Device

SRAM: Static Random-Access Memory

EEPROM: Electrically Erasable Programmable Read-Only Memory

UART: Universal Asynchronous Receiver-Transmitter

I2C: Inter-Integrated Circuit

SPI: Serial Peripheral Interface

USB: Universal Serial Bus

CAN : Convertisseurs analogique-numérique

CC : Courant Continu

IDE: Integrated Development Environment

RISC: Reduced Instruction Set Computer

LCD : Liquid Crystal Display (En français : écran à cristaux liquides)

PWM : Pulse-Width Modulation (modulation de largeur d'impulsion)

RFID : Radio Frenquency Identification (identification par radiofréquence)

CAO : Conception assistée par ordinateur

AVR : Advanced Virtual RISC

Introduction Générale

Le nombre de véhicules augmente chaque jour dans tout le monde. Cela cause le problème d'embouteillage. Le parking parmi les facteurs la plus important pour régler cette situation. Le parking de voiture est à base de barrières. Il est un sujet qui a toujours été très présent car il touche la mobilité de la plupart des personnes. Les conducteurs doivent toujours se stationner pour se rendre à leurs lieux de destination. Cette facilité de circulation a été obtenue à l'aide des barrières.

Les barrières sont des éléments physiques qui sont utilisés pour délimiter, protéger ou restreindre l'accès à certaines zones ou à certains objets. Elles sont utilisées dans une variété de contextes, que ce soit pour des raisons de sécurité, de contrôle d'accès ou de gestion des flux de personnes.

Les barrières peuvent être équipées de divers mécanismes pour faciliter leur ouverture et leur fermeture. Certaines barrières sont également dotées de dispositifs de sécurité supplémentaires, comme des capteurs ou des caméras de surveillance, pour renforcer la protection des zones qu'elles délimitent. Dans le domaine de la sécurité, les barrières sont utilisées pour contrôler et limiter l'accès aux zones sensibles ou les installations industrielles. Elles jouent un rôle essentiel dans la prévention des intrusions non autorisées et la protection des personnes et des biens.

Avec l'apparition des systèmes automatisés, toutes les méthodes manuelles qui ont été utilisés dans les différents domaines et sites ont également évolué pour s'adapter à ces nouvelles technologies. L'un de ces systèmes est la barrière qui fait une transition remarquable pour être une barrière automatique avec un système automatisé avancé.

Ce genre des barrières automatiques sont généralement composées d'un bras métallique horizontal qui bloque l'entrée ou la sortie d'un véhicule. Le bras peut être soulevé ou abaissé automatiquement en fonction des instructions reçues. Ces barrières automatiques offrent de nombreux avantages, tels que la sécurité en limitant l'accès aux seuls utilisateurs autorisés, la collecte de données sur les entrées et sorties.

Les barrières automatiques sont souvent intégrées à un système de gestion de parking ou à un système de contrôle d'accès plus large. Elles peuvent être activées par différents moyens, tels

que des cartes d'accès, des capteurs de proximité ou des systèmes de reconnaissance de plaques d'immatriculation...

Lorsqu'un véhicule s'approche de la barrière, il doit s'identifier ou présenter un moyen d'accès valide pour que la barrière s'ouvre. Une fois que l'accès est autorisé, la barrière s'ouvre automatiquement, permettant au véhicule de passer. Après le passage du véhicule, la barrière se referme automatiquement pour empêcher l'accès non autorisé.

Dans cette situation, l'objectif de notre travail est de réaliser une maquette d'une barrière levant automatique à base d'Arduino (matériaux électroniques et logiciels) et tester la commande de cette barrière à la gestion d'entrée et sortie de véhicules. Dans notre réalisation, nous avons choisi une carte Arduino Mega 2560 car elle contient plus des portes entrées/sorties, et la bonne gestion pour piloter un système un peu compliqué.

Ce mémoire est structuré en 4 chapitres à travers lesquels nous décrivons le travail effectué pour la réalisation de notre système : Le chapitre 1 cherche à une idée générale sur les systèmes automatisés, la structure de leurs fonctionnements et l'objectif de l'automatisation. En addition, nous arrivons à donner les différents types de commande et savoir la surveillance d'un système automatisé.

Le chapitre 2 vise à identifier les barrières levantes manuelles et automatiques, puis nous avons donné la structure de fonctionnement d'une barrière automatique avec tous les mécanismes qui faciliter leur gestion.

Le chapitre 3 est consacré pour la description des deux importantes parties de projet par l'étude d'Arduino :

Partie matériel utilisé : nous arrivons à savoir les différentes cartes d'Arduino et donner une présentation générale pour la carte choisie (Arduino Mega 2560). Après, nous étudions la conception de chaque composant électronique utilisé.

Partie logiciel utilisé, nous travaillons à voir une présentation sur le logiciel utilisé et montrer la structure de programmation utilisé et comment injecter le sur la carte.

Le chapitre 4 vise à donner une simulation pour notre application par présenter tous les organigrammes et les programmes associés au fonctionnement de notre système. Après, nous donnons les étapes de réalisation et fonctionnement de notre application.

Enfin, une conclusion est présentée avec quelques perspectives.

Chapitre 1: Les systèmes automatisés

INTRODUCTION :

En général un système est un ensemble d'éléments en interaction dynamique, organisé en fonction d'un but. La fonction globale de tout système est de conférer une valeur ajoutée à un ensemble de matières d'œuvre dans un contexte donnée. Un système de production est dit industriel si l'obtention de cette valeur ajoutée a un caractère reproductible et peut être exprimée et quantifiée en terme économique. L'objectif de ce chapitre est de présenter les fonctions globales d'un système, automatisation et structure des systèmes automatisés, Pré-actionneurs (contacteurs, Triac, ...), actionneurs (vérins, moteurs, ...), capteurs, classification des systèmes automatisés, spécification des niveaux du cahier des charges, outils de représentation des spécifications fonctionnelles. [1]

1.1 HISTORIQUE :

- Les précurseurs tels que balise pascal (1623-1662) qui invente la première calculette "la Pascaline".
- Origine de l'invention de l'ordinateur. Principe de la carte perforée, utilisé jusqu'à la fin des années 1970.
- Bardeen, brattain and shockley inventent le transistor en 1949.
- Moore invente le microprocesseur et crée Intel en 1966.
- Grenelle (1972) invente le premier micro-ordinateur.
- 1974 premiers automates programmables industriels. [2]

1.2 DEVELOPPEMENT DU SYSTEME AUTOMATISE :

Le développement des systèmes automatisés a connu une évolution significative jusqu'à aujourd'hui. Voici une vue d'ensemble des principales étapes et avancées :

1. **Révolution industrielle** : La première phase du développement des systèmes automatisés remonte à la révolution industrielle, qui a débuté au 18e siècle. L'utilisation

de machines mécaniques a permis de remplacer le travail manuel dans de nombreux processus de fabrication.

2. **Automatisation programmable** : Dans les années 1940, les premiers systèmes automatisés programmables sont apparus. Ces systèmes utilisaient des relais électromécaniques et des circuits logiques pour contrôler les machines. Ils étaient limités en termes de complexité et de flexibilité.
3. **Contrôle numérique** : L'avènement de l'électronique et de l'informatique a ouvert la voie au développement des systèmes automatisés numériques. Dans les années 1950 et 1960, les premiers contrôleurs numériques ont été introduits, utilisant des calculateurs pour gérer les processus industriels.
4. **Automatisation industrielle** : Dans les années 1970 et 1980, l'automatisation industrielle s'est généralisée. Les systèmes automatisés ont été utilisés dans divers domaines tels que l'automobile, l'industrie chimique, l'agroalimentaire, etc. Les robots industriels ont également fait leur apparition et ont été intégrés aux lignes de production.
5. **Systèmes de contrôle en temps réel** : Les années 1990 ont vu l'émergence de systèmes de contrôle en temps réel, capables de prendre des décisions instantanées en fonction des données de capteurs en temps réel. Ces systèmes ont permis d'améliorer l'efficacité, la précision et la sécurité des processus automatisés.
6. **Intelligence artificielle et apprentissage automatique** : Ces dernières années, l'intégration de l'intelligence artificielle (IA) et de l'apprentissage automatique (machine learning) a considérablement enrichi les capacités des systèmes automatisés. Les algorithmes d'IA permettent aux machines de prendre des décisions plus complexes, d'apprendre à partir de données et de s'adapter à des situations changeantes.
7. **Automatisation des services** : En parallèle de l'automatisation industrielle, les systèmes automatisés se sont également développés dans le secteur des services. Par exemple, les chatbots et les assistants virtuels utilisent l'IA pour fournir un support client automatisé, tandis que les systèmes de réservation en ligne automatisent les processus de réservation dans l'industrie du voyage et de l'hôtellerie.
8. **Internet des objets (IdO)** : L'Internet des objets a également joué un rôle important dans le développement des systèmes automatisés. Les objets connectés et les capteurs sont de plus en plus utilisés pour surveiller et contrôler les processus en temps réel, ce qui permet une automatisation plus étendue et précise.

- 9. Automatisation intelligente :** Aujourd'hui, l'automatisation intelligente est devenue une réalité. Les technologies telles que la robotique avancée, la vision par ordinateur, le traitement du langage naturel et la robotique collaborative. [3]

Exemple :

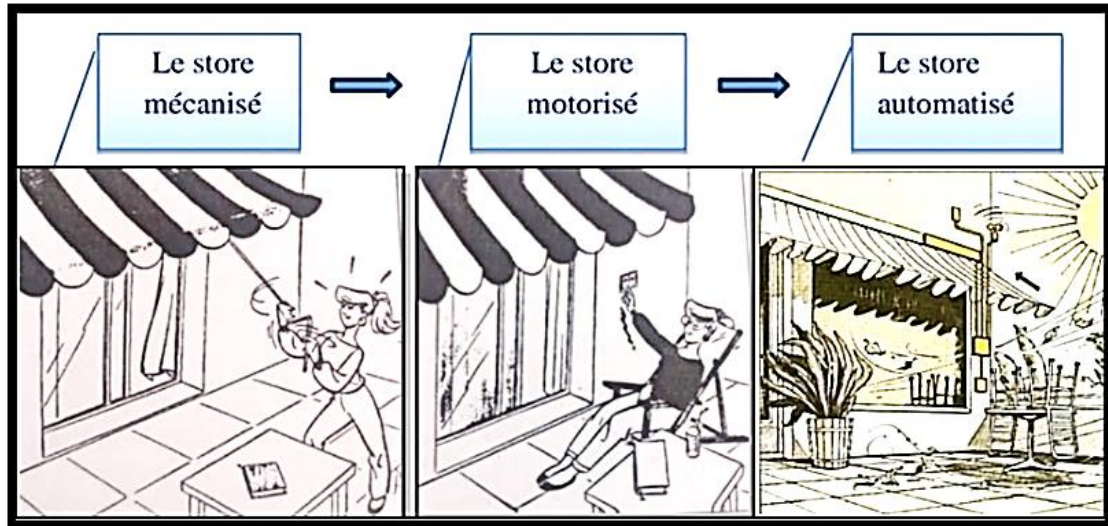


Figure 1.1 : Développement de store [22]

1.3 DEFINITIONS :

1.3.1 Définition d'un système :

Ensemble fermé d'éléments ou de relations, soustraits à toute autre influence que celle qu'ils ont les uns sur les autres. Il est possible de constituer un système clos où on isolera, pour en constater et en faire varier les effets, l'action d'une cause déterminée. [4]

1.3.2 Définition d'un système automatisé :

Un système automatisé est composé de plusieurs éléments qui exécutent un ensemble de tâches programmées sans que l'intervention de l'homme ne soit nécessaire. Exemples : le passage à niveau automatique, la porte de garage, etc... [5]

1.4 STRUCTURE D'UN SYSTEME AUTOMATISE :

Tous les Système Automatisés de Production (SAP) possèdent une structure qui se présente sous la forme suivante :

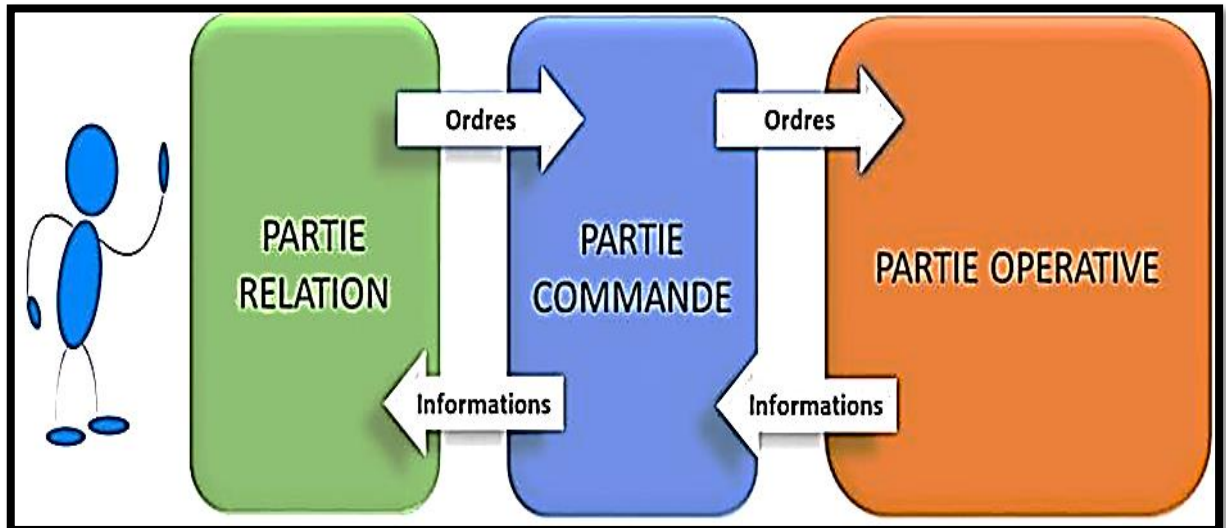


Figure 1.2 Structure d'un système automatisé

1.4.1 La Partie Relation (en abrégé P.R) :

C'est la partie qui permet le dialogue entre l'homme et la machine, elle regroupe les capteurs opérateurs et les composants de signalisation visuels et / ou sonores. Le pupitre de commande sert de support aux éléments de la P. R. [6]



Figure 1.3 La Partie Relation (pupitre)

1.4.2 La Partie Commande (en abrégé P.C) :

C'est la partie qui permet de gérer, d'organiser l'enchaînement des actions, des mouvements du système. Elle regroupe les constituants et les composants destinés au traitement des informations (signaux) émises par les capteurs machines de la P. O. et les capteurs opérateurs de la P. R. (C'est le cerveau du système). [6]



Figure 1.4 La Partie Commande (automate programmable)

1.4.3 La Partie Opérative (en abrégé P.O) :

C'est la partie qui permet de réaliser les mouvements, les actions sur le produit. Elle regroupe les effecteurs, les actionneurs et leurs pré-actionneurs ainsi que les capteurs machines nécessaires au contrôle du déplacement des actionneurs et au contrôle de présence des objets ou des personnes.[6]

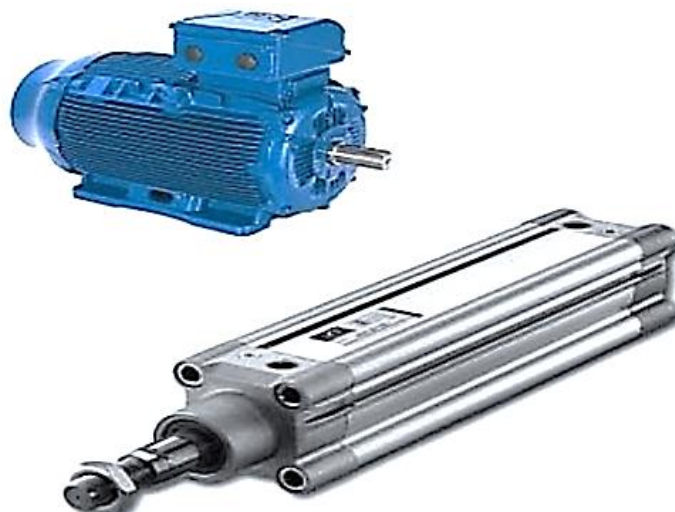


Figure 1.5 La Partie Opérative (actionneur, effecteur)

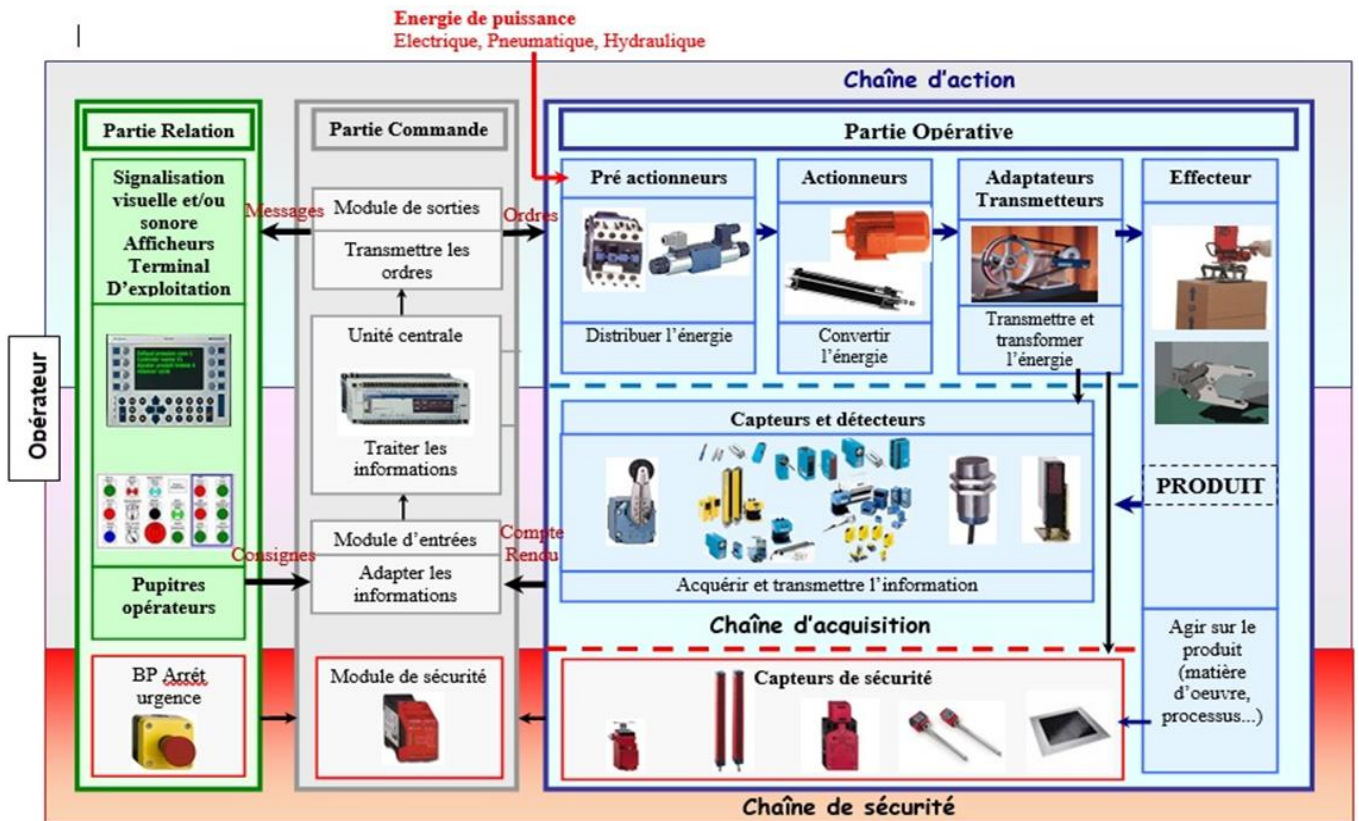


Figure 1.6 Architecture générale de système automatisé

1.5 LES INTERFACES HOMME MACHINE (IHM) :

Une interface homme-machine (IHM) est l'ensemble des dispositifs matériels et logiciels qui permettent à un utilisateur humain d'interagir avec une machine.

L'IHM renvoie des informations de façon visuelle à l'utilisateur pour qu'il supervise par exemple le bon déroulement d'une tâche. Elle peut également permettre d'envoyer des informations à l'unité de traitement pour déclencher une action précise. [8]

Les IHM peuvent prendre différentes formes. Il peut s'agir d'écrans directement intégrés aux machines, d'écrans d'ordinateur, de tablettes tactiles, et bien plus encore.

Quelle que soit la forme, le but de ces interfaces reste toutefois le même : fournir des insights sur les performances et les progrès des machines. [7]

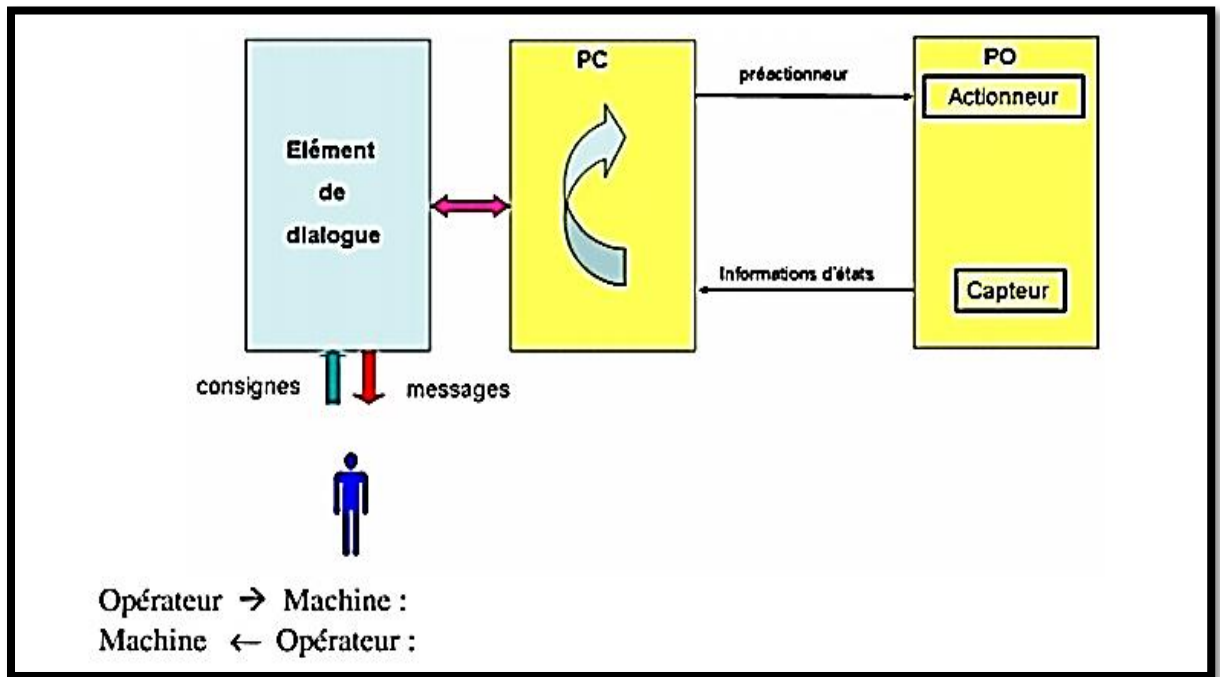


Figure 1.7 La relation homme machine [9]

1.5.1 Utilisateurs des IHM :

Les personnes chargées d'utiliser les IHM sont principalement, les opérateurs, les intégrateurs systèmes, et les ingénieurs (plus particulièrement les ingénieurs de système de contrôle). Ils utilisent ces interfaces comme des ressources essentielles pour surveiller les processus, diagnostiquer les problèmes, et visualiser les données. [7]

1.5.2 Organes de dialogue :

L'interface de communication entre l'homme et la machine peut prendre des formes très variées suivant :

- La complexité des tâches décisionnelles propre à l'opérateur, souvent liées à la complexité du système à piloter.
- Le degré d'intégration de l'opérateur dans la conduite du procédé.
- Le niveau de dialogue.
- Le type d'information.

Dans la pratique, suivant le degré de complexité du dialogue, on trouve trois techniques dans la conception de l'organe de dialogue

- Le pupitre simple à boutons et voyants.
- Le pupitre avec terminal de dialogue.

- Le superviseur. [9]



Figure 1.8 Utilisation d'interface homme machine

1.6 CHAÎNE FONCTIONNELLE :

Un système automatisé est composé de plusieurs éléments qui exécutent un ensemble de tâches programmées sans que l'intervention de l'homme ne soit nécessaire. Exemples : le passage à niveau automatique, la porte de garage, etc... [10]

Il est composé de :

1.6.1 Chaîne d'information / La partie commande (PC) :

Elle donne les ordres et reçoit les informations de l'extérieur ou de la partie opérative. Elle peut se présenter sous 3 manières différentes : un boîtier de commande, un microprocesseur (cerveau électronique), ou un ordinateur.

Acquérir : Fonction qui permet de prélever des informations à l'aide de capteurs.

Traiter : C'est la partie commande composée d'un automate ou d'un microcontrôleur.

Communiquer : Cette fonction assure l'interface l'utilisateur et/ou d'autres systèmes.

Transmettre : Cette fonction assure l'interface avec l'environnement de la partie commande.

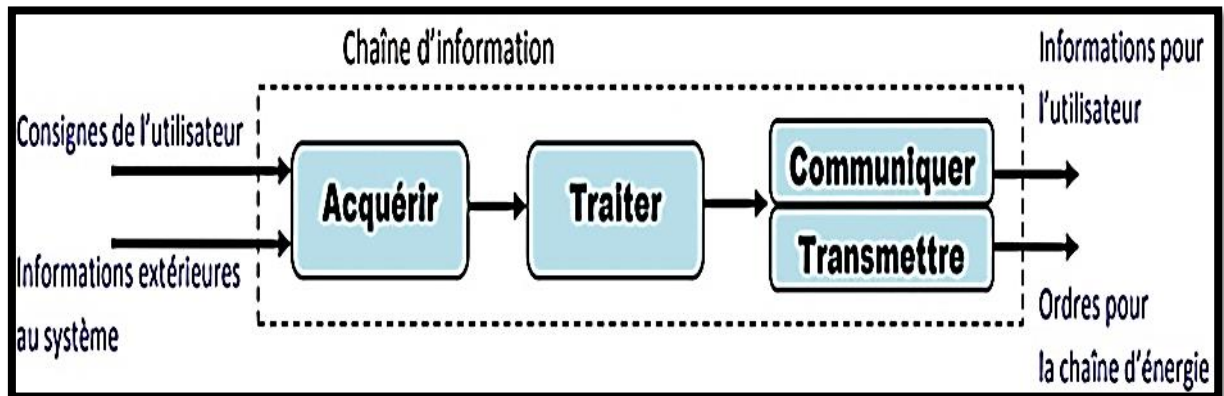


Figure 1.9 Chaîne d'information d'un système automatisé [10]

1.6.2 Chaîne d'énergie / La partie opérative (PO) :

C'est la partie d'un système automatisé qui effectue le travail. Autrement dit, c'est la machine. C'est la partie qui reçoit les ordres de la partie commande et qui les exécute. Elle comporte les capteurs et les actionneurs :

- **Un actionneur** est un élément de la partie opérative qui est capable de produire une action physique tel qu'un déplacement, un dégagement de chaleur, une émission de lumière ou de son à partir de l'énergie qu'il a reçu.
- **Un capteur** est un élément de la partie opérative qui permet de recueillir des informations et de les transmettre à la partie commande. Les capteurs sont choisis en fonction des informations qui doivent être recueillies. [10]

Cette partie fonctionne selon 4 essentiels éléments :

Alimenter : Mise en forme de l'énergie externe en énergie compatible pour créer une action.

Distribuer : Distribution de l'énergie à l'actionneur réalisée par un distributeur ou un contacteur.

Convertir : L'organe de conversion d'énergie appelé actionneur peut être un vérin, un moteur...

Transmettre : Cette fonction est remplie par l'ensemble des organes mécaniques de transmission de mouvement et d'effort : engrenages, courroies, accouplement, embrayage....

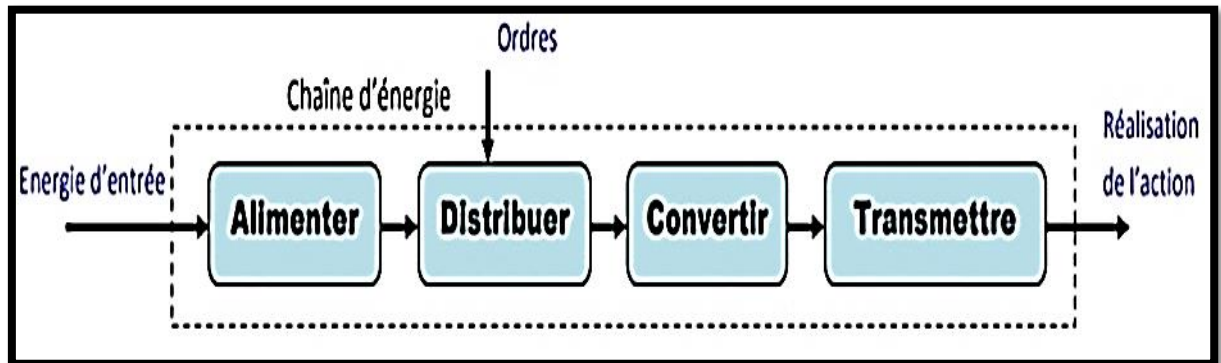


Figure 1.10 Chaîne d'énergie d'un système automatisé [10]

La partie opérative d'un système automatisé peut être décrite en une ou plusieurs chaînes fonctionnelles, ces dernières comportant une chaîne d'action et une chaîne d'acquisition.

- **Une chaîne d'action** est un ensemble organisé de composants dont le rôle est de convertir un ordre émis par la partie commande en effet sur la matière d'œuvre.
- **Une chaîne d'acquisition** est un ensemble organisé de composants dont le rôle est de prélever des grandeurs physiques sur la partie opérative ou sur l'environnement et de les convertir en signaux interprétables par la partie commande. [10]

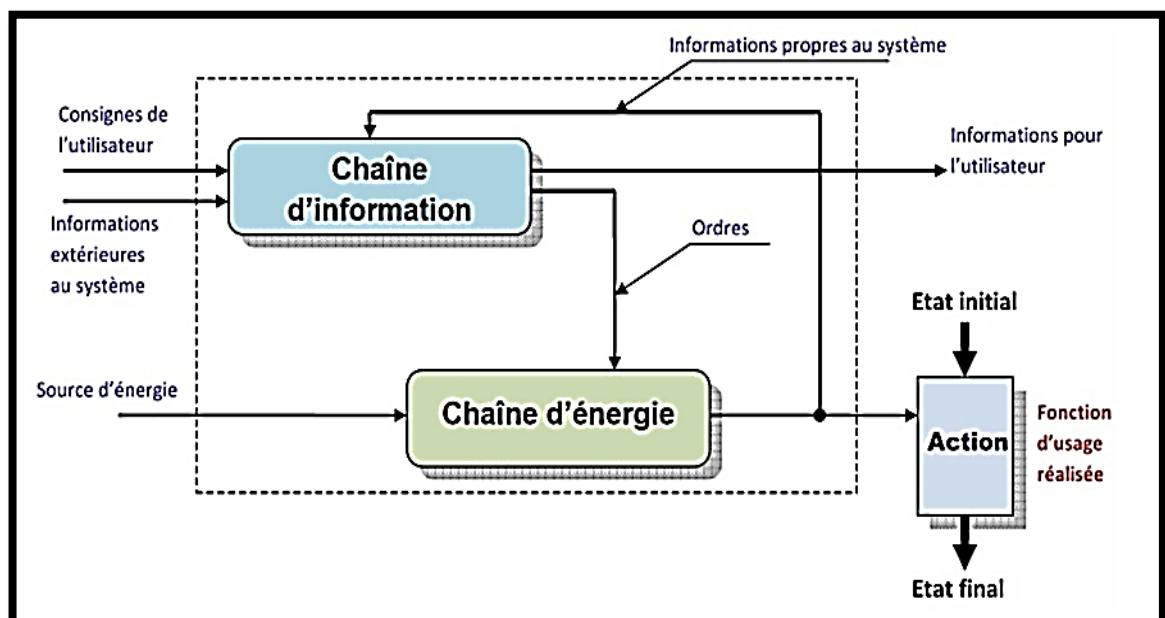


Figure 1.11 Schéma d'un système automatisé [10]

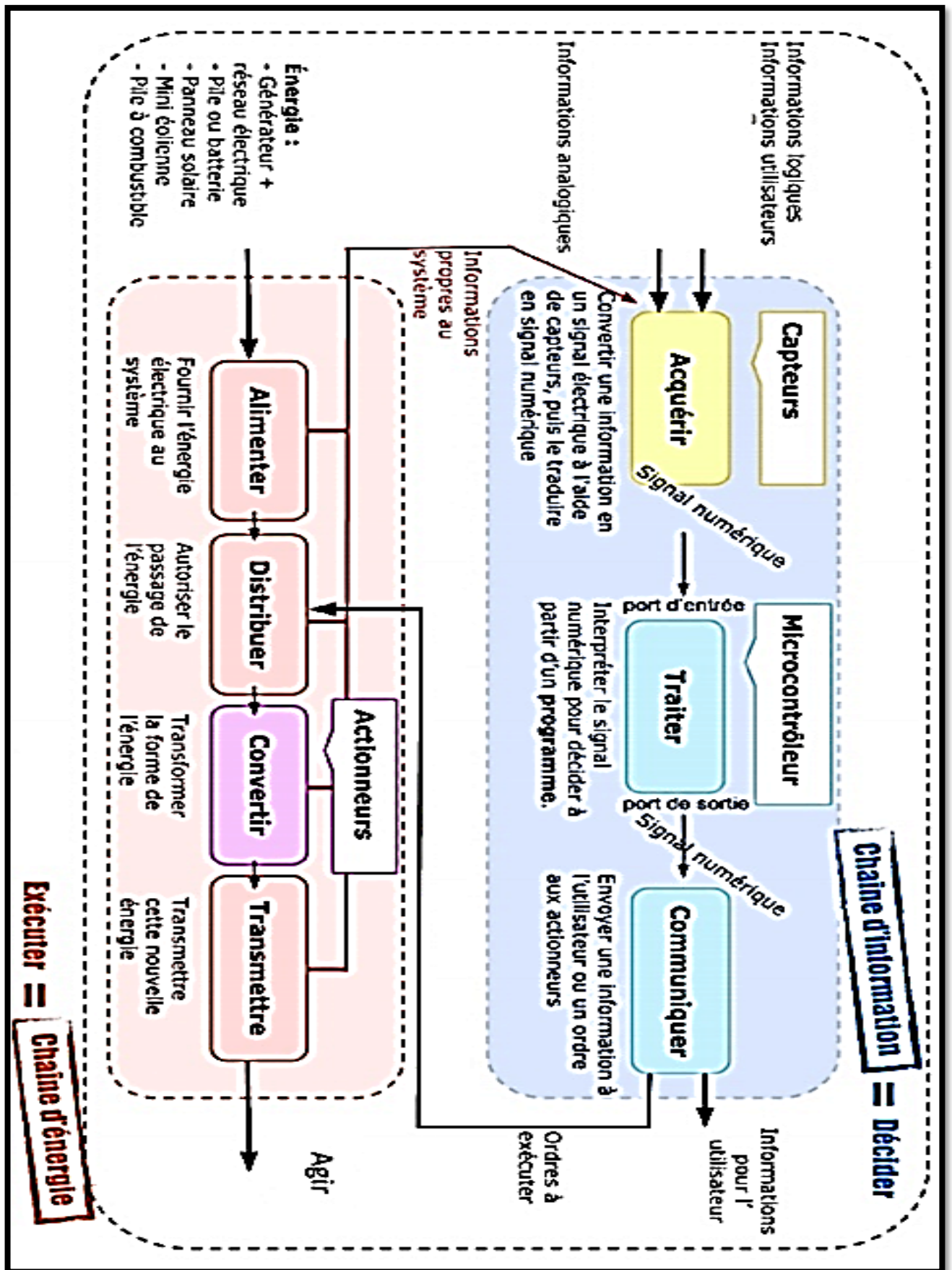


Figure 1.12 Chaîne fonctionnelle d'un système automatisé

1.7 REPRESENTATION DES FONCTIONS D'AUTOMATISATION :

Un système automatisé est représenté par une boîte appelée ACTIGRAMME dans laquelle on inscrit sa FONCTION GLOBALE du point de vue de l'utilisateur

Les données de contrôle sont réparties de la façon suivante :

- **W** : Données de contrôle d'énergie – Ex : énergie électrique, pneumatique, hydraulique, chimique...
- **C** : Données de contrôle de configuration – Ex : programmation d'un API, d'un programmeur mécanique...
- **R** : Données de contrôle de réglage – Ex : réglages de vitesse, courses, paramètres de fonctionnement...
- **E** : Données de contrôle d'exploitation – Ex : dialogue homme machine, ordre opérateur...

La fonction globale de tout système est de donner une valeur ajoutée, à un ensemble de matières d'œuvre dans un contexte donné. [11]

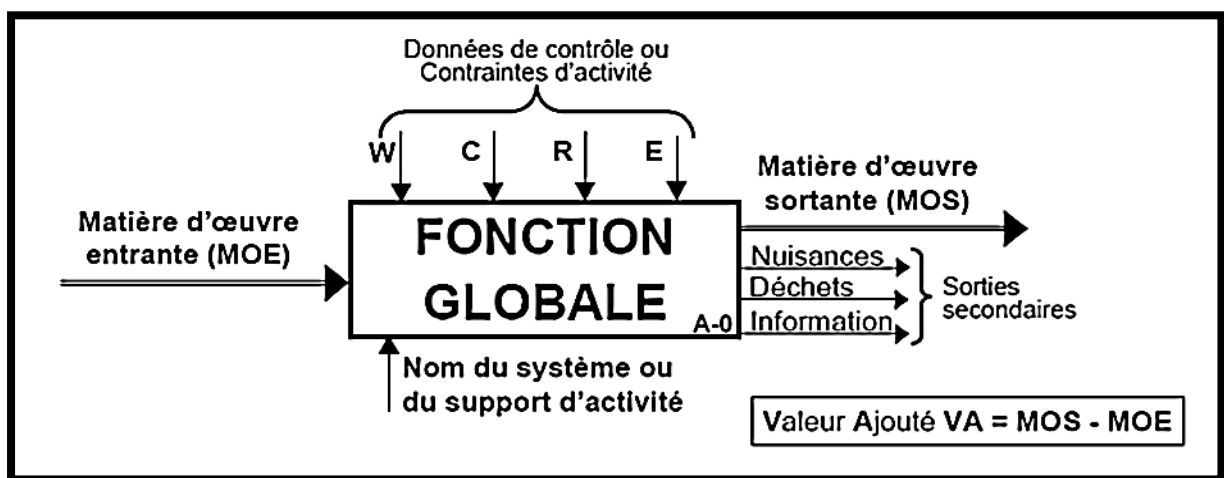


Figure 1.13 Fonction globale d'un système automatisé

1.8 OBJECTIFS D'AUTOMATISATION :

L'objectif de l'automatisation est de créer des outils capables d'effectuer ce que font des êtres humains, en mieux. Par exemple, un humain peut effectuer n'importe quelle tâche au moins une fois. Mais lorsque la tâche nécessite des exécutions répétitives, en particulier sur de longues périodes, un système automatisé est mieux équipé pour effectuer ce travail avec des résultats plus prévisibles et sans erreur.

L'automatisation peut aussi être conçue pour améliorer la vitesse. Lorsque vous perfectionnez ces objectifs d'automatisation, vous pouvez créer des systèmes plus rapides, répétitifs et pouvant être exécutés quotidiennement. [12]

1.9 SURVEILLANCE :

La surveillance des systèmes, en vue d'assurer un bon fonctionnement, consiste à suivre et à piloter un processus automatisé. Nous nous intéressons à la localisation et à l'élimination de toutes défaillances (fautes) dans le système surveillé. Un dysfonctionnement peut apparaître sur les capteurs, les actionneurs ou le système de contrôle. Les causes des anomalies ou des dysfonctionnements sont multiples, elles peuvent être provoquées par une panne d'un composant, par des coupures de l'alimentation électrique, par des perturbations extérieures.

Lors de la surveillance d'un système, nous devons assurer les tâches suivantes :

- Détection et isolation des fautes.
- Diagnostic des causes et de l'influence de la faute sur le composant concerné et sur le système entier.
- Reconfiguration et retour au fonctionnement normal du système. [13]

Il y a trois méthodes de surveillance :

- Méthode de surveillance par modélisation fonctionnelle et matérielle.
- Méthode de surveillance par modélisation physique.
- Méthode de surveillance par analyse des signatures externes. [13]

1.10 DIFFERENTES TYPES DE COMMANDE :

On a deux types des systèmes pour commander :

- Système à logique combinatoire.
- Système à logique séquentiel.

Système à logique combinatoire :

À une combinaison des entrées correspondent une seule combinaison des sorties (logique combinatoire). Ces systèmes n'utilisent aucun mécanisme de mémorisation ; ils n'ont pas de

mémoire. Les outils utilisés pour les concevoir sont l'algèbre de Boole, les tables de vérité, les tableaux de Karnaugh. [14]

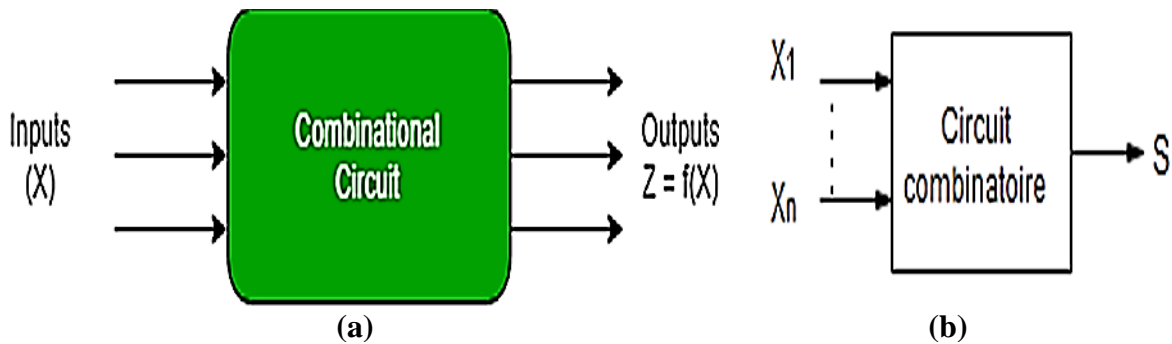


Figure 1.14 Système à logique combinatoire à plusieurs entrées (a) plusieurs sorties (b) une sortie

Système à logique séquentiel :

Le déroulement s'effectue étape par étape, séquence par séquence. Pour ces systèmes, à une situation des entrées peuvent correspondre plusieurs situations de sortie. La sélection d'une sortie ou d'une autre dépend de la situation antérieure du dispositif (étape précédente). Les mécanismes de mémorisation, ou mémoires, sont à la base de la logique séquentielle. [14]

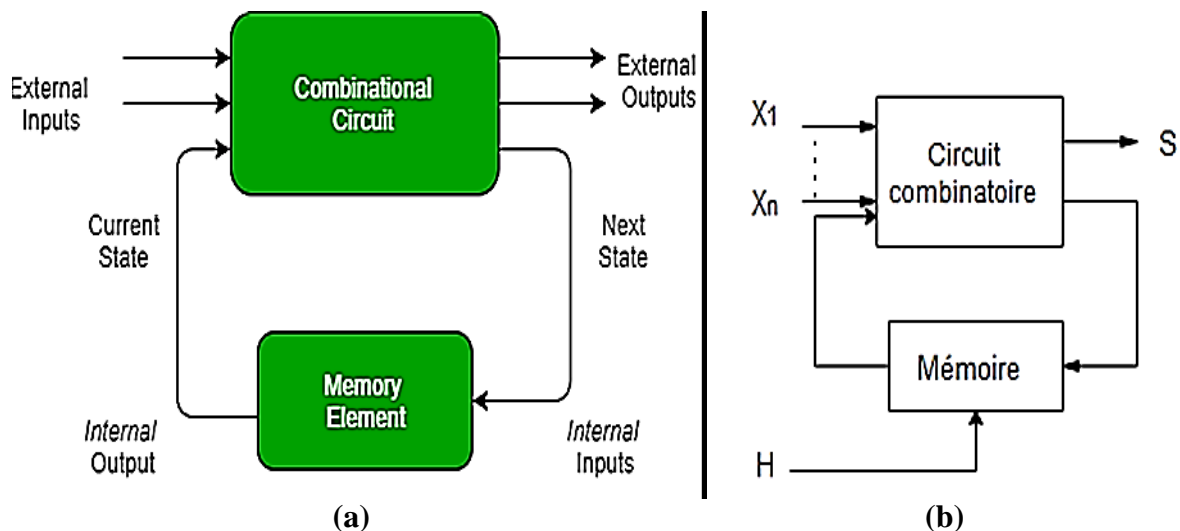


Figure 1.15 Système à logique séquentiel (a) asynchrone (b) synchrone

1.11 CONCLUSION :

Dans ce chapitre, on a vu les systèmes automatisés avec leurs structure et fonctions, qui sont souvent utilisée pour exécuter des tâches répétitives avec des équipements dédiés afin d'améliorer l'efficacité de la production et la cadence. Donc les systèmes automatisés renforcent la productivité, la fiabilité, la disponibilité et les performances.

Chapitre 2: Les barrières levantes

INTRODUCTION :

La Barrière Automatique offre une large gamme de barrières automatiques levantes répondant aux exigences techniques, de robustesse et économiques des exploitants de péage et de parcs de stationnement tous comme les sites industriels ou établissements recevant du public. [15]

La barrière automatique est dotée d'un mécanisme d'ouverture et fermeture automatique qu'on peut installer facilement dans différents sites. Elle permet ainsi de laisser rapidement le passage aux véhicules autorisés et contrôler les intrusions ou les véhicules suspects. L'installation de ce type de barrière est aussi rapide et facile. Sa technologie optimale facilite son utilisation. [16]

2.1 DEFINITION :

2.1.1 Définition d'une barrière levante :

Les barrières levantes électriques sont associées à des systèmes de contrôle d'accès permettant de déterminer de quelle manière l'ordre d'ouverture ou de fermeture va être envoyé à la barrière.

Lorsqu'un véhicule désire entrer, l'ouverture des barrières se produit lorsque l'utilisateur présente son contrôle d'accès : cela peut être un badge, un système de détection automatique par boucle magnétique, par lecture de plaque, par carte, ...

Lorsqu'un véhicule désire sortir de la même façon, l'envoi de l'ordre de la commande peut être avec ces mêmes possibilités. [15]

2.1.2 Définition d'une barrière de parking automatique :

Une barrière de parking automatique est un dispositif de gestion de l'accès à un parking qui contrôle l'entrée et la sortie des véhicules en utilisant un système automatisé. Elles sont souvent utilisées dans les parkings privés ou payants pour contrôler l'accès et gérer le paiement des frais de stationnement.

Elles sont généralement actionnées par un moteur électrique et peuvent être contrôlées à distance via une télécommande ou un lecteur de badge. Certaines barrières de parking automatiques sont équipées de capteurs de présence ou de détecteurs de mouvement pour ouvrir et fermer automatiquement la barrière lorsqu'un véhicule approche ou s'éloigne. [17]

Et nous les trouvons manuellement ou automatiquement et être de différents types et formes.

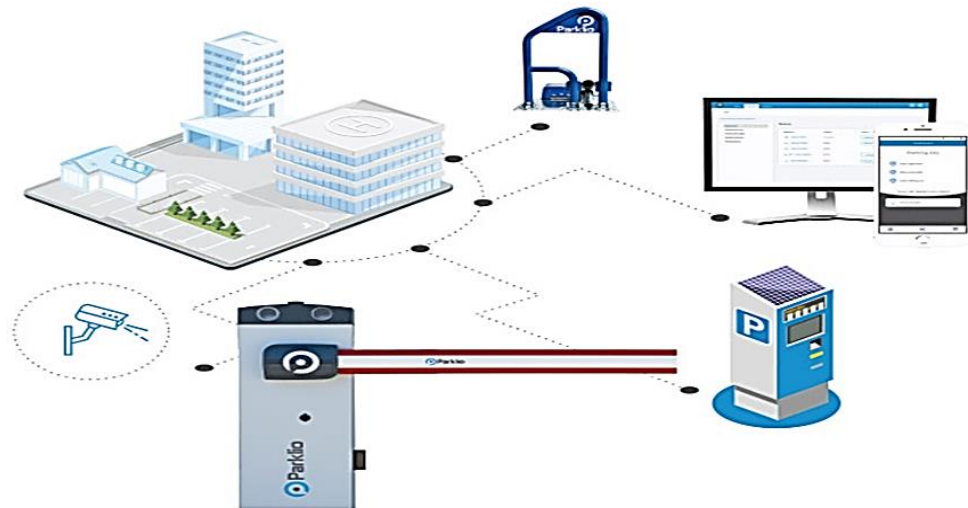


Figure 2.1 Système de Barrière Levante



Figure 2.2 Barrière Levante

2.2 LES TYPES DES BARRIERES :

Les barrières sont classées aux deux grandes catégories :

- Barrières manuelles.
- Barrières automatiques.

2.2.1 Barrière manuelle :

Une barrière levante manuelle est un appareil destiné au contrôle d'accès des véhicules sur un site extérieur. Elle peut être installée en milieu urbain, tertiaire ou encore industriel. La particularité de la barrière levante manuelle est sa facilité de mise en place et d'installation. En effet, sans motorisation, elle ne nécessite que peu de génie civil, sans alimentation électrique.

Le propre de la barrière manuelle est d'être obligatoirement soumis à surveillance et/ou manipulation par un opérateur. Par définition, c'est l'action humaine qui activera l'ouverture ou la fermeture de la barrière manuelle. [18]

Il existe deux types de barrières manuelles :

1. Barrière manuelle BML :

C'est une barrière manuelle levante, sa fonction est de contrôler une voie d'accès peu fréquentée. On l'actionne grâce à un système de contre poids, sa maintenance quasi inexistante et son installation rapide en font la barrière idéale pour les sites nécessitant un contrôle des flux occasionnel. En fonction de la largeur du passage, elle peut mesurer jusqu'à 10m de long.



Figure 2.3 Barrière Manuelle Levante (BML)

Construction :

Elle est composée d'une lisse, d'un poteau principal pied et d'un repos de lisse à partir de 4m de long. Elle appartient à la gamme de barrières manuelles. [19]

2. Barrière manuelle BMP :

La barrière manuelle pivotante, légère et facile à manœuvrer est la solution idéale pour la sécurisation des accès peu fréquentés. La barrière BMP est très simple à installer et nécessite peu de maintenance.



Figure 2.4 : Barrière Manuelle Pivotante

Construction :

Elle est composée d'un poteau principal, Une lisse de forme ovoïde en aluminium avec bandes réfléchissantes rouges, Un repos de lisse fixe ou articulé selon les besoins. [20]

Tableau 2.1 Caractéristiques des Barrières manuelles BML et BMP [19] [20]

Caractéristiques	BML	BMP
Longueur max de la lisse	Sans rideau de lisse : De 2 à 10 m Avec rideau de lisse : De 2 à 7.5 m	De 2 à 10 m
Hauteur de la barrière	1100 mm	1100 mm
Mécanisme	Contre poids	Pivot
Verrouillage de la lisse	En bout de lisse	Manuel
Type d'usage	Usage ponctuel	Usage ponctuel

2.2.2 Barrière automatique :

Les barrières levantes automatiques sont des dispositifs de gestion des accès de véhicules, intégrant une motorisation permettant à la barrière de procéder sans intervention humaine. Elle permet ainsi de faciliter l'accès et la gestion des véhicules, sans avoir à procéder à une manipulation manuelle par l'utilisateur. [21]

1. Barrière levant FB:

Barrière levante automatique pour la gestion d'accès longue portée, jusqu'à 9m. elle a été spécialement conçue pour équiper les sites industriels aux larges accès, la FB s'adapte à tous les types d'environnement. [21]

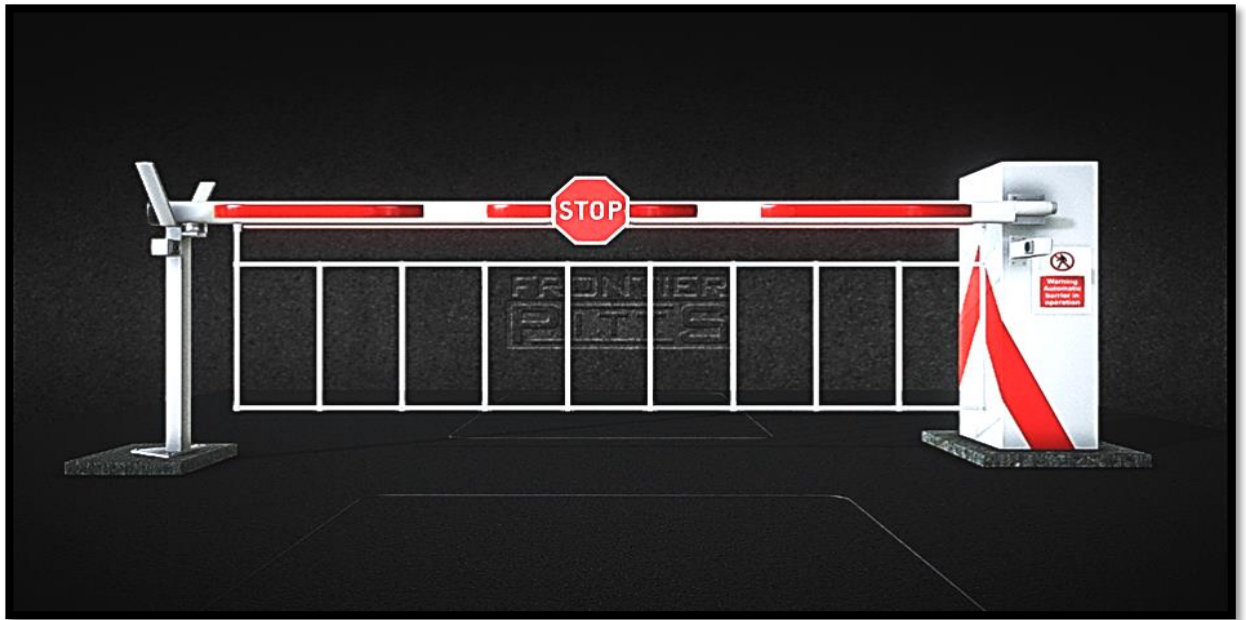


Figure 2.5 : Barrière Levante FB

Construction :

La barrière automatique FB est composée de 3 éléments :

- Un coffret de commande en tôle d'acier, traité et peint.
- Une lisse d'ovoïde en aluminium avec bandes rouges.
- Un repos de lisse fixe ou articulé selon les besoins.

Tableau 2.2 Caractéristiques de Barrière Levante FB [21]

Caractéristiques	Barrière levante FB
Longueur de la barrière	De 2 à 9 m
Hauteur de la barrière	1000 mm
Manœuvre	Automatique, débrayable manuellement en cas de coupure de courant
Position du bras	A gauche ou à droite
Tension d'alimentation	240 V
Verrouillage de la lisse	Ventouse (option)
Temps d'ouverture	7 ou 10 s (selon longueur)
Rideau de lisse bas	Possible jusqu'à 7,5 m
Rideau de lisse intégral	Disponible en option jusqu'à 5,5 m
Type d'usage	Usage intensif

2. Barrière levante FBC :

La barrière automatique FBC est une barrière de moyenne portée (jusqu'à 5.5m) qui assure une parfaite gestion des flux de passage des campings, collectivités, parkings et bâtiments tertiaires. La FBC allie simplicité et très bon rapport qualité/prix. [21]



Figure 2.6 Barrière Levante FBC

Construction :

La barrière FBC est composée de 3 éléments :

- Un coffret de commande en tôle d'acier 2mm, traité et peint.
- Une lisse ovoïde en aluminium avec bandes réfléchissantes rouges.
- Un repos de lisse fixe ou articulé selon les besoins.

Facile à poser, à installer et à entretenir.

3. Barrière levante FBX :

La barrière automatique FBX est une barrière de moyenne/longue portée qui assure une parfaite gestion des flux de passage. Elle offre un large choix d'options permettant notamment l'intégration d'un rideau de lisse jusqu'à 5m afin d'obstruer le passage des piétons. [21]



Figure 2.7 Barrière Levante FBX

Construction :

La barrière FBX est composée de 3 éléments :

- Un coffret de commande en tôle d'acier, traité et peint.
- Une lisse ovoïde en aluminium avec bandes rouges.
- Un repos de lisse fixe ou articulé selon les besoins.

La FBX est de construction très robuste, Facile à poser, à installer et à entretenir.

4. Barrière levante HB :

La barrière automatique HB est une barrière levante électro-hydraulique de grande portée. Fiable et robuste, elle permet la sécurisation des voies de grande largeur et l'intégration de rideau de lisse intégral jusqu'à 7.5m. Barrière levant automatique de grande portée - Jusqu'à 11m. [21]



Figure 2.8 Barrière Levante HB

Construction :

- Une lisse en aluminium.
- Le pied et le repos de lisse en acier.
- La qualité de ses matériaux de construction et son mécanisme hydraulique font de la HB la plus puissante.

Version ATEX :

La version HB ATEX a été conçue pour les sites gaziers à haute risque d'explosion, son mécanisme est déporté dans un boîtier ATEX.

Tableau 2.3 Caractéristiques des Barrières Levantes FBC, FBX et HB [21]

Caractéristiques	Barrière levante FBC	Barrière levante FBX	Barrière levante HB
Longueur de la barrière	De 2 à 5.50 m	De 2 à 6 m	De 4 à 10 m
Hauteur de la barrière	900 mm	1000 mm	1000 mm
Manœuvre	Automatique, débrayable manuellement en cas de coupure de courant	Automatique, débrayable manuellement en cas de coupure de courant	Hydraulique, débrayable manuellement en cas de coupure de courant
Position du bras	A gauche	A gauche ou à droite	A gauche ou à droite
Tension d'alimentation	240 V	240 V	240 V
Verrouillage de la lisse	-	Ventouse (option)	Ventouse (option)
Temps d'ouverture	3 ou 6 s (selon longueur de la lisse)	3.5 ou 7 s (selon longueur de lisse)	10 à 15 s (selon motorisation)
Rideau de lisse bas	Possible jusqu'à 4 m	Possible jusqu'à 5 m	Possible jusqu'à 9 m
Rideau de lisse intégral	-	Possible jusqu'à 3 m	Possible jusqu'à 7,5 m
Type d'usage	Usage intensif	Usage intensif	Usage intensif

5. Barrière automatique TERRA compact :

La barrière levante COMPACT est une barrière de moyenne/longue portée (jusqu'à 6m) qui allie robustesse et design. Elle résiste aux impacts d'un véhicule de 3500 kg lancé à 50 km/h. La barrière COMPACT assure une très bonne gestion des flux de passage tout en assurant un très haut degré de sécurité. [21]



Figure 2.9 Barrière Automatique TERRA Compact

Construction :

La barrière TERRA Compact est une barrière anti bélier composée de 3 éléments :

- Un coffret de commande en acier haute résistance.
- Une lisse abritant un câble en acier.
- Un poteau de réception en acier haute résistance.

6. Barrière automatique TERRA ULTIMATE :

La barrière ULTIMATE est une barrière anti bélier automatique de courte/moyenne portée (jusqu'à 4.50m). Elle résiste aux impacts d'un véhicule de 7500kg lancé à 80km/h. Cette barrière automatique est la solution idéale pour la sécurisation des sites les plus vulnérables. [21]



Figure 2.10 Barrière Automatique TERRA ULTIMATE

Construction :

La barrière TERRA ULTIMATE est une barrière anti bélier composée de 3 éléments en acier haute résistance :

- Un coffret de commande.
- Une poutre.
- Un poteau de réception.

Tableau 2.4 Caractéristiques des Barrières Automatiques TERRA Compact et TERRA ULTIMATE [21]

Caractéristiques	Barrière automatique TERRA compact	Barrière automatique TERRA ULTIMATE
Longueur de la barrière	De 2 à 6 m	De 2 à 4,5 m
Hauteur de la barrière	900 mm	1050 mm
Manœuvre	Hydraulique, débrayable manuellement en cas de coupure de courant	Hydraulique, débrayable manuellement en cas de coupure de courant
Position du bras	A gauche ou à droite	-
Tension d'alimentation	380 V triphasé + N + T	380V triphasé + N + T
Verrouillage de la lisse	En bout de lisse	-
Temps d'ouverture	10 à 15 s (selon motorisation et longueur de lisse)	10 à 15 s (selon motorisation/longueur)
Rideau de lisse bas	Inclus	-
Profondeur de génie civil	500 mm	500mm
Type d'usage	Usage intensif	Usage intensif
Matériau de construction	Acier haute résistance	Acier haute résistance
Certification	PAS68	PAS 68

7. Barrière à poutre coulissante (BC) :

La poutre BARRICADE est un barrage routier coulissant qui offre une réponse parfaitement adaptée aux besoins de sécurisation et gestion des flux de passage. Bien que non crash testée, cette poutre peut résister à un véhicule de 3000 kg à 15 km/h. Sa structure en acier haute résistance fait de cette poutre une solution alternative aux équipements de la gamme TERRA.

C'est la solution anti vandale pour la sécurisation de vos accès véhicules. Jusqu'à 8m (en standard). Existe en version automatique et manuelle. [21]



Figure 2.11 Barrière à poutre coulissante (BC)

Construction :

La poutre Barricade est composée de 3 parties principales :

- Poteau principal
- Poutre
- Poteau de réception

Cette poutre ne nécessite que 40cm de profondeur de génie civil, elle peut être personnalisée selon vos envies et besoins (accessoires, barreaudages en partie basse, etc.)

Tableau 2.5 Caractéristiques de Barrière à poutre coulissante (BC) [21]

Caractéristiques	Barrières à poutre coulissante
Largeur de passage libre	Jusqu'à 8 m
Hauteur de la poutre barricade	900 mm
Profondeur de génie civil	40 cm
Vitesse d'ouverture/fermeture	7 à 15 s pour la version automatique
Mécanisme	Electrique 220V ou manuel
Type d'usage	Usage intensif

2.3 SOURCES D'ALIMENTATION :

Les barrières levantes nécessitent une source d'alimentation pour leur fonctionnement. Les sources couramment utilisées comprennent :

- **Electricité** : de nombreuses barrières levantes sont alimentées par l'électricité, soit directement via le réseau électrique, soit à l'aide de batteries rechargeables. L'alimentation électrique permet de soulever et d'abaisser rapidement la barrière avec précision.
- **Hydraulique** : certains mécanismes des barrières levantes utilisent un système hydraulique pour générer la force nécessaire pour soulever la barrière. Un fluide hydraulique est pompe dans des vérins hydrauliques qui exercent une pression pour élever la barrière. Ce système offre généralement une grande capacité de levage et une bonne fiabilité.
- **Pneumatique** : les barrières levantes pneumatiques utilisent de l'air comprimé pour soulever la barrière. Un compresseur d'air est utilisé pour générer la pression d'air nécessaire pour actionner les vérins pneumatiques qui soulèvent la barrière. Ce système

est souvent utilisé lorsque l'alimentation électrique n'est pas disponible ou difficile à mettre en place.

- **Manuel** : dans certaines situations, les barrières levantes peuvent être actionnées manuellement. Cela peut se faire en utilisant un levier, une manivelle ou tout autre dispositif mécanique pour soulever ou abaisser la barrière. Ce type de mécanisme est généralement utilisé dans des situations où l'utilisation d'une source d'alimentation externe n'est pas pratique.

2.4 LA STRUCTURE DE BARRIERE LEVANTE :

Le schéma suivant montre la structure principale d'une barrière qui est composée de 2 chaînes principales :

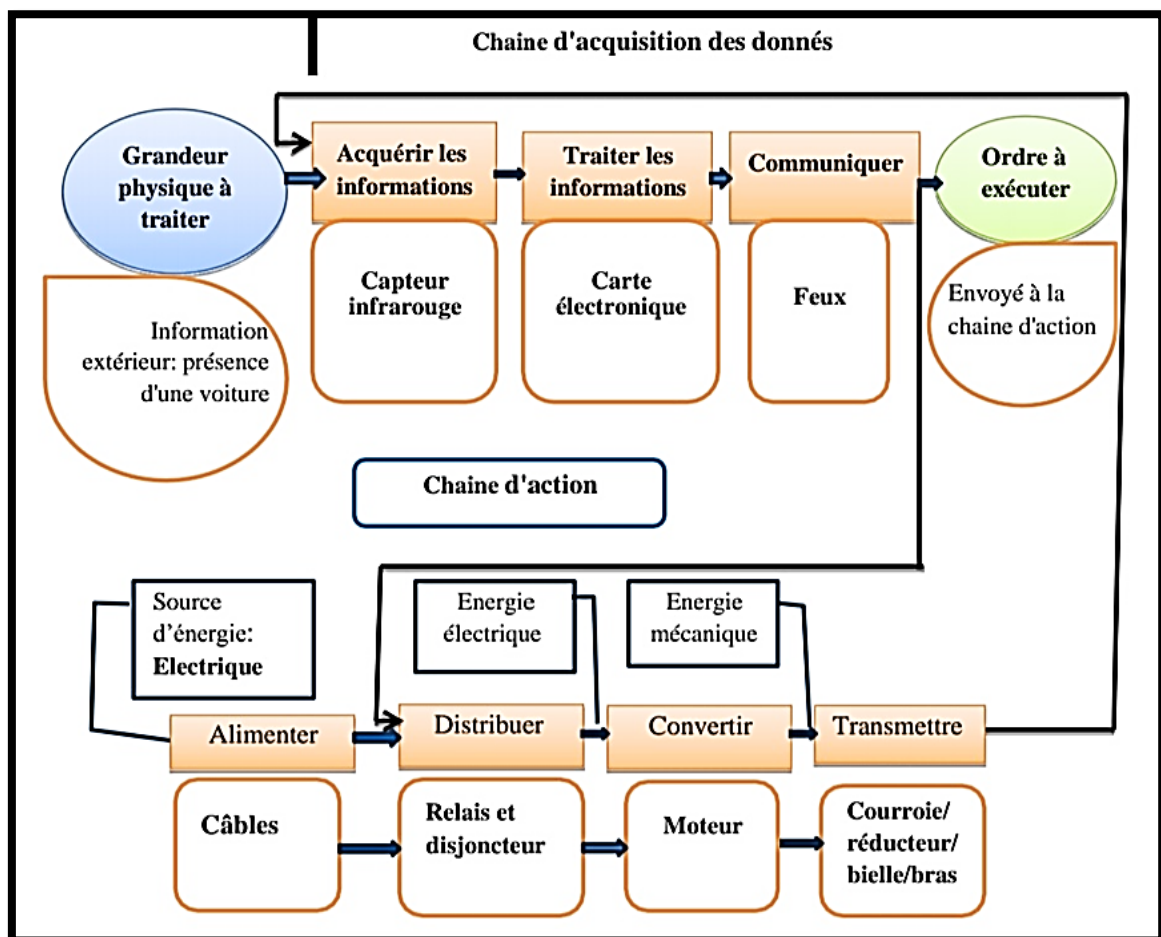


Figure 2.12 Chaîne d'acquisition des données et Chaîne d'action [22]

- **Acquérir** : le récepteur du badge détecte l'arrivée d'un usager et le photocellule contrôle le passage du véhicule.

- **Traiter** : le circuit électrique traite l'information reçue par la présence du véhicule.
- **Communiquer** : le badge envoie le bon ordre en fonction de la position de la lisse. Par exemple, si la lisse est fermée, le message envoyé sera de l'ouvrir.

❖ **On peut définir cette énergie comme**

L'énergie électrique envoyée se transforme en énergie mécanique. C'est cette énergie mécanique qui va permettre le mouvement de la lisse.

- **Source d'énergie** : c'est l'énergie électrique.
- **Alimenter** : le bornier de raccordement fournit l'énergie électrique à tout le système.
- **Distribuer** : la bonne quantité d'énergie est envoyée à l'ordre à exécuter : lever ou baisser la lisse est envoyée au motoréducteur en fonction des ordres reçus.
- **Convertir** : le motoréducteur transforme l'énergie électrique en énergie mécanique.
- **Transmettre** : l'énergie mécanique est transmise du motoréducteur à la lisse.
- **Action** : la barrière s'ouvre ou se ferme, selon l'ordre reçu. [22]

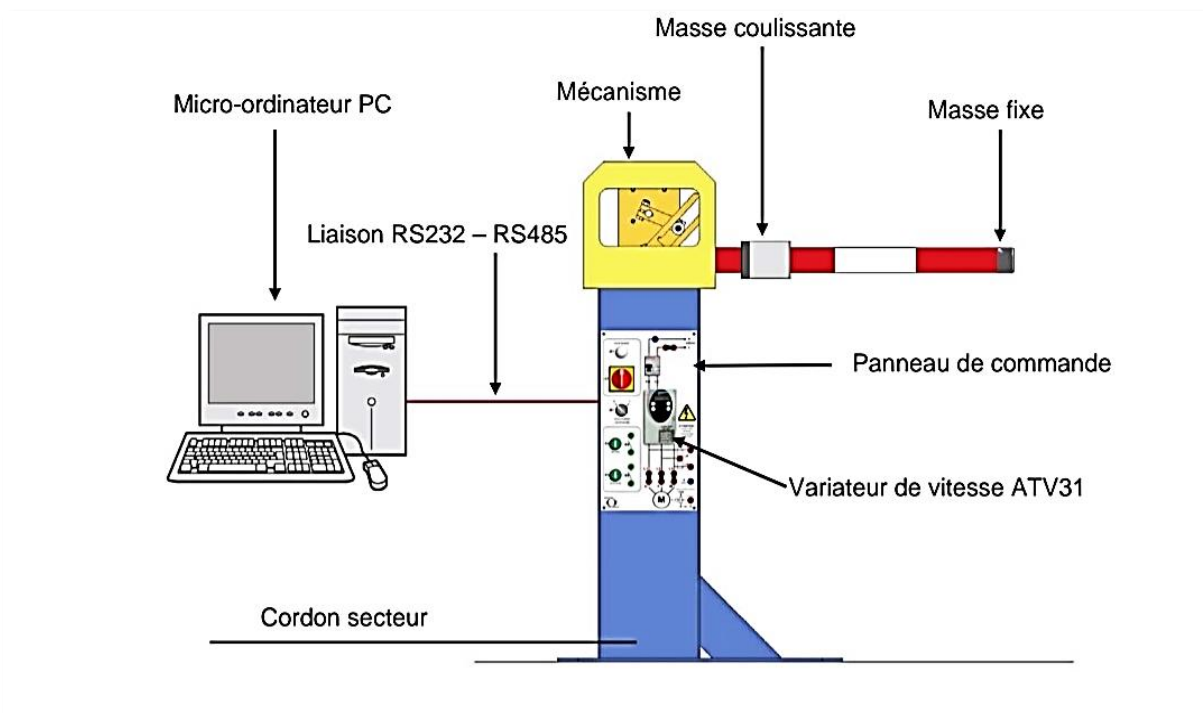


Figure 2.13 Schéma pour le système de la barrière levante [23]

2.5 MECANISME DES BARRIERES LEVANTES :

Le mécanisme de la barrière levante est largement utilisé dans divers domaines pour contrôler et réguler le flux de circulation. Que ce soit sur les routes, dans les parkings, les passages à niveau ou les zones de sécurité, les barrières levantes sont essentielles pour assurer la sécurité et la gestion efficace du trafic. Ces dispositifs fonctionnent en utilisant un mécanisme qui permet de soulever et d'abaisser une barrière de manière contrôlée.

2.5.1 Principe de Fonctionnement mécanique :

La partie opérative (tête de barrière) est constituée d'un mécanisme de transformation de mouvement de type bielle-manivelle mu par un moteur asynchrone triphasé piloté dans les deux sens de marche pour réaliser les mouvements de montée et de descente. [23]

Le système de transformation de mouvement est constitué d'une manivelle pivotant autour de l'axe du moteur sur une plage de 270°.

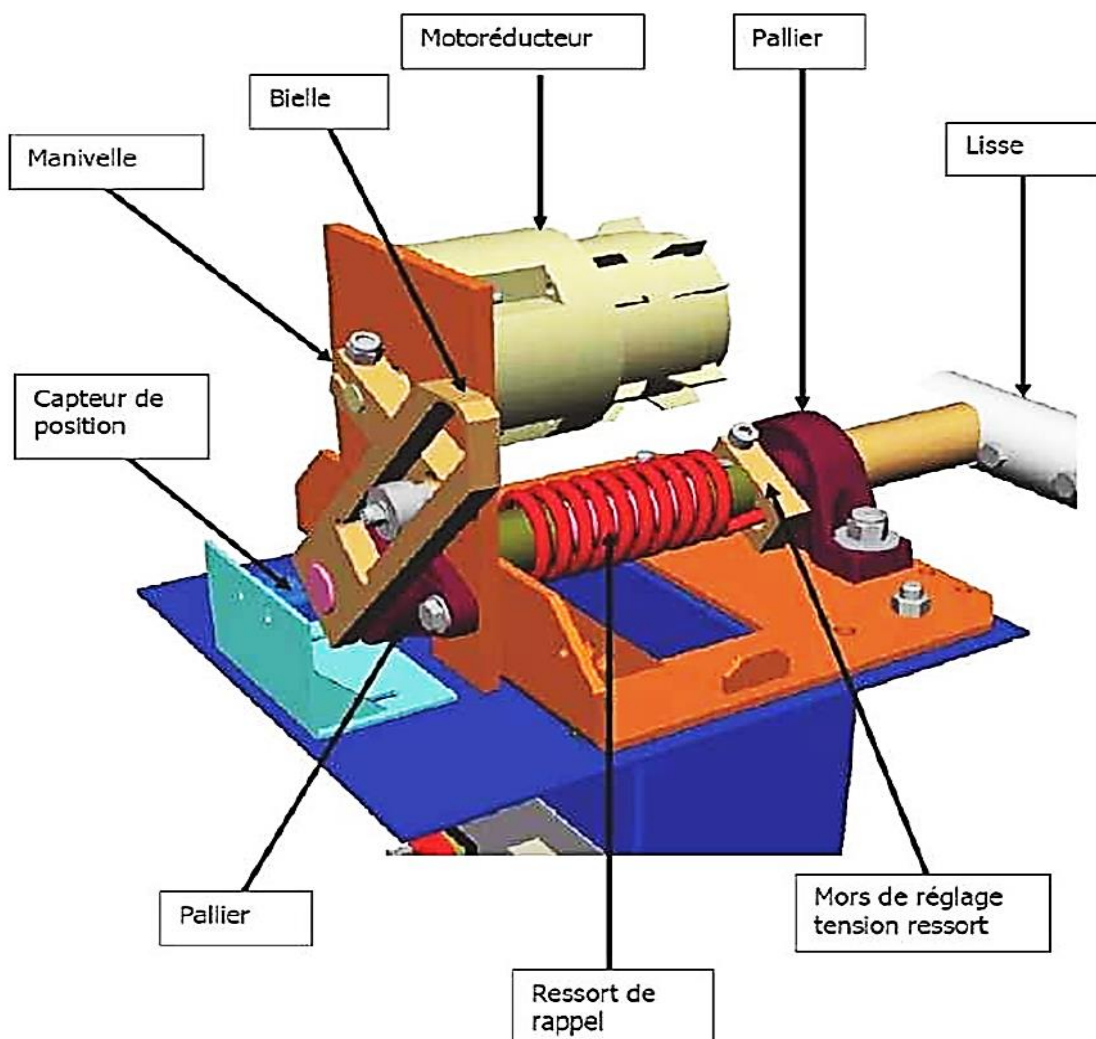


Figure 2.14 Le mécanisme de la tête de barrière levante [23]

L'extrémité de la manivelle est pourvue d'un galet qui vient rouler dans la rainure d'une bielle fixée à une extrémité d'un axe dont l'autre extrémité supporte la lisse.

La limitation de la plage angulaire de rotation de la manivelle est assurée par une butée caoutchouc située dans la partie basse de la rainure de la bielle qui joue également un rôle d'amortisseur. [23]

Ce débattement limité permet d'avoir une symétrie parfaite des positions de la bielle pour les positions de lisse basse et haute car la butée mécanique sert pour les deux sens de rotation du moteur.

Le mouvement de la manivelle entraîne la bielle dans un mouvement angulaire autour de l'axe de support de la lisse. Ce dernier étant solidaire de la bielle, la lisse effectue le même mouvement que la bielle en amplitude mais décalé suivant la position de fixation de la lisse.

Le démarrage s'effectue à couple très faible car la position de démarrage de la manivelle se situe dans une plage angulaire proche de la perpendiculaire à la direction de l'effort. [23]

La caractéristique de cette transformation de mouvement est de donner une vitesse constante à la bielle et donc à la lisse pour une vitesse de rotation constante de la manivelle dans la plage angulaire de 90° de rotation de bielle. Cette transformation de mouvement est donc une combinaison de 2 rotations d'amplitudes différentes : 270° pour la rotation de la manivelle et 90° pour la rotation de la bielle (donc de la lisse). [23]

Rôle du Ressort :

- **Rôle d'accumulateur d'énergie potentielle :** La compression (déformation) du ressort provoqué par la force de pesanteur de la lisse lors de la descente de celle-ci permet l'accumulation d'une énergie potentielle élastique qui est restituée lorsque la déformation de ce dernier disparaît lors de la montée de la lisse.
- **Rôle de sécurité :** Le ressort est taré de manière à provoquer la remontée de la lisse en position verticale sans l'aide d'une force extérieure : En cas de coupure d'énergie électrique, la remontée de la lisse permettra une évacuation rapide dans le cas d'un parc de stationnement sous-terrain par exemple. [23]

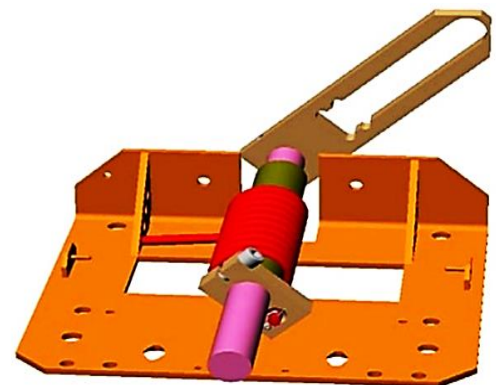


Figure 2.15 Ressort

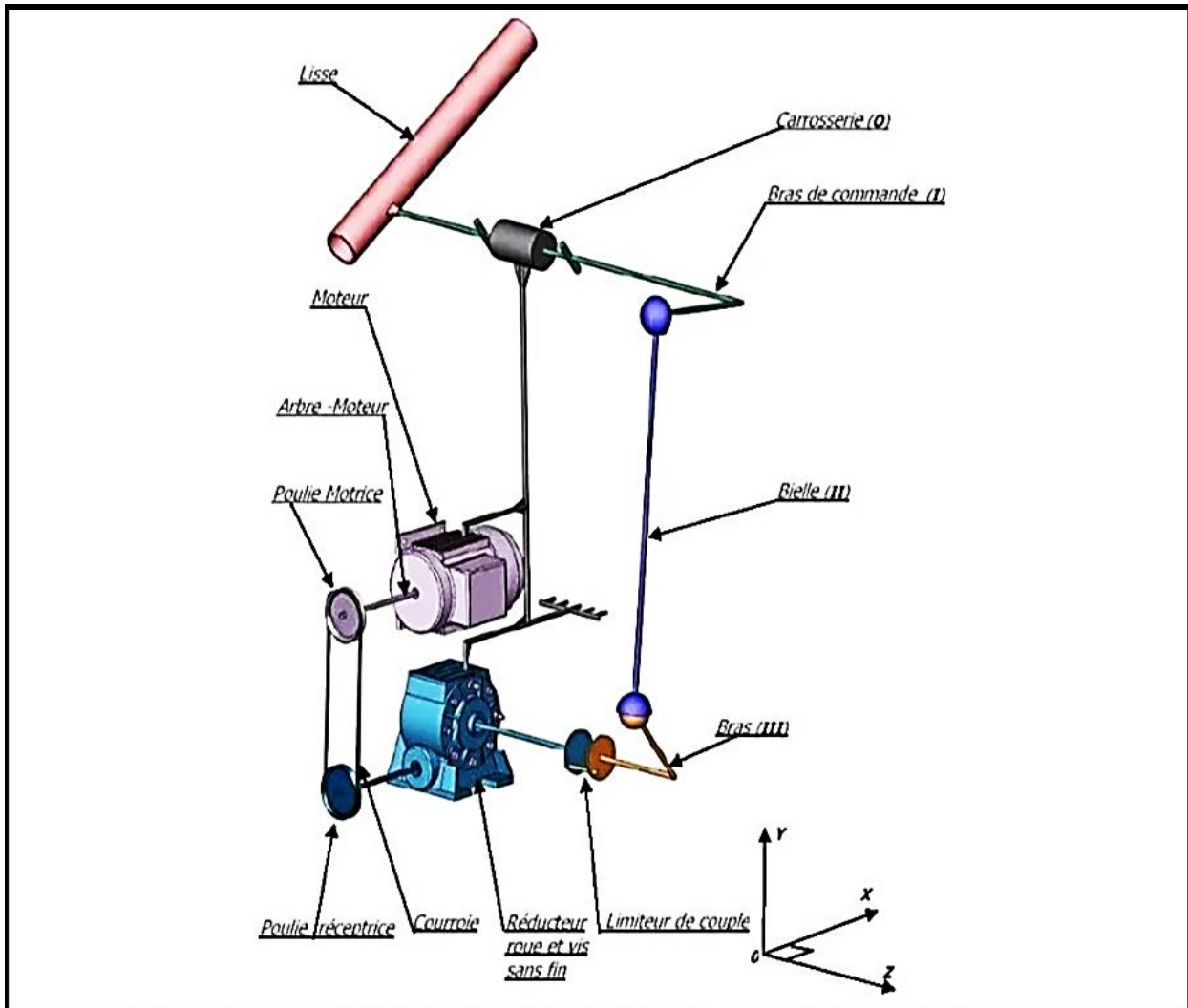


Figure 2.16 Schéma cinématique des mécanismes de barrière levante [24]

2.5.2 Schéma bloc de la chaîne d'énergie :

Dans la figure suivante on va voir le schéma bloc de la chaîne d'énergie d'une barrière levante :

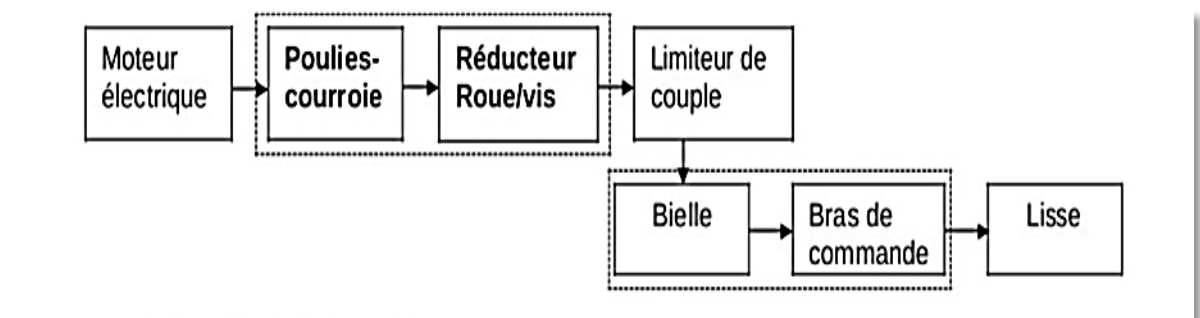


Figure 2.17 Schéma bloc de la chaîne d'énergie [24]

2.5.3 Le diagramme fonctionnel de barrière levante :

Dans la figure suivante, on va présenter le diagramme fonctionnel de la barrière :

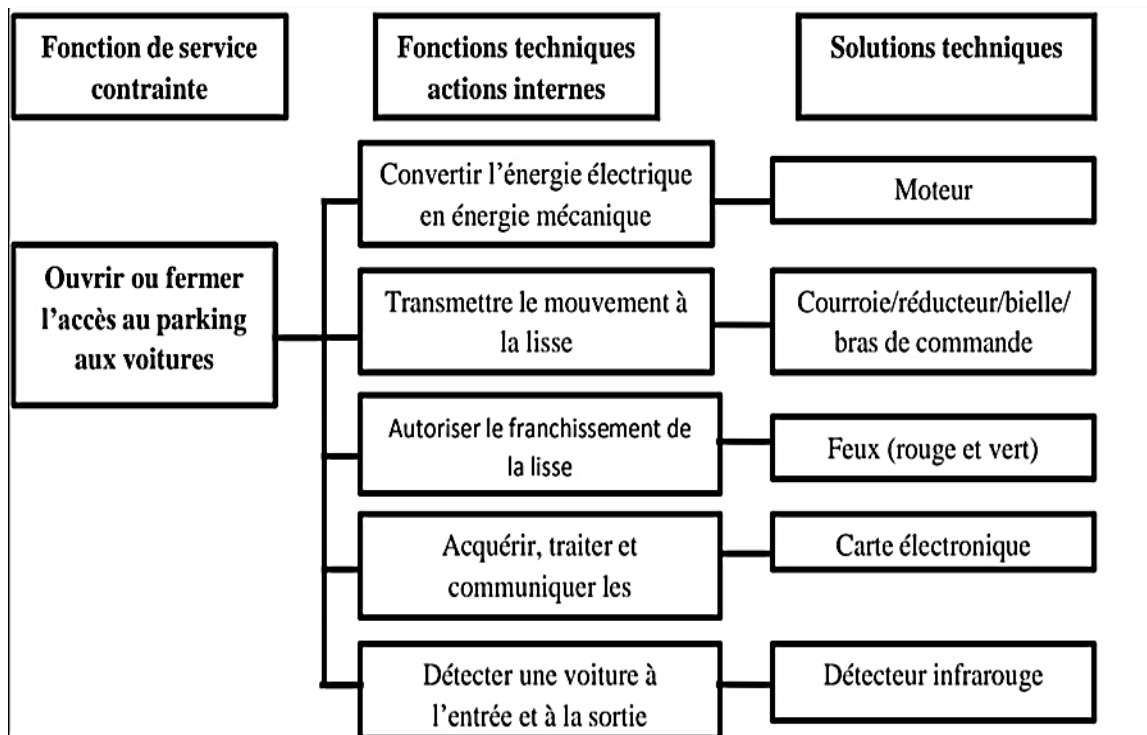


Figure 2.18 Le diagramme fonctionnel de barrière levante [22]

A partir de ce qui précède, nous trouvons que :

- L'ouverture de la lisse est commandée par l'utilisateur (via un interrupteur à clef, un bouton poussoir, un émetteur radio), par des boucles de détections enterrées sous la chaussée, ou par un organe extérieur.
- La fermeture est commandée de la même manière, ou automatiquement au terme d'une temporisation.
- Une commande STOP arrête immédiatement le moteur mais ne cesse pas complètement le mouvement de la lisse : celle-ci peut continuer à descendre progressivement en raison de son poids et de sa position angulaire.
- L'option "capteur de position analogique" permet de réguler la position de la lisse autour du STOP et de palier à ce phénomène.
- Le mouvement créé par le motoréducteur est transmis à la lisse par un système de bielle/manivelle.

- Un ou deux ressorts d'équilibrage jouent le rôle de contrepoids, de manière à assister le moteur tant à l'ouverture qu'à la fermeture de la barrière.
- La vitesse de mouvement de la lisse contrôlée par variateur de fréquence est réglable tant à l'ouverture qu'à la fermeture. Les mouvements sont paramétrés en usine pour offrir une accélération vive et une décélération douce en fin de mouvement. [25]

2.6 LES COMPOSANTS, DISPOSITIFS DE SECURITE ET DISPOSITIFS DE COMMANDE :

2.6.1 Les composants de la barrière levante :

Les barrières sont notamment charpentées d'un coffre (Fut) et d'une lisse de fer ou d'aluminium...de longueur jusqu'à 6m (au plus), elles sont fonctionnées pour autoriser à l'interdire le passage des voitures ou les Personnes. La boîte se compose d'un moteur, d'un réducteur, une carte électronique, d'un ressort, bielle et un bras de commande.

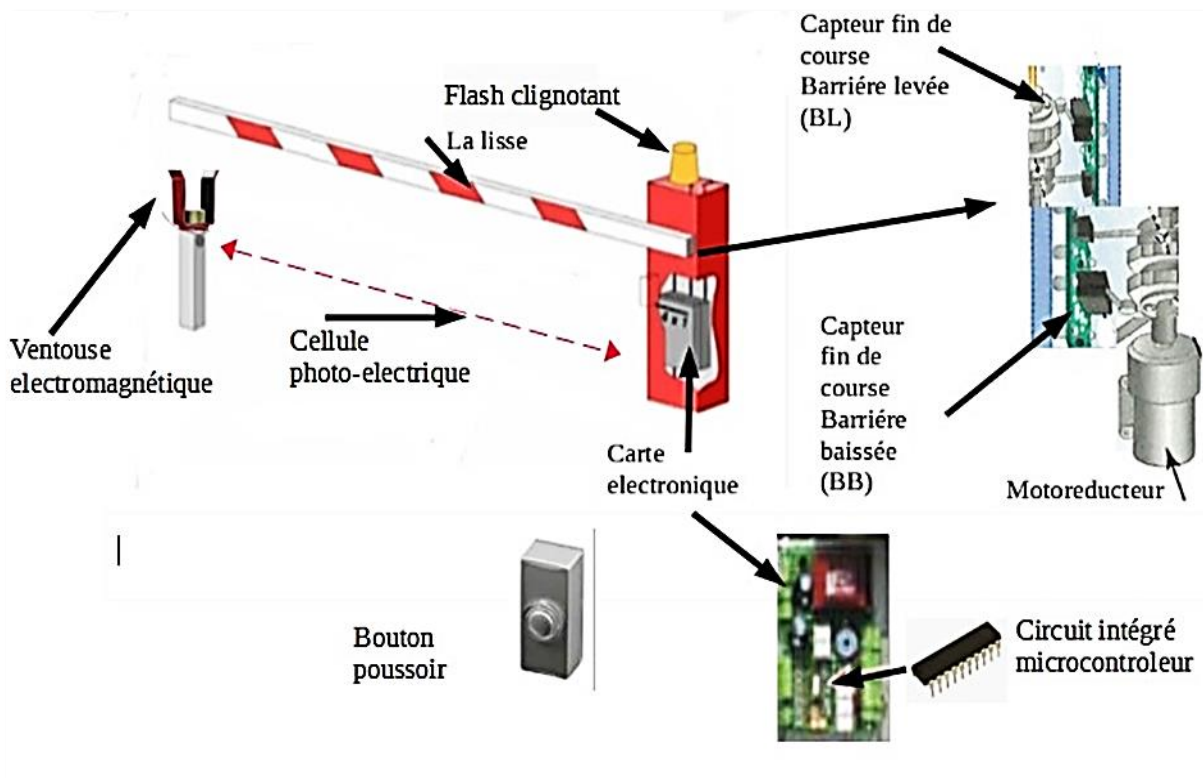


Figure 2.19 Les composants de la barrière levante automatique à l'extérieur

Explication de rôle des composants de la barrière levante :

- **Le fût ou le coffre de la barrière** : il contient notamment le moteur de la barrière, la couleur du coffre peut être choisie (selon le modèle et le fournisseur). Le fût permet de lever ou baisser la lisse.
- **La lisse** : elle peut être ronde ou rectangulaire. Elle interdit ou autorise le passage d'un véhicule. Une extension peut être ajoutée à la barrière selon la dimension souhaitée.
- **La flasque de fixation.**
- **La plaque de fondation.**
- **Le ressort d'équilibrage.**
- **Le flash** : il permet d'indiquer au passant l'ouverture ou la fermeture de la barrière.
- **Les photocellules** : elles ont pour rôle de détecter le passage d'un véhicule au moment de la fermeture de la barrière. Ainsi, en cas de passage, celle-ci se réouvre aussitôt.
- **La boucle de détection au sol** : il en existe deux sortes. Celle de sécurité permet d'empêcher la barrière de se refermer sur un véhicule. Celle d'ouverture permet d'ouvrir la barrière automatiquement quand un véhicule passe dessus. [26]

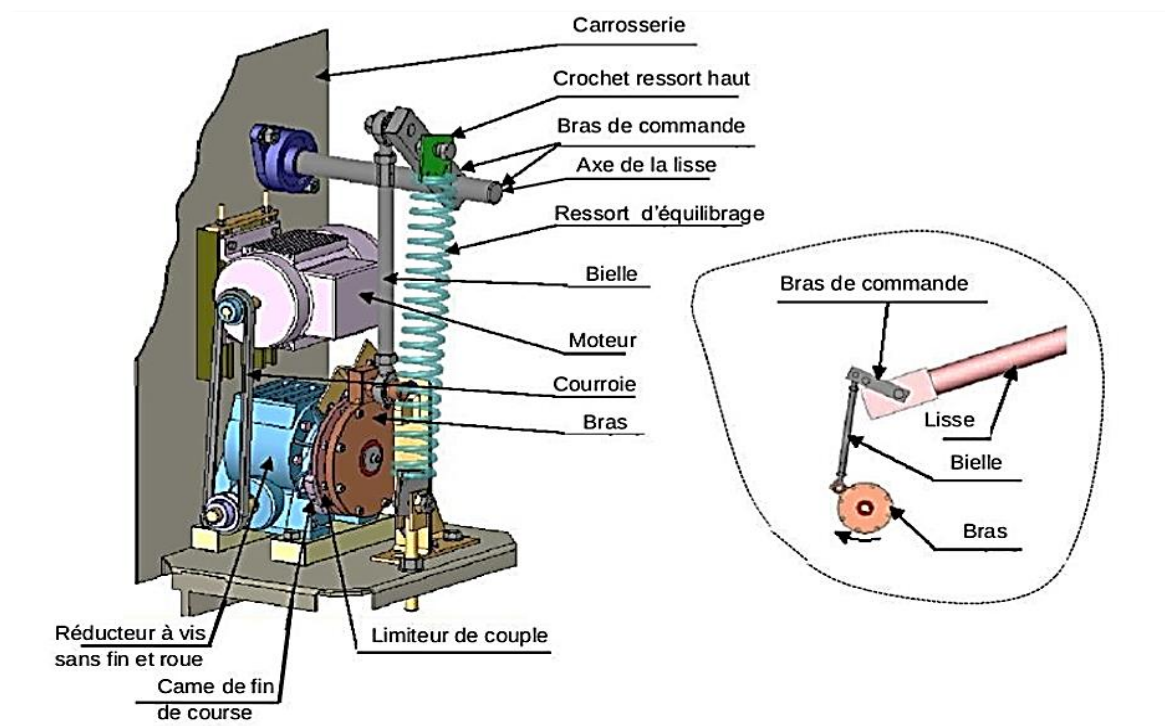


Figure 2.20 Les composants du groupe mécanique la barrière à l'intérieur selon la première façade

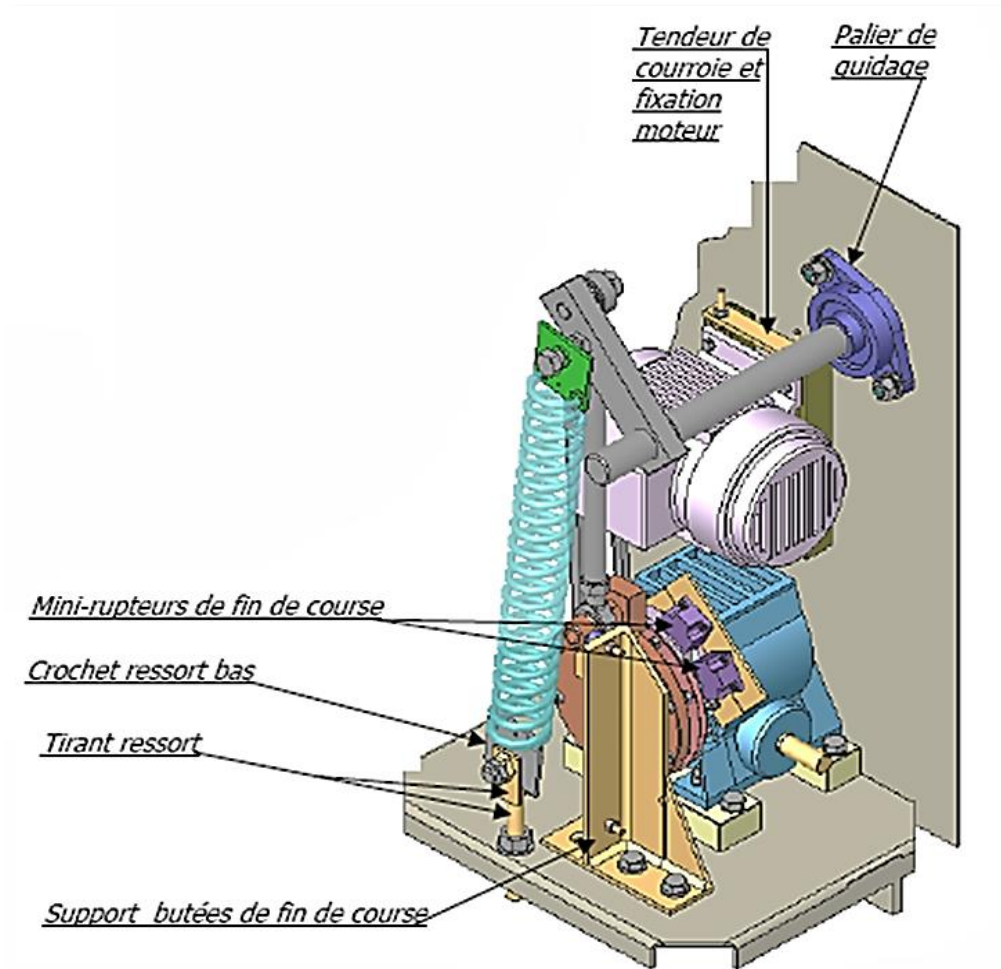


Figure 2.21 Les composants du groupe mécanique la barrière à l'intérieur selon la deuxième façade [24]

2.6.2 Dispositifs de sécurité :

- Couverture de protection transparente permet de voir le déplacement de la lisse.
- Une manivelle autorise l'ouverture et la fermeture manuelle de la barrière en interdisant toutes les commandes électriques.
- Un arrêt général sur le pupitre du Coffret de Commande Didactique provoque la mise hors tension des organes de puissance de la barrière.

❖ Ouverture de sécurité en cas d'urgence :

- En cas de coupure d'électricité ou d'anomalie, le système de déblocage permet l'ouverture manuelle de la barrière levante, La fermeture automatique peut être désactivée.

- Le verrouillage et déverrouillage réalisé manuellement pour placer la lisse en position verticale et permettre l'entrée ou la sortie des véhicules au parc de stationnement. [22]

2.6.3 Dispositifs de commande :

La logique de commande coordonne l'activité de la barrière : gestion des mouvements, des options, traitement des informations entrantes et sortantes, etc. Ces informations peuvent cependant être rapatriées et traitées par un terminal extérieur. [25]

La logique enregistre et affiche l'historique des dernières opérations effectuées ainsi que les éventuels défauts empêchant le mouvement de la barrière. La logique de commande est l'interface entre l'utilisateur et la barrière, qui gère toutes les actions de cette dernière, y compris les éventuelles options. [25]

➤ Pour commander l'ouverture d'une barrière levante, plusieurs solutions sont envisageables :

- Un bouton poussoir.
- Une télécommande.
- Une carte magnétique.
- Une cellule photo-électrique.
- Une boucle inductive de détection.
- Un clavier.
- Un code.

2.7 CONCLUSION :

Dans ce chapitre, nous avons la chance de connaître les différents types des barrières levant (manuelles et automatiques) qu'ils sont des appareils importants pour contrôler le flux de circulation et garantir la haute sécurité dans de nombreuses fonctions et conditions. Notamment, nous avons connu que ces barrières peuvent être alimentées par plusieurs sources comme l'électricité, un système hydraulique...etc, selon les besoins de chaque utilisateur.

Quelle que soit la source d'alimentation utilisée, la structure et le mécanisme de cette barrière levant permet de gérer tous les mouvements de manière contrôlée et plus sécurisé pour le trafic et les utilisateurs.

Chapitre 3: Arduino et périphériques

INTRODUCTION :

Pont tendu entre le monde réel et le monde numérique, Arduino permet d'étendre les capacités de relations humain/machine ou environnement/machine. Arduino est un projet en source ouverte : la communauté importante d'utilisateurs et de concepteurs permet à chacun de trouver les réponses à ses questions. [28]

Notre objectif dans ce chapitre est de mettre l'accent sur la description des différents matériels et logiciels qu'on a utilisé pour la réalisation.

3.1 PRESENTATION GENERAL :

Arduino est une plate-forme de prototypage rapide : un ensemble d'outils développés pour faciliter la conception de montages à base de microcontrôleur, sans perdre trop de temps à en apprendre les tenants et aboutissants.

Les circuits à microcontrôleur, grâce à la grande facilité de leur mise en œuvre et la diminution de leur coût sont de plus en plus fréquemment employés et remplacent avantageusement les circuits électroniques classiques.

La plate-forme **Arduino** comporte :

- Du **matériel** : une collection de cartes à microcontrôleurs,
- Du **logiciel** : permettant la programmation, la communication et intégrant de nombreuses bibliothèques de fonctions,
- Un **site Internet** : information, téléchargements, documentation, forums... [29]

3.2 DESCRIPTION DE LA PARTIE MATERIELLE :

Sélection du matériel a été faite judicieusement pour assurer un fonctionnement fluide et efficace. Nous possédons les moyens à notre disposition pour construire une barrière levant automatique. Nous pouvons utiliser Arduino pour gérer efficacement les parkings miniatures, et la même approche peut également être appliquée aux parkings à échelle réelle. La plate-forme de développement de choix pour moi est l'environnement de développement Arduino. Le domaine de l'électronique et de l'informatique embarquée est chargé de produire plusieurs cartes

électroniques. Pour la mise en œuvre, nous avons utilisé l'Arduino MEGA comme outil programmable. À l'avenir, ce qui suit décrira notre approche et notre méthodologie. Nous fournirons un aperçu complet des cartes électroniques, en nous concentrant spécifiquement sur la carte Arduino MEGA et d'autres composants connexes. Le projet a utilisé une combinaison d'outils logiciels et de composants électroniques.

3.2.1 Définition d'une carte Arduino :

Une carte Arduino est une carte électronique de prototypage rapide sur laquelle on peut raccorder des capteurs (entrées) et des actionneurs (sorties). Elle est constituée de plusieurs composants électroniques dont le principal est un microcontrôleur permettant de stocker et d'exécuter un programme informatique.

Cette carte permet de réaliser la fonction TRAITER l'information. [30]

Autrement, Une carte Arduino est une carte de développement électronique basée sur un microcontrôleur. Elle est conçue pour être facilement utilisable par les développeurs et les hobbyistes pour créer des projets électroniques. [37]

3.2.2 Les différents types des cartes Arduino :

Les cartes Arduino sont principalement utilisées pour contrôler des appareils électroniques en utilisant du code informatique, mais elles peuvent également être utilisées pour traiter et afficher des données. Elles sont souvent utilisées dans des projets de robotique, de domotique, de domaine de l'Internet des objets (IoT) et dans de nombreux autres domaines de l'électronique de loisir et professionnelle. [37]

Il existe différents types de cartes Arduino, chacune ayant ses propres spécifications et fonctionnalités. Voici quelques-uns des principaux types de cartes Arduino : [31]

- **Arduino Uno** : C'est l'une des cartes Arduino les plus populaires et largement utilisées. Elle est basée sur le microcontrôleur ATmega328P et possède 14 broches d'E/S numériques, 6 broches d'entrée analogiques, un oscillateur, un port USB, etc. C'est une carte polyvalente adaptée à de nombreux projets. [31]



Figure 3.1 La carte Arduino UNO

- **Arduino Mega** : Cette carte est basée sur le microcontrôleur ATmega2560. Elle offre beaucoup plus de broches d'E/S numériques et analogiques que l'Arduino Uno, ce qui la rend adaptée aux projets nécessitant de nombreuses connexions. [31]

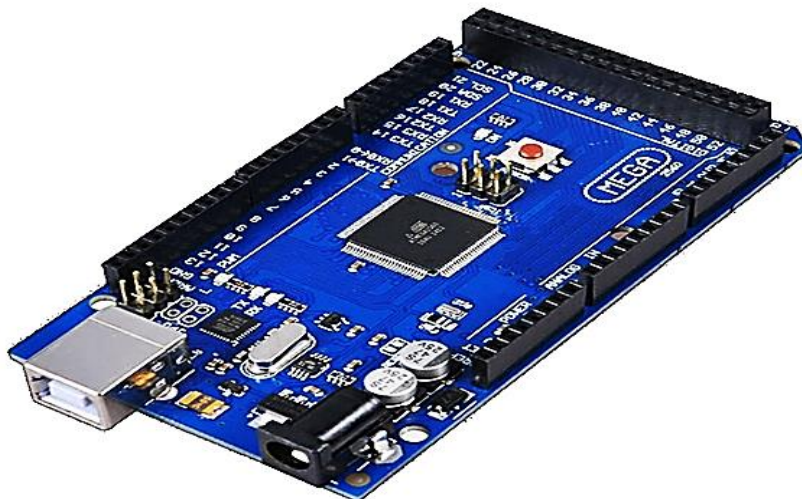


Figure 3.2 La carte Arduino MEGA

- **Arduino Nano** : Cette carte est une version compacte de l'Arduino Uno, offrant des fonctionnalités similaires mais dans un format plus petit. Elle est très utilisée dans les projets où l'espace est limité. [31]

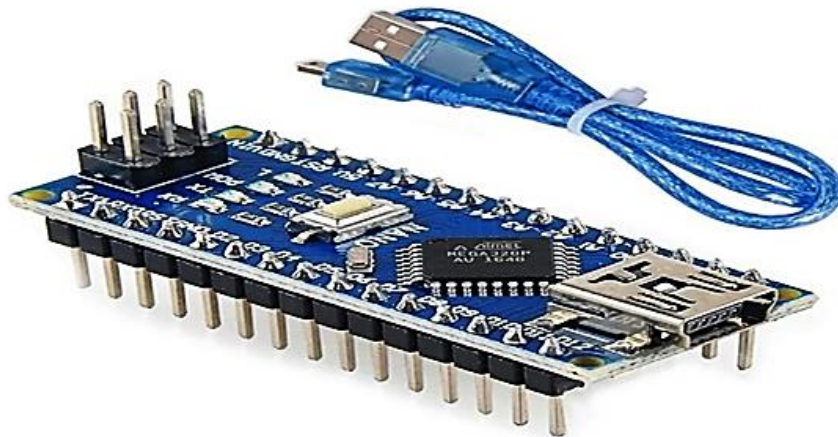


Figure 3.3 La carte Arduino NANO

- **Arduino Leonardo** : Cette carte utilise le microcontrôleur ATmega32U4 et se distingue par sa capacité à agir en tant que périphérique USB HID (Human Interface Device). Cela permet à l'Arduino Leonardo d'émuler un clavier ou une souris, ce qui est utile pour les projets d'interface utilisateur. [31]



Figure 3.4 La carte Arduino LEONARDO

- **Arduino Due** : Cette carte est basée sur le microcontrôleur SAM3X8E, qui utilise une architecture ARM Cortex-M3 32 bits. Elle offre une plus grande puissance de traitement et une plus grande mémoire que les cartes Arduino basées sur l'architecture AVR, ce qui la rend adaptée aux projets nécessitant des calculs complexes ou des applications nécessitant une grande quantité de données. [31]

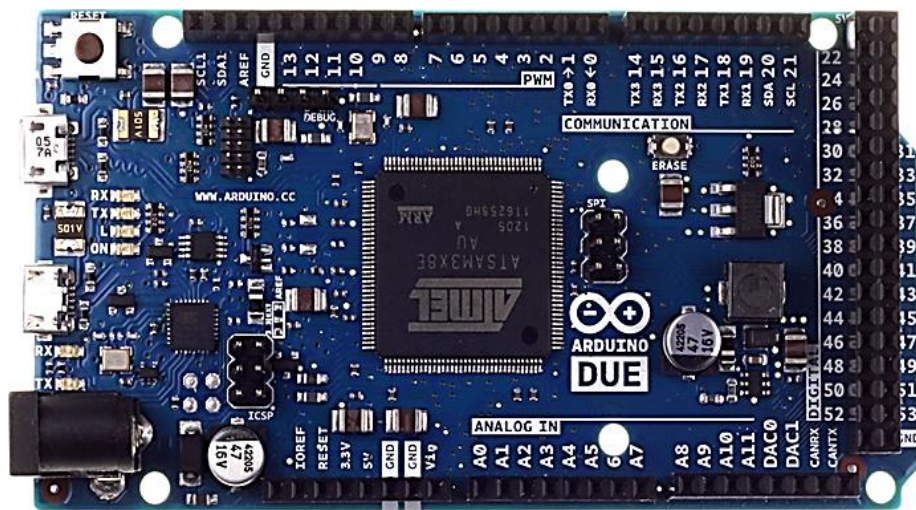


Figure 3.5 La carte Arduino DUE

En plus de ces cartes, il existe d'autres variantes et modèles d'Arduino, tels que l'Arduino Mini, l'Arduino Pro, l'Arduino Nano Every, l'Arduino MKR, etc. Chaque carte a ses propres caractéristiques et avantages, permettant aux utilisateurs de choisir celle qui convient le mieux à leur projet spécifique. [31]

3.2.3 Présentation de la carte Arduino MEGA 2560 :

La carte Arduino Mega 2560 est basée sur un ATmega2560 disposant de 54 E/S dont 14 PWM, 16 analogiques et 4 UARTs. Cette carte est idéale pour des applications exigeant des caractéristiques plus complètes que l'Uno. [32]

La carte Arduino Mega 2560, bien qu'elle soit un peu moins connue, est la grande sœur, plus rapide, de la carte Uno. Dont les capacités équivalent à quatre cartes Uno combinées, comporte un microcontrôleur et est construite autour du puissant microprocesseur ATmega2560. Cette carte robuste résiste à presque tout. [32]



Figure 3.6 Présentation de la carte Arduino MEGA

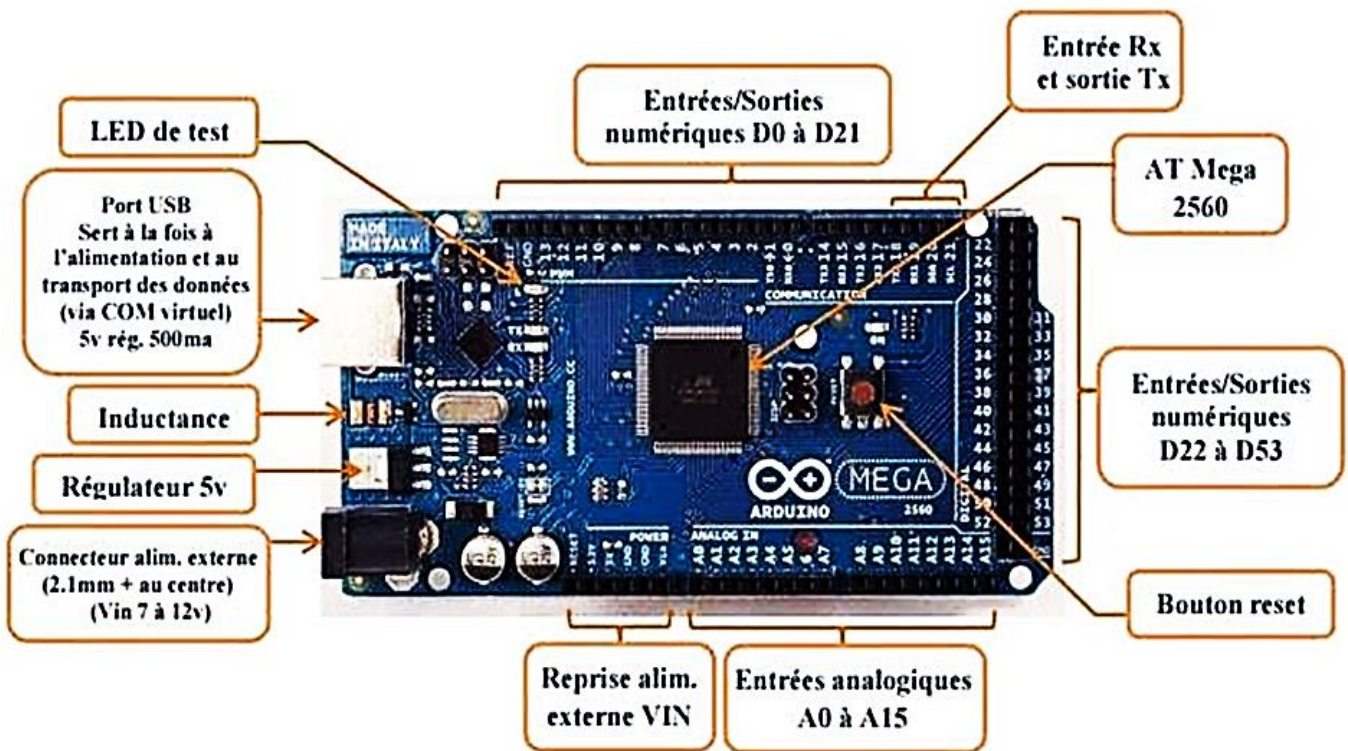


Figure 3.7 Les composants de la carte Arduino MEGA

Les caractéristiques de la carte Arduino MEGA 2560 :

La carte Arduino Mega 2560 est basée sur un ATmega2560 disposant de 54 E/S dont 14 PWM, 16 analogiques et 4 UARTs. Cette carte est idéale pour des applications exigeant des caractéristiques plus complètes que l'Uno.

En plus de ces caractéristiques, il existe d'autres caractéristiques d'Arduino MEGA 2560, tels que la gestion des interruptions, fiche USB B et 3 ports supplémentaires série. [33]

Tableau 3.1 Caractéristiques de la carte Arduino MEGA 2560 [33]

Caractéristiques	Carte Arduino MEGA 2560
Alimentation	via port USB ou 7 à 12 V sur connecteur alimentation
Microprocesseur	ATMega2560
Mémoire flash	256 KB
Mémoire SRAM	8 KB
Mémoire EEPROM	4 KB
Interfaces	54 broches d'E/S dont 14 PWM 16 entrées analogiques 10 bits
Intensité par broche E/S	40 mA
Cadencement	16 MHz
Les bus des données	I2C et SPI
Version	Rev 3
Dimensions	107 x 53 x 15 mm

Le microcontrôleur ATmega2560 :

Le microcontrôleur ATmega2560 est un composant électronique de la famille des microcontrôleurs AVR, fabriqué par Microchip Technology (anciennement Atmel). Il est largement utilisé dans le domaine de l'électronique et de l'automatisation en raison de sa puissance de traitement, de sa mémoire généreuse et de sa polyvalence. [34]

Atmega2560, couramment trouvé dans l'Arduino Mega 2560 en tant que microcontrôleur principal. Il s'agit d'un microcontrôleur basé sur AVR RISC qui exécute des instructions puissantes en un seul cycle d'horloge. Cela lui permet de trouver un juste équilibre entre la consommation d'énergie et la vitesse de traitement [35]. La figure montre un microcontrôleur ATmega 2560, qu'on trouve sur la carte Arduino Mega 2560 :



Figure 3.8 Le microcontrôleur ATmega2560

Voici une description plus détaillée des principales caractéristiques du microcontrôleur ATmega2560 (Voir Annexe A).

3.2.4 Périphériques :

1. Servomoteur – Arduino:

Description :

Un servomoteur est de manière général un moteur asservi en position, c'est à dire qu'un système mesure la position du moteur et le contrôle de telle sorte à ce qu'il atteigne la position demandée. Un vaste choix de dimensions et de puissances existe, mais nous allons nous concentrer sur le SG90 (ou sa version plus costaud le MG90). C'est en effet le servomoteur le plus abordable et celui communément trouvé au sein de modélisme ou de projet hobbyiste.

Son contrôle est d'une simplicité enfantine et c'est une solution rapide pour mettre en mouvement un mécanisme qui nécessite peu de couple et un positionnement précis dans une plage de 270°. [37]

Composition d'un servomoteur :

- Les fils, qui sont au nombre de trois
- L'axe de rotation sur lequel est monte un accessoire en plastique ou en métal
- Le boîtier qui le protège
- Un moteur à courant continu

- Des engrenages pour former un réducteur (en plastique ou en métal)
- Un capteur de position de l'angle d'orientation de l'axe (un potentiomètre bien souvent)
- Une carte électronique pour le contrôle de la position de l'axe et le pilotage du moteur à courant continu. [37]

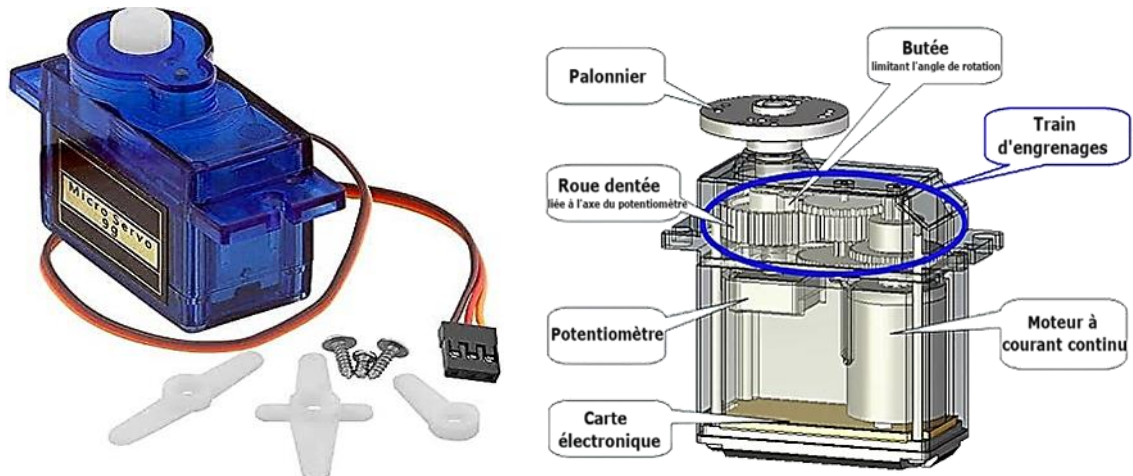


Figure 3.9 Servomoteur

Les types des servomoteurs :

Il existe divers types de servomoteurs selon la taille, poids et couple. On peut noter deux servomoteurs les plus utilisables :

Servomoteur SG90.

Servomoteur MG966 ou MG905.

Pour notre projet, nous utilisons le servomoteur SG90 :



Figure 3.10 Servomoteur SG90

Tableau 3-2 Caractéristiques du servomoteur SG90 [37]

Caractéristiques	Moteur pas à pas 28BYJ - 48
Dimensions	22 x 11.5 x 27mm
Poids	9g
Tension d'alimentation	4.8V à 6V
Vitesse	0.12 s / 60° sous 4.8V
Couple	1.2 Kg / cm sous 4.8V
Amplitude	De 0 à 180°

Les avantages d'un servomoteur :

- Facile d'utilisation.
- Positionnement plutôt précis pour un système aussi compact et bon marché.
- Prix bas. [37]

2. Capteur de proximité Infrarouge – FC 51 – Arduino :

Le capteur infrarouge peut être utilisé comme capteur de contact. On fait une mesure avec la led infrarouge éteinte et une avec la led infrarouge allumée. S'il n'y a aucun obstacle proche, la valeur lue est la même. Sinon, l'obstacle aura réfléchi la lumière infrarouge et la deuxième mesure donnera un résultat plus élevé. [38]

Le capteur de proximité FC51 utilise la lumière dans le domaine de l'infra-rouge (insensible à la luminosité ambiante) pour détecter la présence d'un objet sans contact. Il est composé d'un émetteur et d'un récepteur infra-rouge et a un fonctionnement de type reflex. [39]

Le capteur infrarouge peut aussi être utilisé en capteur de distance en mesurant l'angle avec lequel le rayon réfléchi arrive sur le récepteur. En fonction de la distance entre l'émetteur et le récepteur, on peut en déduire la distance de l'obstacle. [38]



Figure 3.11 Capteur de proximité Infrarouge – FC 51

Tableau 3.3 Caractéristiques du Capteur de proximité Infrarouge – FC 51 [39]

Caractéristiques	Capteur de proximité Infrarouge – FC 51
Distance de détection	De 2 à 30 cm
Alimentation	3.3 – 5 V
Consommation	20 mA max
Brochage	Equipé de 3 broches seules GND et S sont à câble
Dimensions	32 x 15 x 8 mm
Réglage	Deux résistances variables à un tour
Poids	2 g
OUT	Interface de sortie numérique de la carte (0 si détection, et 1 si aucune détection)

3. Ecran LCD – Arduino :

Les cristaux liquides (LCD) sont constitués de molécules complexes. Tout comme l'eau, ils changent leur état de solide à liquide, selon la température à laquelle ils sont exposés. Lorsqu'elles sont à l'état liquide, les molécules se déplacent mais sont susceptibles de former une ligne dans une certaine direction, leur permettant de réfléchir la lumière. Les cristaux sont disposés dans une matrice avec des groupes de trois cristaux de couleurs rouge, vert et bleu, formant un segment appelé pixel. [40]

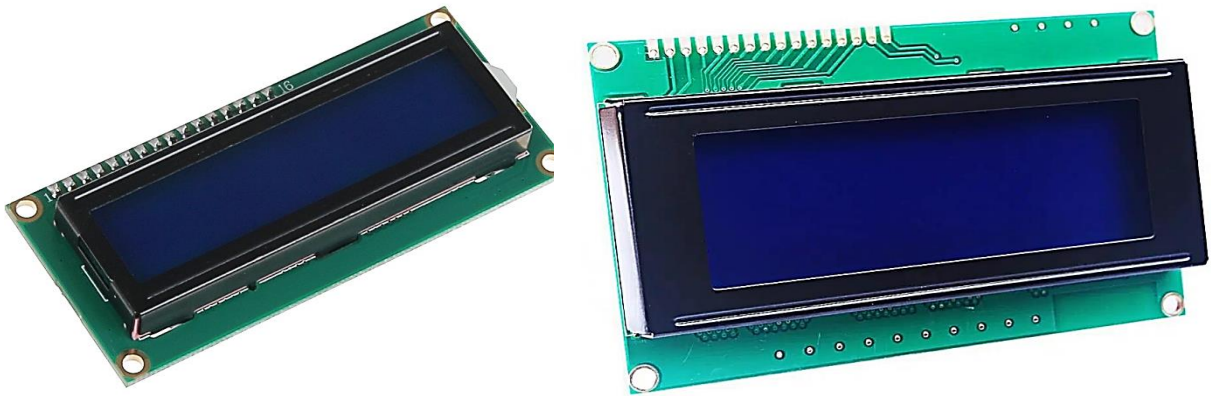


Figure 3.12 Ecran LCD

Tableau 3.4 Caractéristiques de l'Ecran LCD

Caractéristiques	Ecran LCD
Mode d'affichage	STN, BLUB
Format d'affichage	16 caractères x 2 lignes
Données d'entrée	4 ou 8 bits
Alimentation	5 V
Rétro-éclairage (coté)	LED (blanc)

La première information à connaître est le nombre de caractère affichable par ligne. Pour ce modèle, c'est 16 caractères sur deux lignes soit au total 32 caractères. On recueille également l'information du mode de transmission de données sur quatre (4) ou huit (8) bits.

En outre, les informations suivantes à connaître sont les tensions d'exploitations, caractéristiques électriques et mécaniques de l'afficheur LCD. [40]

4. BOUTON POUSSIOR 4 PATTES – Arduino :

Un Bouton Poussoir connecte deux points d'un circuit entre eux lorsqu'il est appuyé, il agit comme un interrupteur.

Souvent ils ont 4 pattes (comme sur l'image ci-dessous). Si c'est le cas, sachez que les broches sont reliées deux à deux. Cela signifie qu'elles fonctionnent par paire. Il faut donc se méfier lorsque vous le branchez sinon vous obtiendrez le même comportement qu'un fil (si vous connectez deux broches reliés). Utilisez un multimètre pour déterminer quelles broches sont distinctes. Pour ne pas se tromper, on utilise en général deux broches qui sont opposées sur la diagonale du bouton. [41]



Figure 3.13 Bouton Poussoir 4 Pattes

Fonctionnement de bouton poussoir :

Lorsque le bouton poussoir est ouvert (non appuyé), il n'y a pas de connexion entre les broches opposées du bouton poussoir, et donc la broche de la carte ARDUINO est connectée à la masse (au 0V) au travers de la résistance de "rappel-au-moins" et la broche sera lue à l'état BAS. Lorsque le bouton est fermé (appuyé), cela crée une connexion entre les broches opposées du bouton poussoir, et la broche de la carte ARDUINO est connectée au 5 volts, et elle sera lue à l'état HAUT. [41]

Remarque : quand le bouton est relâché, pour ne pas laisser flottante l'entrée du port (cas d'un interrupteur mécanique par exemple), il est nécessaire d'utiliser une résistance de rappel ou de tirage. [41]

5. Buzzer actif 5V :

Les buzzers sont un moyen simple, rapide et pratique d'ajouter une sortie audio à vos projets. Les alertes audios sont très efficaces pour attirer l'attention immédiatement, même à distance.

Cela les rend très utiles pour les alarmes et les avertissements d'urgence. Bien sûr, vous pouvez également décider de les utiliser pour des applications musicales ou autres.

Les buzzers actifs n'ont besoin que d'une source d'alimentation CC constante pour produire leurs sons à une fréquence particulière. [42]

Ces produits de buzzer sont plus fiables dans le travail et peuvent être installés facilement dans n'importe quel endroit. Ils sont aussi durables, antichoc et facile à utiliser. [43]



Figure 3.14 Buzzer actif

Tableau 3.5 Caractéristiques du Buzzer actif [43]

Caractéristiques	Buzzer actif
Tension nominale	5 V
Tension de fonctionnement	4 à 7 V
Sortie sonore	minimum 85db
Consommation de courant	maximum 25Ma
Température de stockage	-30 à +70 degrés Celsius

6. LED 5 mm :

Une LED est la conversion directe de l'énergie électrique en lumière. En simplifiant, l'architecture d'une diode électroluminescente se compose de trois parties :

- Une puce émettrice, le cristal semi-conducteur.
- Un réflecteur secondaire.
- Une lentille en plastique.

Le câblage minimal d'une LED consiste donc à lui ajouter une résistance en série afin de limiter le courant qui va la parcourir. [44]



Figure 3.15 LED diffusante 5mm

Tableau 3.6 Caractéristiques du LED 5mm [44]

Caractéristiques	LED
Boîtier	5 mm
Couleur d'émission + matériau	vert – rouge – bleu – jaune – blanc...
Longueur d'onde	565 nm
Intensité lumineuse (If = 20mA)	20mcd
Angle de vue	60

3.3 DESCRIPTION DE LA PARTIE LOGICIELLE :

Dans cette partie, on va présenter les différents logiciels qu'on a utilisé et dans la réalisation et la simulation pour notre application.

3.3.1 Logiciel IDE :

1. Généralité sur logiciel l'IDE:

Les créateurs d'Arduino ont développé un logiciel pour que la programmation des cartes Arduino soit visuelle, simple et complète à la fois.

C'est ce que l'on appelle une IDE, qui signifie Integrated Development Environment ou Environnement de Développement « Intégré » en français (donc EDI). L'IDE Arduino est le logiciel qui permet de programmer les cartes Arduino.

L'IDE affiche une fenêtre graphique qui contient un éditeur de texte et tous les outils nécessaires à l'activité de programmation. [45]

L'image suivante montre l'écran principal qui apparaît au lancement de l'IDE :

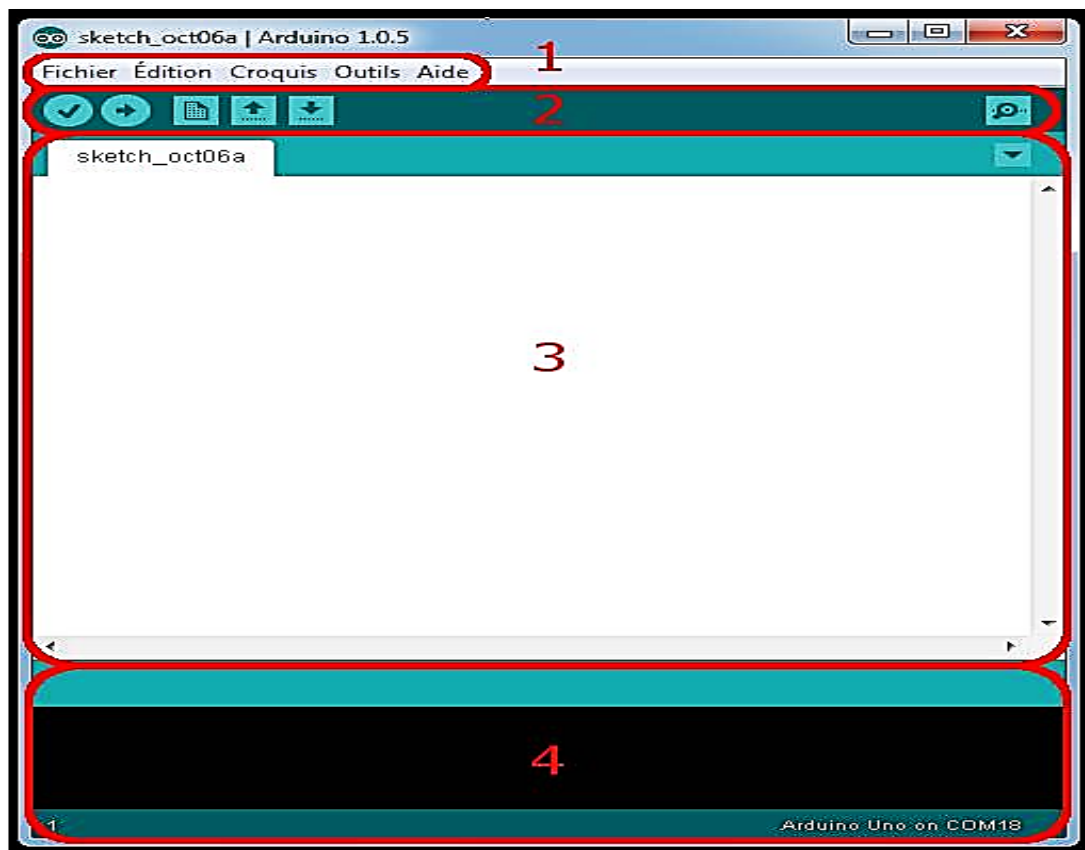


Figure 3.16 Ecran principal au lancement de l'IDE

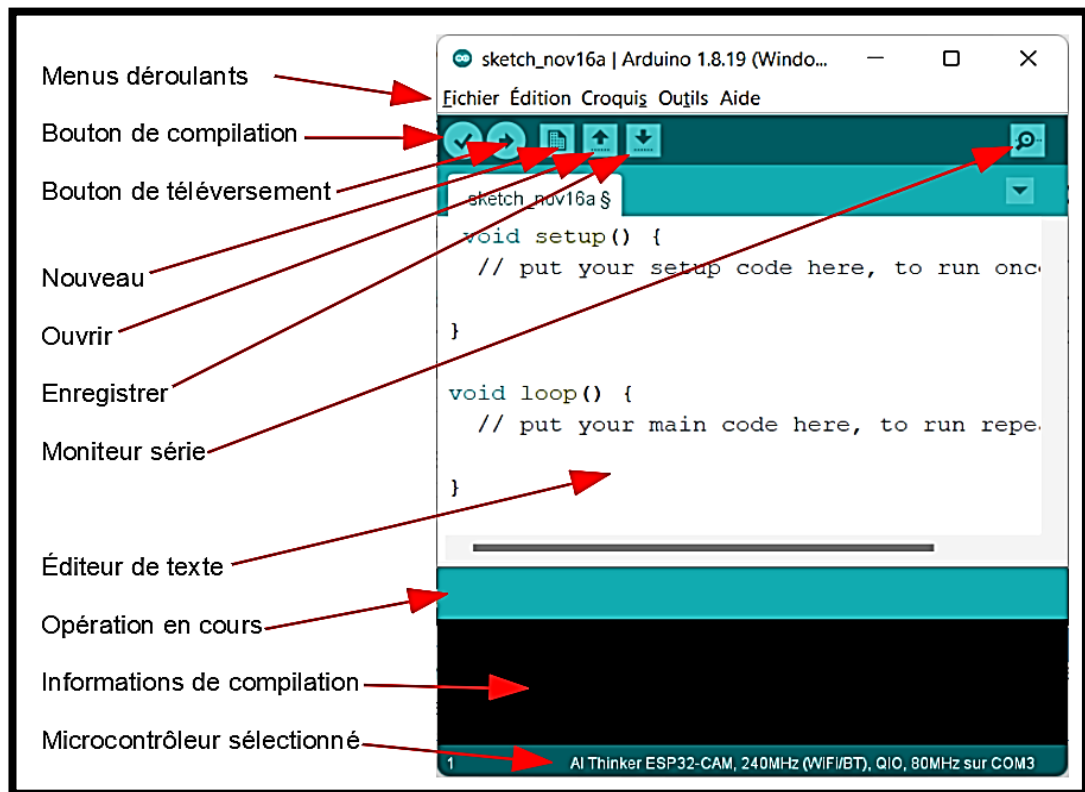


Figure 3.17 L'interface de l'IDE Arduino

L'image si-dessus (figure 3.20) montre toute la structure de l'interface de l'IDE Arduino et leurs différentes barres et fonctions :

2. Le langage de programmation :

Un langage de programmation est un langage permettant à un être humain d'écrire un ensemble d'instructions (code source) qui seront directement converties en langage machine grâce à un compilateur (c'est la compilation). L'exécution d'un programme Arduino s'effectue de manière séquentielle, c'est-à-dire que les instructions sont exécutées les unes à la suite des autres. Voyons plus en détail la structure d'un programme écrit en Arduino. [46]

3. Structure d'un programme :

Le langage utilisé avec Arduino est très proche du langage C ou C++. C'est notamment le cas pour la syntaxe et les types de variables. Cependant, ce langage présente quelques différences, principalement dans la structure d'un programme. [22]

Un programme Arduino se décompose en deux parties principales : une première partie appelée setup, et une seconde appelée loop. Ces deux parties sont représentatives de la manière de fonctionner d'un microcontrôleur et sont appelées dans cet ordre. En effet, un microcontrôleur est destiné à exécuter les instructions contenues dans sa mémoire de manière cyclique et infinie. Avant de s'exécuter, le microcontrôleur a besoin d'une phase de « configuration » de son environnement, notamment pour paramétrer des broches d'entrées/sorties ou tout simplement pour fixer des variables globales. Cette phase correspond à la fonction setup. [22]

L'image suivante montre toutes les fonctions qui permet de saisir le programme, l'enregistrer, le compiler, le vérifier, le transférer sur une carte Arduino...



Figure 3.18 Les fonctions d'un programme Arduino dans l'IDE

Un programme Arduino comporte trois parties :

- La partie déclaration des variables (optionnelle).
- La partie initialisation et configuration des entrées/sorties : la fonction `setup()`.
- La partie principale qui s'exécute en boucle : la fonction `loop()`.

Dans chaque partie d'un programme sont utilisées différentes instructions issues de la syntaxe du langage Arduino. [46]

Le code contenu dans la fonction `setup()` sera exécuté une seule fois au lancement du programme. Elle sert généralement à l'initialisation des variables et des états du microcontrôleur.

La fonction `loop()` est la fonction principale du programme. Elle tourne en boucle après l'exécution de la fonction `setup()` et ce jusqu'à la mise hors tension du microcontrôleur.

C'est le minimum nécessaire pour que le code Arduino puisse compiler.

Par la suite des variables, des paramètres, des constantes, des sous-fonctions ainsi que des bibliothèques peuvent être ajoutés. [46]

```

programmerArduinoExemple | Arduino 0022
File Edit Sketch Tools Help
programmerArduinoExemple

1 int brocheCapteur = A0; // selection de la broche sur laquelle est connectée le capteur
  int brocheLED = 13; // selection de la broche sur laquelle est connectée la LED
  int valeurCapteur = 0; // variable stockant la valeur du signal reçu du capteur

2 void setup() {
  // broche de la LED configurée en sortie
  pinMode(ledPin, OUTPUT);
}

3 void loop() {
  // lecture du signal du capteur
  valeurCapteur = analogRead(brocheCapteur);
  // allume la LED
  digitalWrite(brocheLED, HIGH);
  // delai de "valeurCapteur" millisecondes
  delay(valeurCapteur);
  // éteint la LED
  digitalWrite(brocheLED, LOW);
  // delai de "valeurCapteur" millisecondes
  delay(valeurCapteur);
}

```

Figure 3.19 Les parties d'un programme Arduino [46]

4. Injection du programme :

Avant de transférer un programme sur la carte Arduino, il est nécessaire de sélectionner le type de la carte (Arduino MEGA) et le numéro de port USB (COM) comme l'exemple dans les figures suivantes. [22]

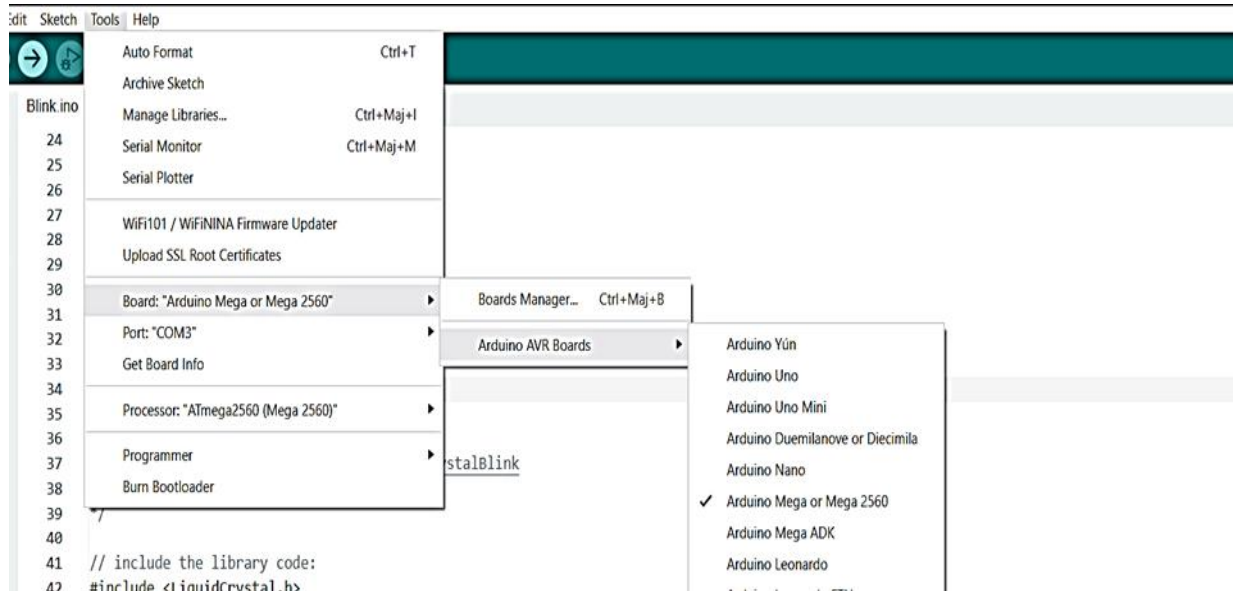


Figure 3.20 Paramétrage de la carte étape 1

Après de finir le paramétrage de la carte Arduino, on peut transférer le programme sur la carte pour l'exécuter.

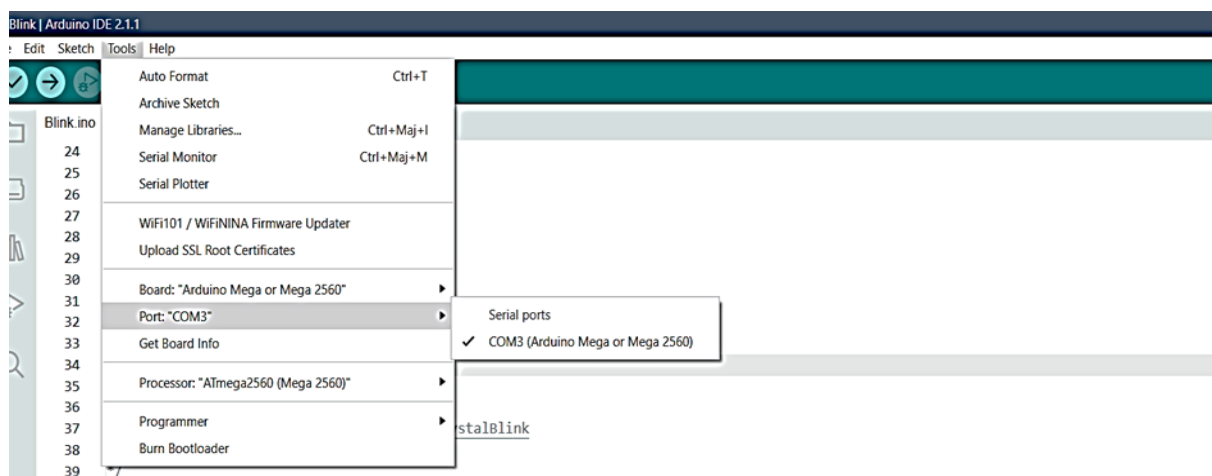


Figure 3.21 Paramétrage de la carte étape 2

3.3.2 Logiciel Proteus V8 :

Proteus est une suite logicielle destinée à l'électronique. Développé par la société Labcenter Electronics, les logiciels inclus dans Proteus permettent la CAO dans le domaine électronique. Deux logiciels principaux composent cette suite logicielle : ISIS, ARES, PROSPICE et VSM. [47]

1. Présentation générale:

Cette suite logicielle est très connue dans le domaine de l'électronique. De nombreuses entreprises et organismes de formation (incluant lycée et université) utilisent cette suite logicielle. Outre la popularité de l'outil, Proteus possède d'autres avantages : [47]

- Pack contenant des logiciels facile et rapide à comprendre et utiliser.
- Le support technique est performant.
- L'outil de création de prototype virtuel permet de réduire les coûts matériel et logiciel lors de la conception d'un projet.

2. Interface Proteus V8 :

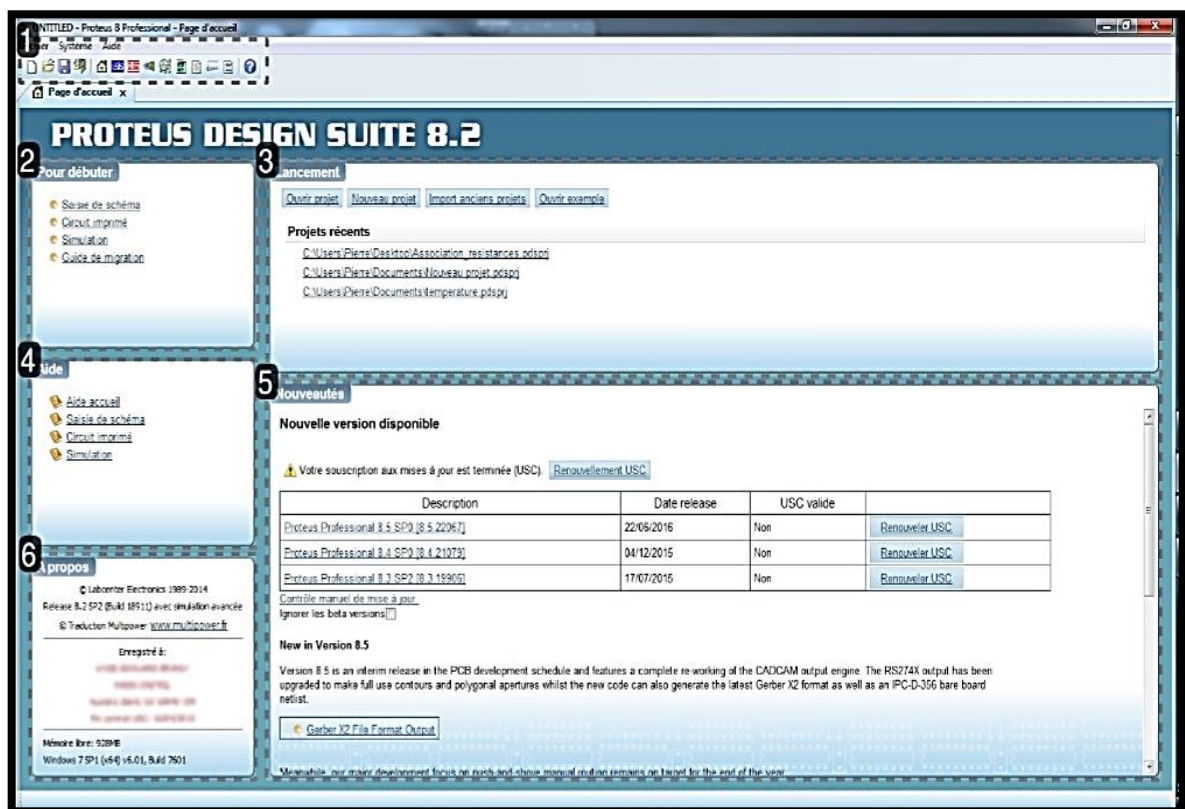


Figure 3.22 L'écran principal au lancement de Proteus [48]

3.3.3 Logiciel Micrografx Flowcharter 7:

Présentation générale:

FlowChart est un type de diagramme qui représente un flux de travail ou un processus, montrant les étapes sous forme de cases de différents types et leur ordre en les reliant avec des flèches. [49]

Types d'organigrammes : les organigrammes peuvent être modélisés (tels que les gestionnaires, les analystes système et les employés) :

- Organigrammes de documents, montrant les contrôles sur un flux de documents à travers un système.
- Organigrammes de données, montrant les contrôles sur un flux de données dans un système.
- Organigrammes du système montrant les contrôles au niveau physique ou des ressources.
- Organigramme du programme, montrant les commandes d'un programme au sein d'un système.

Avantages de l'organigramme :

- Communication : Les organigrammes sont un meilleur moyen de communiquer la logique d'un système à toutes les personnes concernées.
- Analyse efficace : À l'aide d'un organigramme, le problème peut être analysé de manière plus efficace.
- Documentation appropriée : Les organigrammes du programme constituent une bonne documentation du programme, nécessaire à diverses fins. [49]

Avantages de l'utilisation de l'organigramme :

- Codage efficace : Les organigrammes servent de guide ou de modèle pendant la phase d'analyse des systèmes et de développement du programme.
- Débogage approprié : L'organigramme facilite le processus de débogage.
- Maintenance efficace du programme : La maintenance du programme d'exploitation devient facile à l'aide d'un organigramme.

Cela aide le programmeur à concentrer ses efforts plus efficacement sur cette partie

Inconvénients de l'utilisation de Flowchart :

- Logique complexe : Parfois, la logique du programme est assez compliquée. Dans ce cas, l'organigramme devient complexe et maladroit.
- Altérations et modifications : Si des modifications sont nécessaires, l'organigramme peut nécessiter un redessin complet.
- Reproduction : Comme les symboles de l'organigramme ne peuvent pas être saisis, la reproduction de l'organigramme devient un problème. [49]

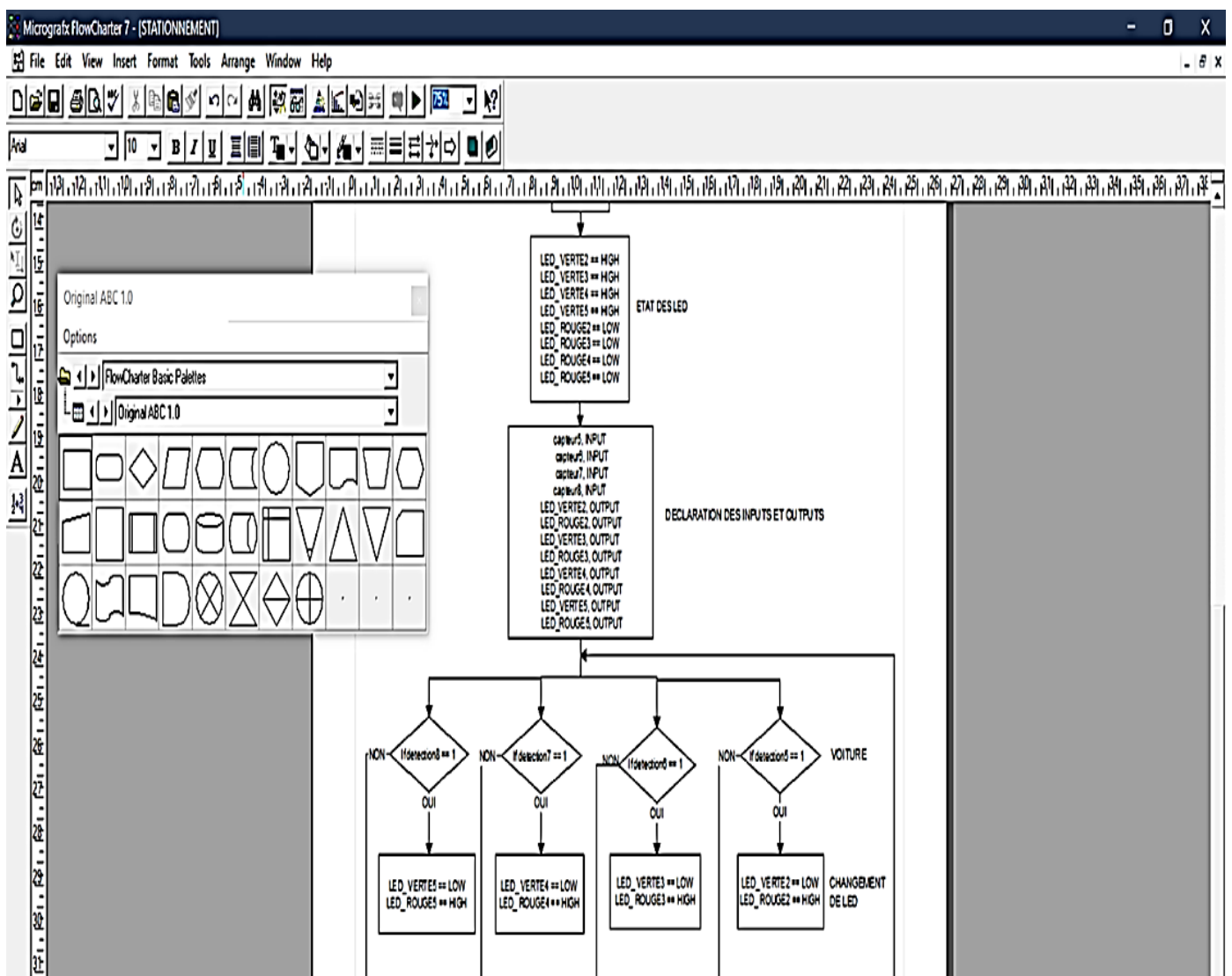


Figure 3.23 Interface principale de FlowCharter 7

3.4 CONCLUSION :

L'Arduino est plus qu'un simple matériel, mais c'est aussi un logiciel (mélange d'électronique et programmation) qui joue un rôle important dans la réalisation de beaucoup projets (mini robot, système d'arrosage, gestion de parking...).

Chapitre 4: Simulation et Application

INTRODUCTION :

Le but de cette partie est de réaliser une application '*d'une barrière automatique pour gérer un parking à base d'Arduino Mega 2560*', et l'exécuter sur une maquette réelle.

Notre maquette fonctionne automatiquement et connecte directement sur les portes de la carte Arduino Mega 2560 à l'aide des câbles de branchement.

Premièrement, nous commençons par donner les différents composants électroniques utilisés dans la maquette à côté d'Arduino, après nous présentons les organigrammes fonctionnels de chaque partie dans ce projet et nous expliquons le fonctionnement de ces parties. Tous cela en créant un programme spécial envoyé sur la carte Arduino pour laisser le parking en marche.

Enfin, nous terminons cette partie avec le schéma fonctionnel sur le logiciel Proteus et citons les différentes étapes de la gestion de parking pour notre maquette.

4.1 DIFFERENTES COMPOSANTS DU PROJET REALISE :

La Maquette parking comprend:

- Pour l'Entrée, il y a :
 - Une barrière mécanique.
 - Un servomoteur SG90.
 - Deux capteurs de proximité infrarouge FC 51(1'un avant la barrière et l'autre après la barrière).
- Pour la Sortie, il y a :
 - Une deuxième barrière mécanique.
 - Un deuxième servomoteur SG90.
 - Deux capteurs de proximité infrarouge FC 51(1'un avant la barrière et l'autre après la barrière).

- Ecran LCD pour afficher tous les informations (le comptage des entrées et sorties d'un parking pour la gestion des places libres).
- Le parking contient 7 places, 4 places pour le parking couvert et 3 places pour le parking ouvert.
- Les 4 places couvertes contiennent un led rouge, un led vert + un capteur de proximité infrarouge FC 51 pour chaque place.
- Module de feux de circulation LED, un buzzer actif.
- 3 boutons poussoir (un pour l'ouverture de la barrière, un pour la fermeture de la barrière, un pour arrêter les barrières).

Tous ces composants sont identifiables et programmables sur le microcontrôleur de l'Arduino.

Notre maquette se fonctionne d'une manière automatique grâce à l'Arduino qui gère toutes les fonctions :

- Gestion d'ENTRÉE.
- Gestion de SORTIE.
- Gestion de STATIONNEMENT.

REMARQUE : La connexion des composants avec la carte Arduino (voir Annexe A).

4.2 ORGANIGRAMME ET CODE DES COMPOSANTS AVEC L'ARDUINO MEGA :

4.2.1 Organigramme de gestion d'Entrée :

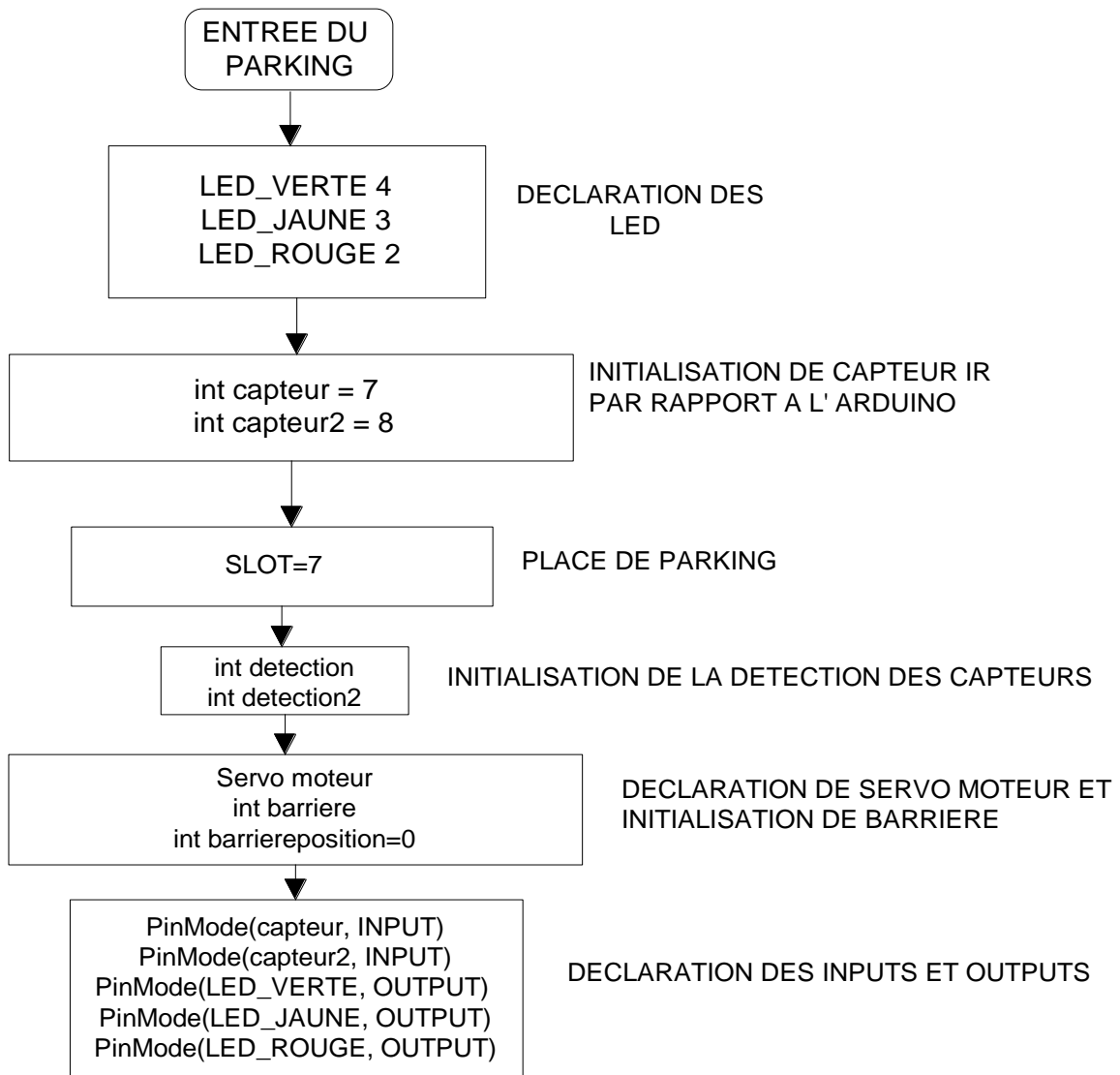


Figure 4.1 Organigramme de déclaration et initialisation des composants(entrée)

Cette partie précédente représente la déclaration et initialisation des composants par rapport à l'Arduino et selon la gestion de logiciel de programmation. (Entrée)

Dans la partie suivante, on va présenter le fonctionnement de la gestion d'entrée :

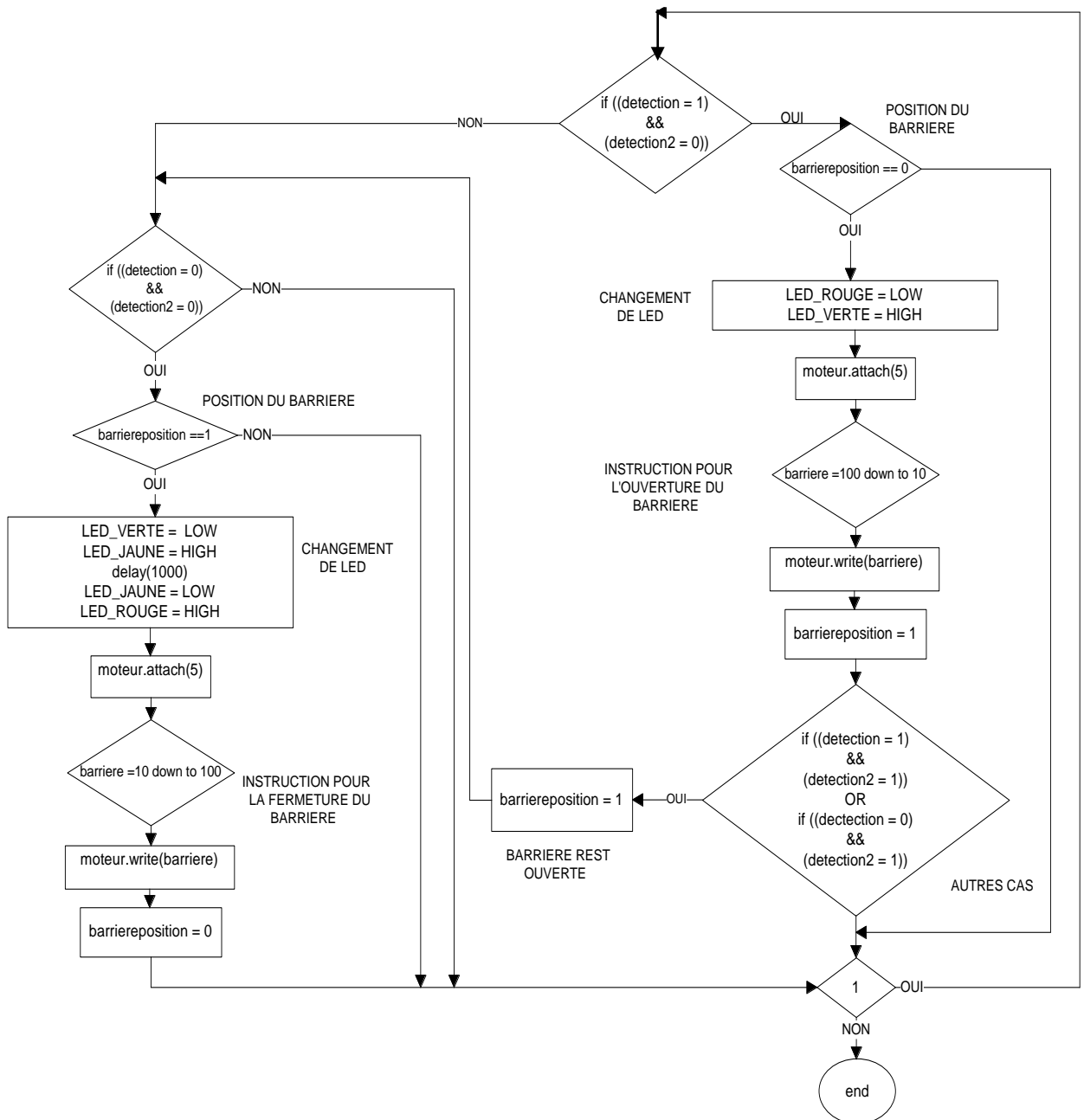


Figure 4.2 Organigramme fonctionnel d'entrée du parking

4.2.2 Le code d'Entrée avec l'Arduino (composants d'Entrée) :

```
#include <Servo.h>
#define LED_VERT 4
#define LED_JAUNE 3
#define LED_ROUGE 2
Servo moteur;
int barriere;
int capteur=7;
int capteur2=8;
```

```
int detection;
int detection2;
int barriereposition=0;

void setup(){
  pinMode(capteur, INPUT);
  pinMode(capteur2, INPUT);
  pinMode(LED_VERTE, OUTPUT);
  pinMode(LED_JAUNE, OUTPUT);
  pinMode(LED_ROUGE, OUTPUT);
  digitalWrite(LED_ROUGE, HIGH);}

void loop(){
  detection = digitalRead(capteur);
  detection2 = digitalRead(capteur2);
  if ((detection == 1) && (detection2 == 1)){ // rien
    fermer();
    delay(1000);}
  else { // voiture
    ouvrir();
    delay(1000);}
}

void ouvrir(){
  if (barriereposition == 0){
    digitalWrite(LED_ROUGE, LOW);
    digitalWrite(LED_VERTE, HIGH);
    delay(500);
    moteur.attach(5);}
  for (barriere = 100; barriere >=10; barriere -=1)
    {moteur.write(barriere);
    delay(20);}
  barriereposition = 1;
  moteur.detach();
}

void fermer(){
  if (barriereposition == 1){
    digitalWrite(LED_VERTE, LOW);
    digitalWrite(LED_JAUNE, HIGH);
    delay(1000);
    digitalWrite(LED_JAUNE, LOW);
    digitalWrite(LED_ROUGE, HIGH);
    moteur.attach(5);}
  for (barriere = 10; barriere <=100; barriere +=1)
    {moteur.write(barriere);
    delay(20);}
  barriereposition = 0;
  moteur.detach();
}
```

4.2.3 Le fonctionnement de la gestion d'Entrée :

- Premièrement, la barrière doit être en position fermée par défaut.
- Le led rouge de la barrière doit être en position allumé par défaut lorsque la barrière est fermée.
- Le mode de fonctionnement de la barrière doit être automatique par défaut.
- L'afficheur doit être éteint par défaut.

Pour l'entrée :

- Voiture approche, le premier capteur détecte la présence de la voiture.
- L'afficheur affiche un message " bienvenue "
- Lire le badge RFID :

Si le badge est validé correctement : l'afficheur affichera "access acceptable "

Si le badge est refusé : l'afficheur affichera " access denied "

- Pour l'ouverture de barrière nous avons besoin d'accès de deux informations vrai (1), lesquels information vrai de capteur (cap1=1) et la lecture de badge est validé (lec=1), après la barrière s'ouvre.
- La voiture marche vers le deuxième capteur :

Le deuxième capteur détecte la présence de voiture (cap2=1), l'afficheur décrémente une place de parking (7 => 6).

Une fois la voiture dépasse le deuxième capteur (cap2=0), la barrière se ferme automatiquement.

Cette opération se répète jusqu'à le parking de stationnement soit complet (parking full, nombre des places disponibles nulle).

4.2.4 Organigramme de gestion de Sortie :

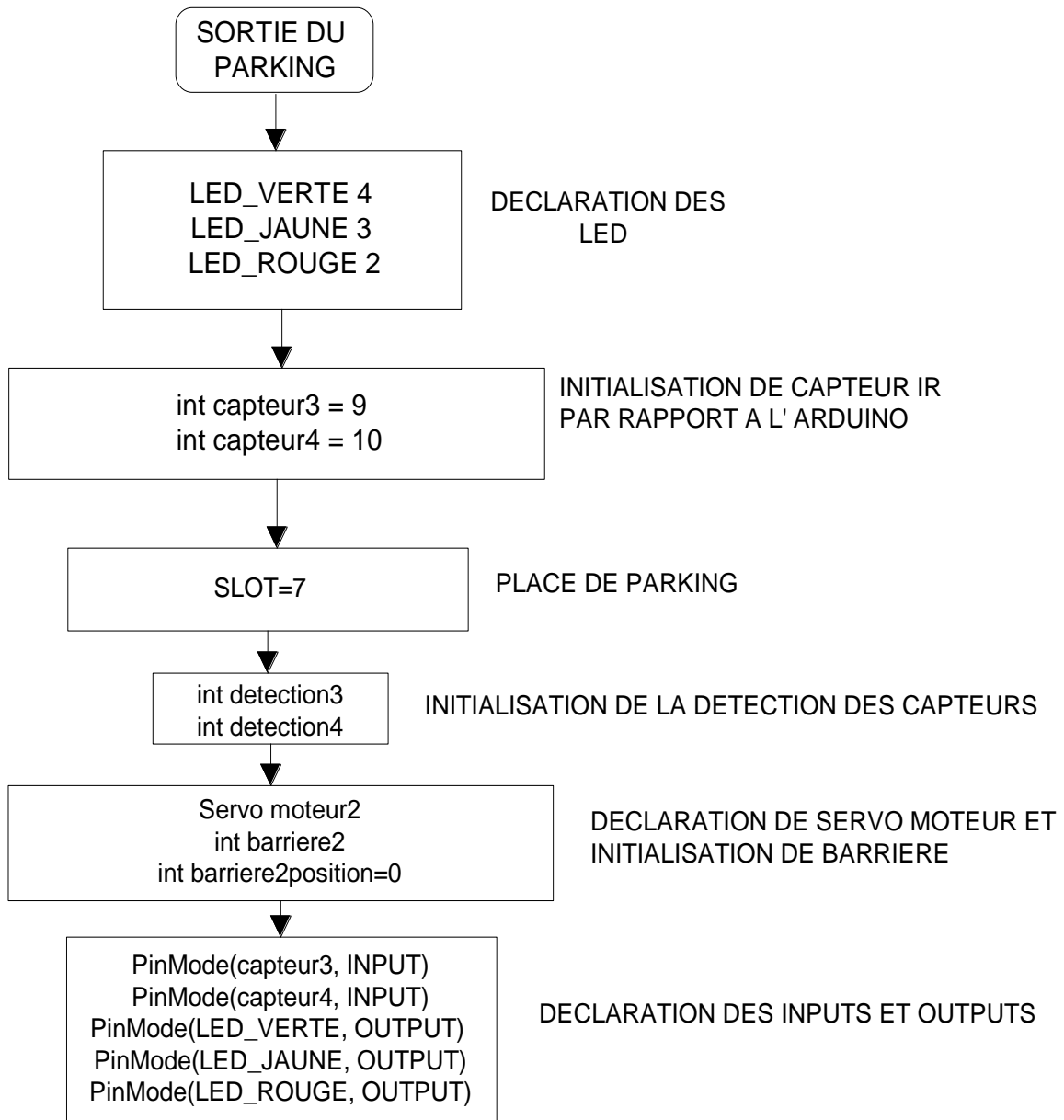


Figure 4.3 Organigramme de déclaration et initialisation des composants(sortie)

Cette partie précédente représente la déclaration et initialisation des composants par rapport à l'Arduino et selon la gestion de logiciel de programmation. (Sortie)

Dans la partie suivante, on va présenter le fonctionnement de la gestion de sortie :

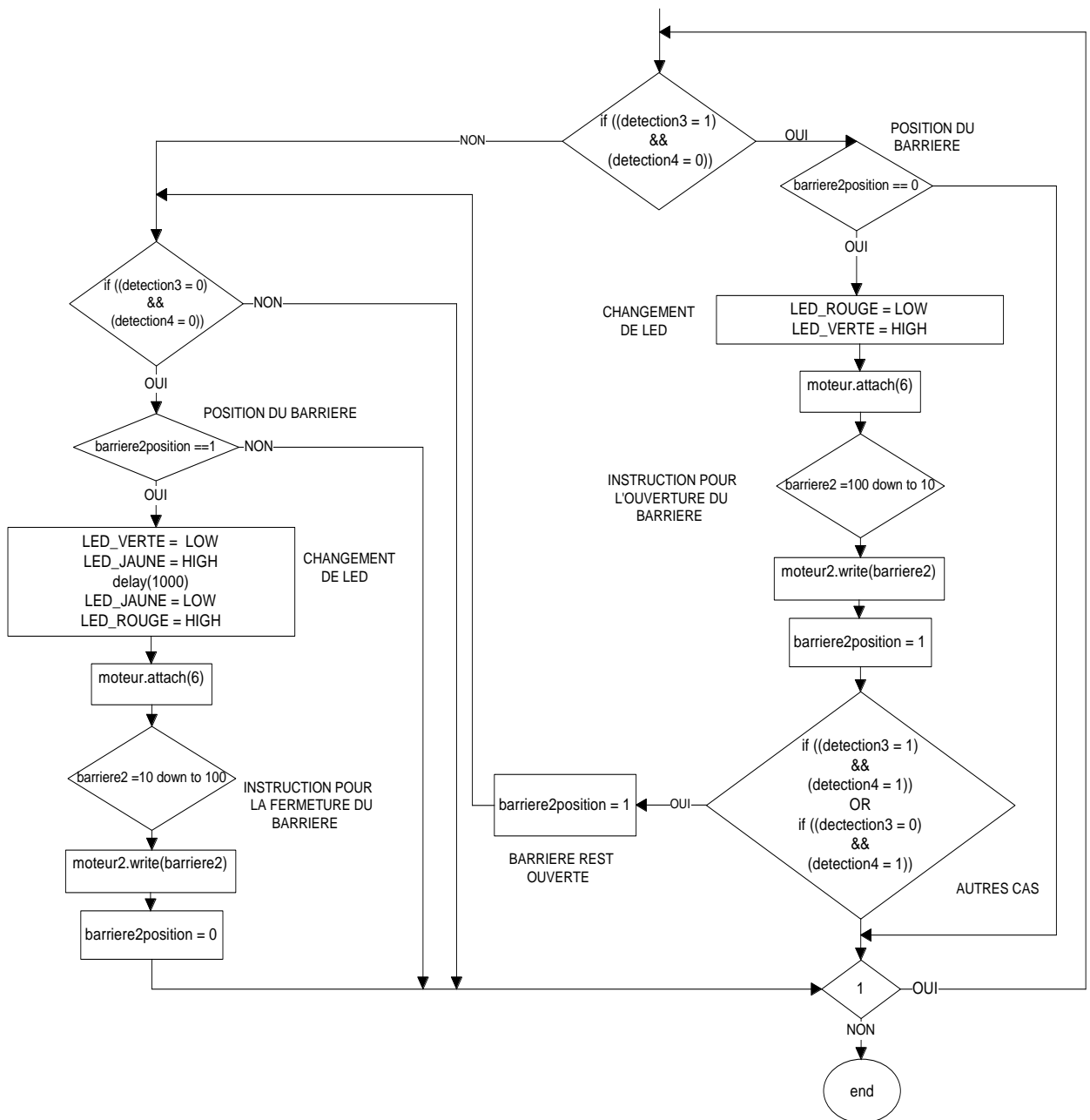


Figure 4.4 Organigramme fonctionnel de sortie du parking

4.2.5 Le code de Sortie avec l'Arduino (composants de sortie) :

```
#include <Servo.h>
#define LED_VERT 4
#define LED_JAUNE 3
#define LED_ROUGE 2
Servo moteur2;
int barriere2;
int capteur3=9;
int capteur4=10;
```

```
int detection3;
int detection4;
int barriere2position=0;
void setup(){
  pinMode(capteur3, INPUT);
  pinMode(capteur4, INPUT);
  pinMode(LED_VERTE, OUTPUT);
  pinMode(LED_JAUNE, OUTPUT);
  pinMode(LED_ROUGE, OUTPUT);
  digitalWrite(LED_ROUGE, HIGH);}

void loop(){
  detection3 = digitalRead(capteur3);
  detection4 = digitalRead(capteur4);
  if ((detection3 == 1) && (detection4 == 1)){ // rien
    fermer2();
    delay(1000);}
  else { // voiture
    ouvrir2();
    delay(1000);}
}

void ouvrir2(){
  if (barriere2position == 0){
    digitalWrite(LED_ROUGE, LOW);
    digitalWrite(LED_VERTE, HIGH);
    delay(500);
    moteur2.attach(6);}
  for (barriere2 = 100; barriere2 >=10; barriere2 -=1)
    {moteur2.write(barriere2);
    delay(20);}
  barriere2position = 1;
  moteur2.detach();
}

void fermer2(){
  if (barriere2position == 1){
    digitalWrite(LED_VERTE, LOW);
    digitalWrite(LED_JAUNE, HIGH);
    delay(1000);
    digitalWrite(LED_JAUNE, LOW);
    digitalWrite(LED_ROUGE, HIGH);
    moteur2.attach(6);}
  for (barriere2 = 10; barriere2 <=100; barriere2 +=1)
    {moteur2.write(barriere2);
    delay(20);}
  barriere2position = 0;
  moteur2.detach();
}
```

4.2.6 Le fonctionnement de la gestion de Sortie :

Pour la sortie :

- Voiture approche, le premier capteur de sortie détecte la présence de la voiture.
- L'afficheur affiche un message " merci pour votre visite "
- Pour l'ouverture de barrière nous avons besoin d'accès d'une seule information vrai (1), laquelle information vrai de capteur (cap3=1), après la barrière s'ouvre.
- La voiture marche vers le deuxième capteur de sortie :

Le deuxième capteur détecte la présence de voiture (cap4=1), l'afficheur augmente une place de parking (6 => 7).

Une fois la voiture dépasse le deuxième capteur (cap4=0), la barrière se ferme automatiquement. Cette opération répète au moins tant qu'il existe une seule voiture dans le parking.

- **Autres fonctions pour l'entrée et la sortie :**

Pour les positions de barrière (entrée et sortie) :

- Si la barrière est fermée, un led rouge s'allume.
- Si la barrière est ouverte, un led vert s'allume.
- Si la barrière est en cours d'ouverture ou de fermeture, un led jaune s'allume.

Pour le mode manuel (seulement l'entrée) :

- La lecture de badge sera éliminée dans le circuit.
- Pour l'ouverture de la barrière, on appuie sur le bouton poussoir « Open ».
- Pour la fermeture de la barrière, on appuie sur le bouton poussoir « Close ».
- Pour l'arrêt des barrières, on appuie sur le bouton poussoir « Off ». (Entrée et sortie)

4.2.7 Organigramme de gestion de Stationnement :

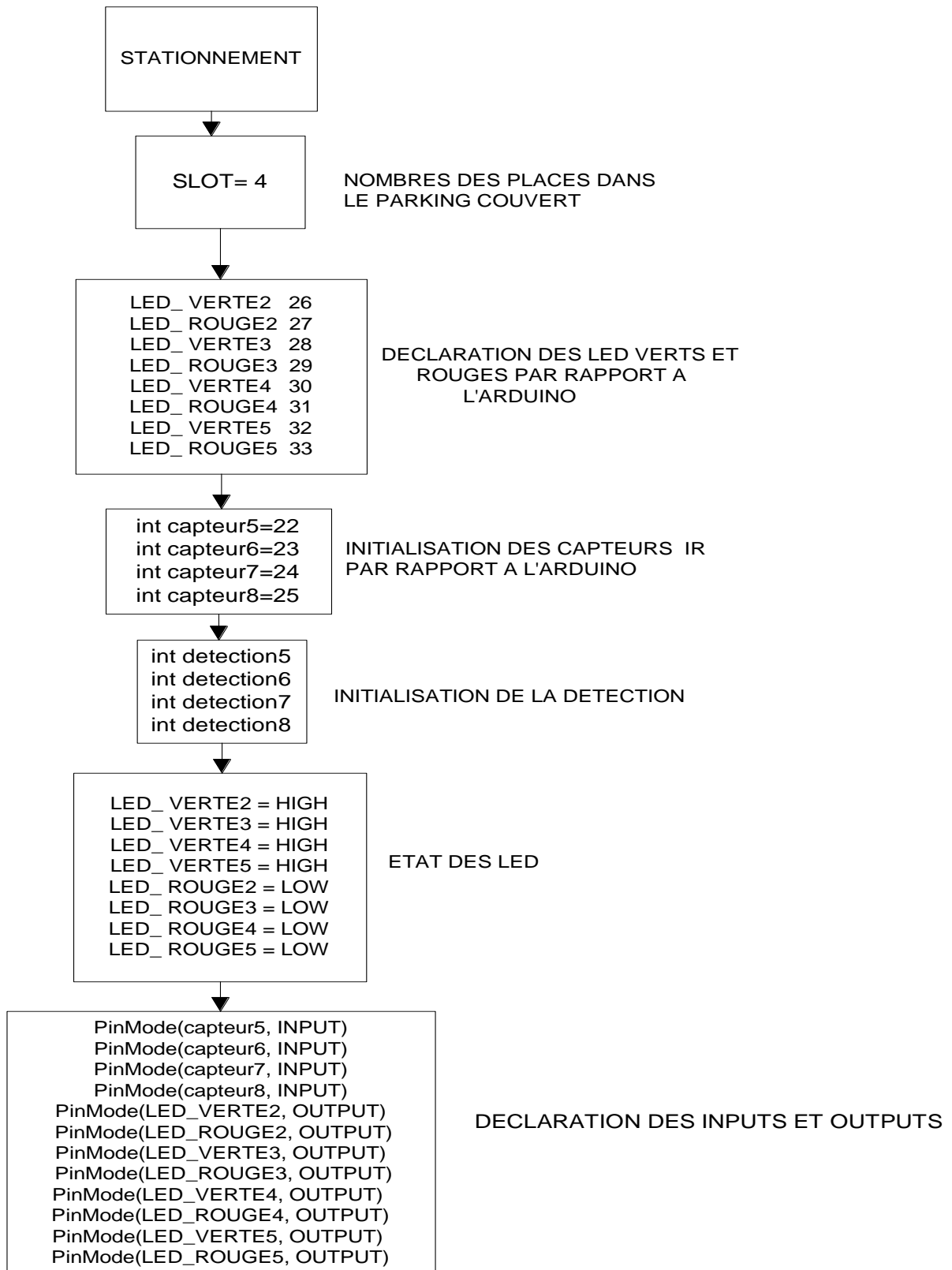


Figure 4.5 Organigramme de déclaration et initialisation des composants

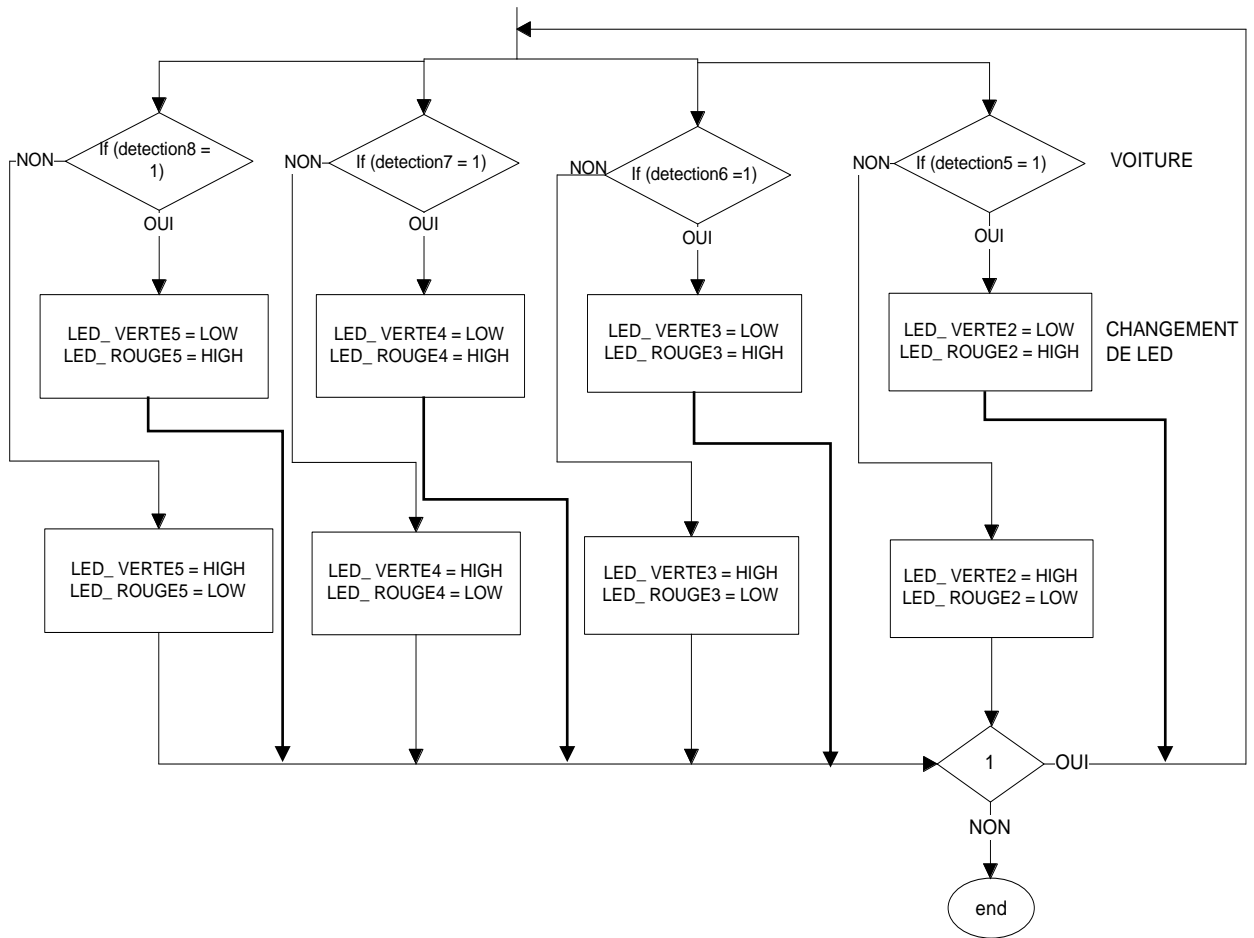


Figure 4.6 Organigramme fonctionnel de stationnement(couvert)

Cette partie représente la déclaration et initialisation des composants par rapport à l'Arduino et selon la gestion de logiciel de programmation. Après représente le fonctionnement de la gestion de stationnement.

4.2.8 Le code de Stationnement avec l'Arduino (composants de stationnement) :

```
int LED1 = 26;
int LED11 = 27;
int LED2 = 28;
int LED22 = 29;
int LED3 = 30;
int LED33 = 31;
int LED4 = 32;
int LED44 = 33;
int capteur5 = 22;
int capteur6 = 23;
int capteur7 = 24;
int capteur8 = 25;
int detection5 = HIGH;
int detection6 = HIGH;
int detection7 = HIGH;
int detection8 = HIGH;
void setup() {
pinMode(LED1, OUTPUT);
pinMode(LED11, OUTPUT);
pinMode(LED2, OUTPUT);
pinMode(LED22, OUTPUT);
pinMode(LED3, OUTPUT);
pinMode(LED33, OUTPUT);
pinMode(LED4, OUTPUT);
pinMode(LED44, OUTPUT);
pinMode(capteur5, INPUT);
pinMode(capteur6, INPUT);
pinMode(capteur7, INPUT);
pinMode(capteur8, INPUT);
pinMode(LED1, HIGH);
pinMode(LED2, HIGH);
pinMode(LED3, HIGH);
pinMode(LED4, HIGH);
Serial.begin(9600);
}

void loop() {
detection5 = digitalRead(capteur5);
if (detection5 == HIGH) {
digitalWrite(LED1, HIGH);
digitalWrite(LED11, LOW);
delay(2000);
} else {
digitalWrite(LED1, LOW);
digitalWrite(LED11, HIGH);
delay(2000);
}
```

```

delay(2000);
detection6 = digitalRead(capteur6);
if (detection6 == HIGH) {
  digitalWrite(LED2, HIGH);
  digitalWrite(LED22, LOW);
  delay(2000);
} else {
  digitalWrite(LED2, LOW);
  digitalWrite(LED22, HIGH);
  delay(2000);
}
delay(2000);
detection7 = digitalRead(capteur7);
if (detection7 == HIGH) {
  digitalWrite(LED3, HIGH);
  digitalWrite(LED33, LOW);
  delay(2000);
} else {
  digitalWrite(LED3, LOW);
  digitalWrite(LED33, HIGH);
  delay(2000);
}
delay(2000);
detection8 = digitalRead(capteur8);
if (detection8 == HIGH) {
  digitalWrite(LED4, HIGH);
  digitalWrite(LED44, LOW);
  delay(2000);
} else {
  digitalWrite(LED4, LOW);
  digitalWrite(LED44, HIGH);
  delay(2000);
}
delay(2000);
}

```

4.2.9 Le fonctionnement de la gestion de Stationnement :

Pour le parking couvert:

- Si la voiture entre dans une place de parking, le capteur détecte cette voiture (cap5=1), donc la voiture stationne dans cette place :

Le led vert s'éteint et le led rouge s'allume.

- Si la voiture quitte sa place, le capteur détecte l'absence de la voiture (cap5=0) :

Le led rouge s'éteint et le led vert s'allume.

Cette opération est applicable pour tous les places de parking couvert (4 places).

4.3 LE ROLE PRINCIPALE DES COMPOSANTS DANS NOTRE MAQUETTE :

Tableau 4-1 Rôle des composants d'entrée

		Les composants	Le rôle principale
ENTREE	1.	Capteur de proximité IR FC 51	La détection de la présence ou l'absence de véhicule
	2.	Servomoteur SG 90	Responsable sur l'ouverture et la fermeture du barrière (lisse)
	3.	Afficheur LCD_I2C	Afficher des messages, incrémenter et décrémenter les places dans le parking, afficher le nombre des places disponibles
	4.	Module de feux de circulation LED	Gérer les positions de la barrière
	5.	Bouton poussoir	Commande d'ouverture ou fermeture manuelle

Tableau 4-2 Rôle des composants de stationnement

		Les composants	Le rôle principale
STATIONNEMENT (Parking couvert)	1.	Capteur de proximité IR FC 51	La détection de la présence ou l'absence de véhicule
	2.	LED vertes LED rouges	Gérer la disponibilité des places

Tableau 4-3 Rôle des composants de sortie

		Les composants	Le rôle principale
SORTIE	1.	Capteur de proximité IR FC 51	La détection de la présence ou l'absence de véhicule
	2.	Servomoteur SG 90	Responsable sur l'ouverture et la fermeture du barrière (lisse)
	3.	Afficheur LCD_I2C	Afficher des messages, incrémenter et décrétement les places dans le parking, afficher le nombre des places disponibles
	4.	Module de feux de circulation LED	Gérer les positions de la barrière

4.4 SCHEMA FONCTIONNEL DU GESTIONNAIRE DE PARKING (PROTEUS):

Les véhicules sont détectés en entrée et en sortie. En entrée l'autorisation de passage a lieu si un véhicule est présent et s'il reste au moins une place dans le parking. Un niveau logique (Tension 5V) est alors envoyé à la commande de la barrière, le nombre de places disponibles est décompté. En sortie, lors de la détection du passage d'un véhicule le nombre de places dans le parking est incrémenté sans dépasser le nombre maximum de places du parking (Max). Une initialisation est possible lorsque le parking est vide, le nombre de places disponibles est alors égal au maximum. Le nombre des places disponibles (7 ou moins dans notre cas) est affiché en permanence sur l'afficheur LCD. [22]

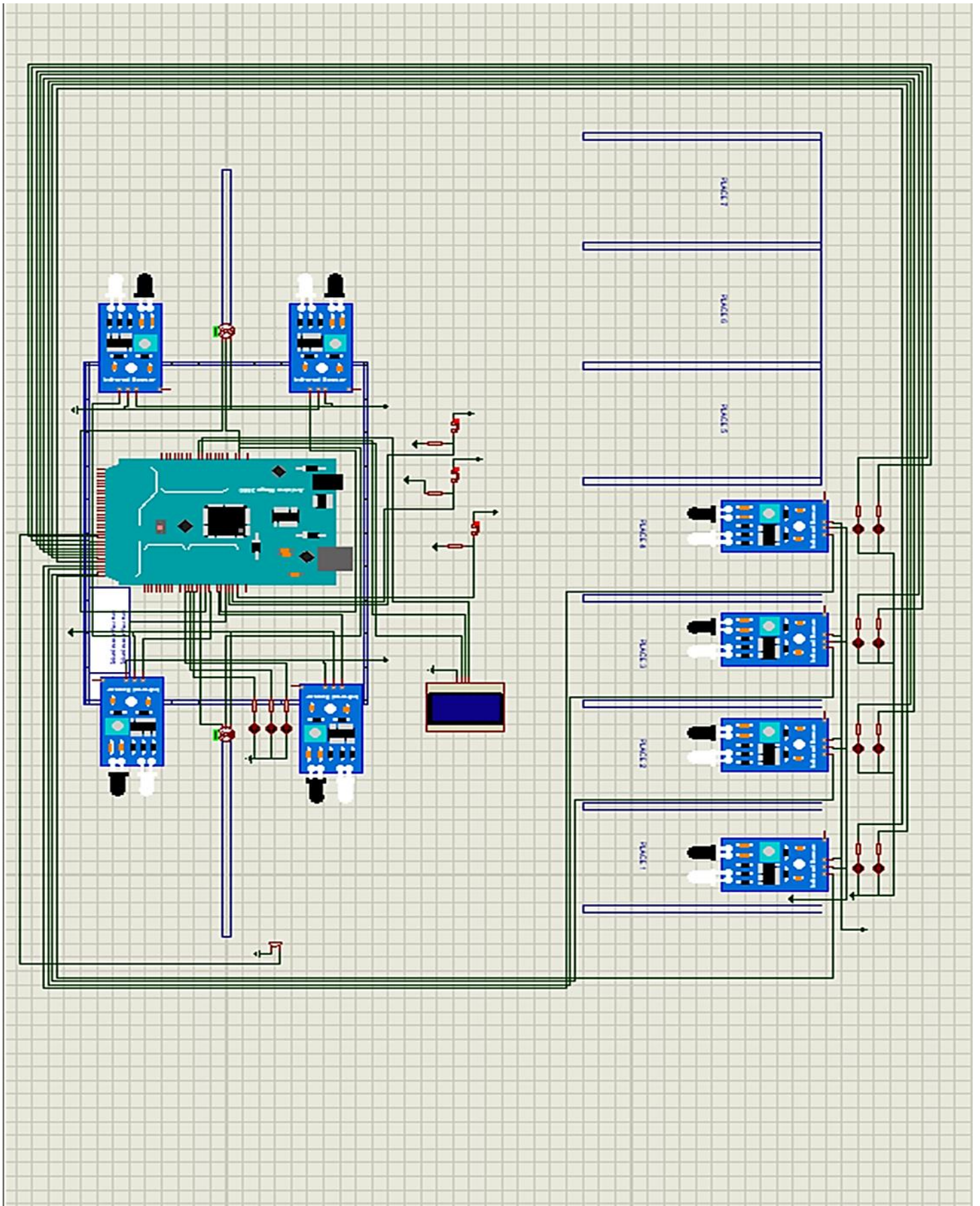


Figure 4.7 Schéma fonctionnel Proteus de parking

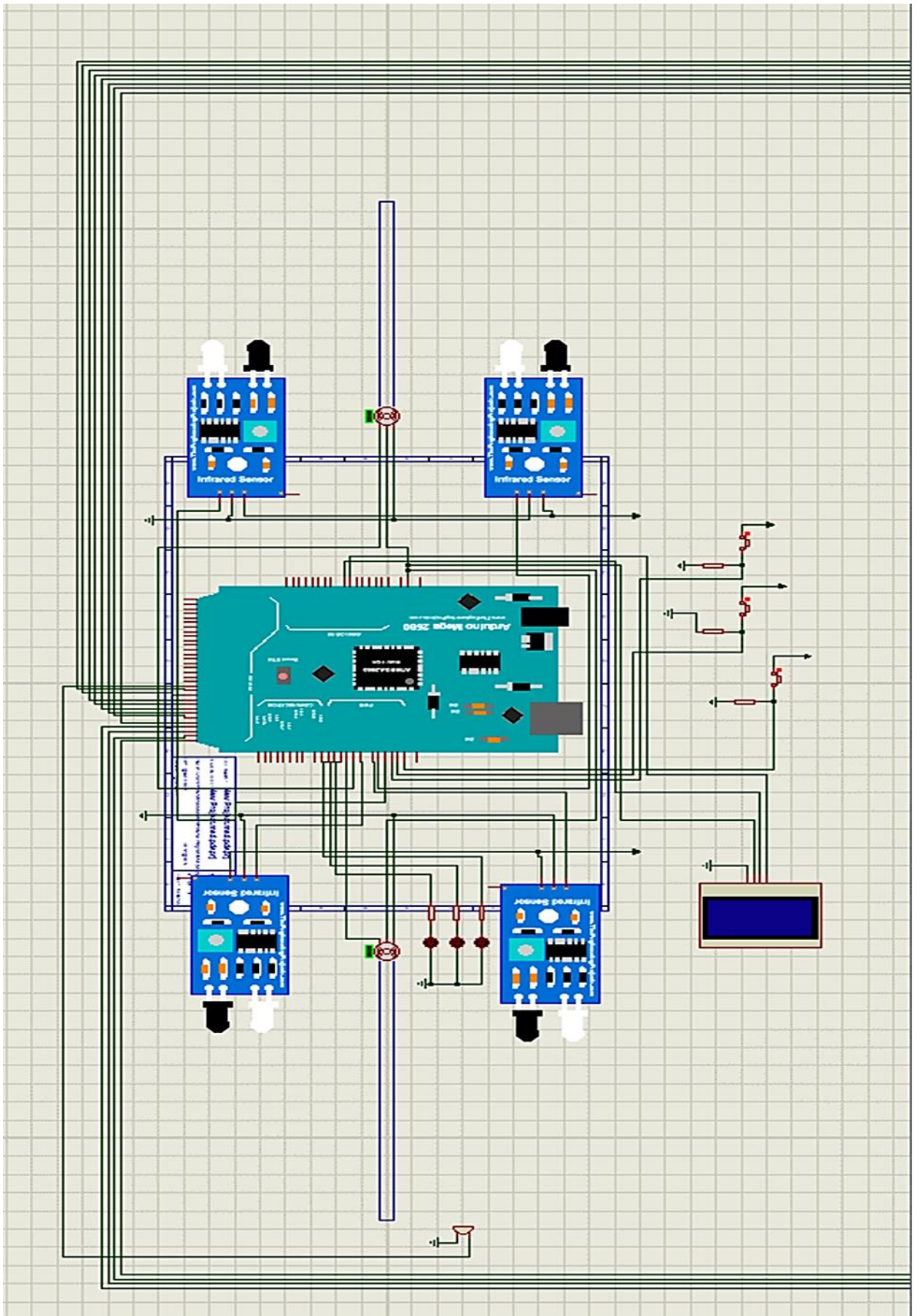


Figure 4.8 Schéma fonctionnel Proteus de la gestion d'entrée et sortie

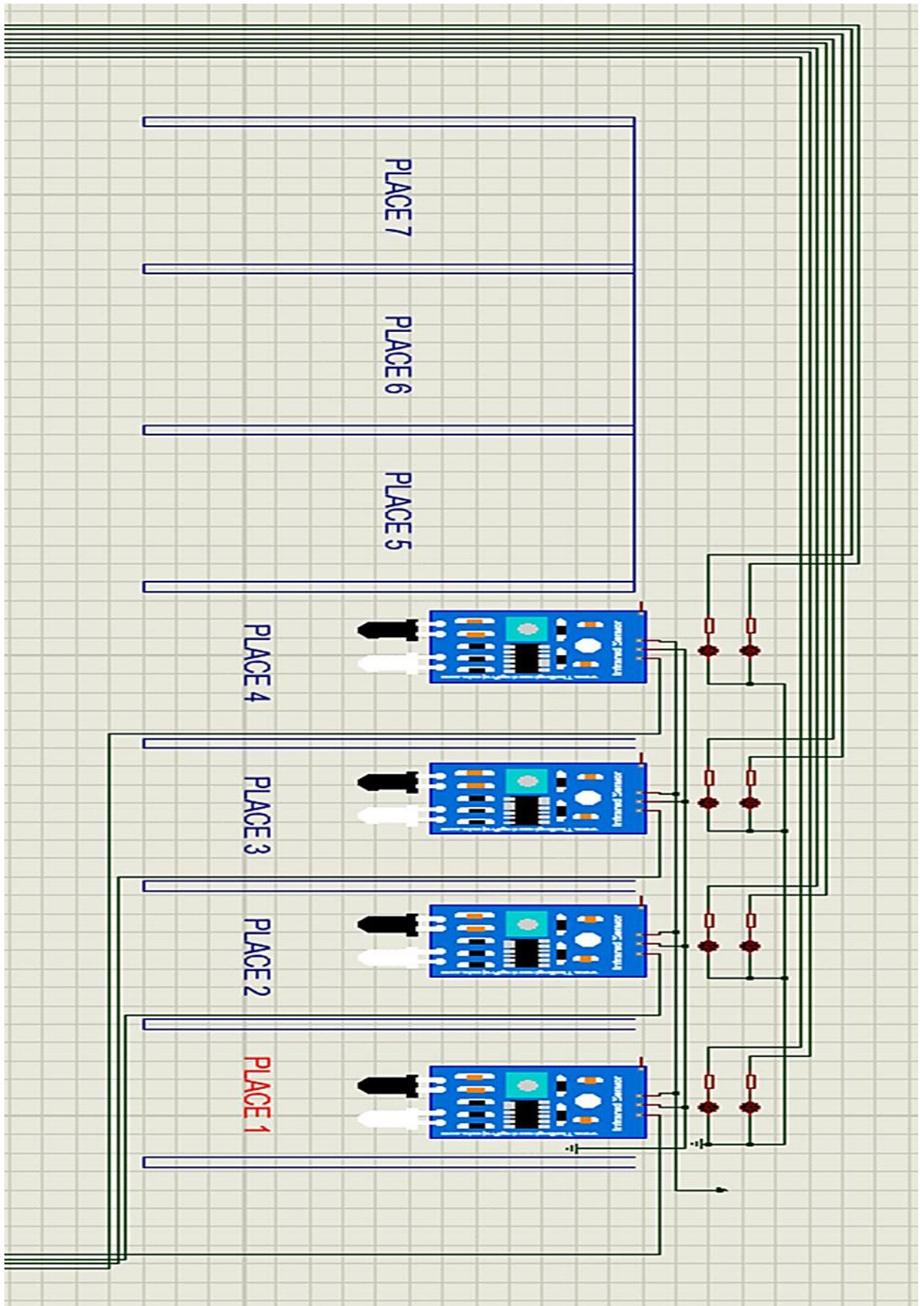
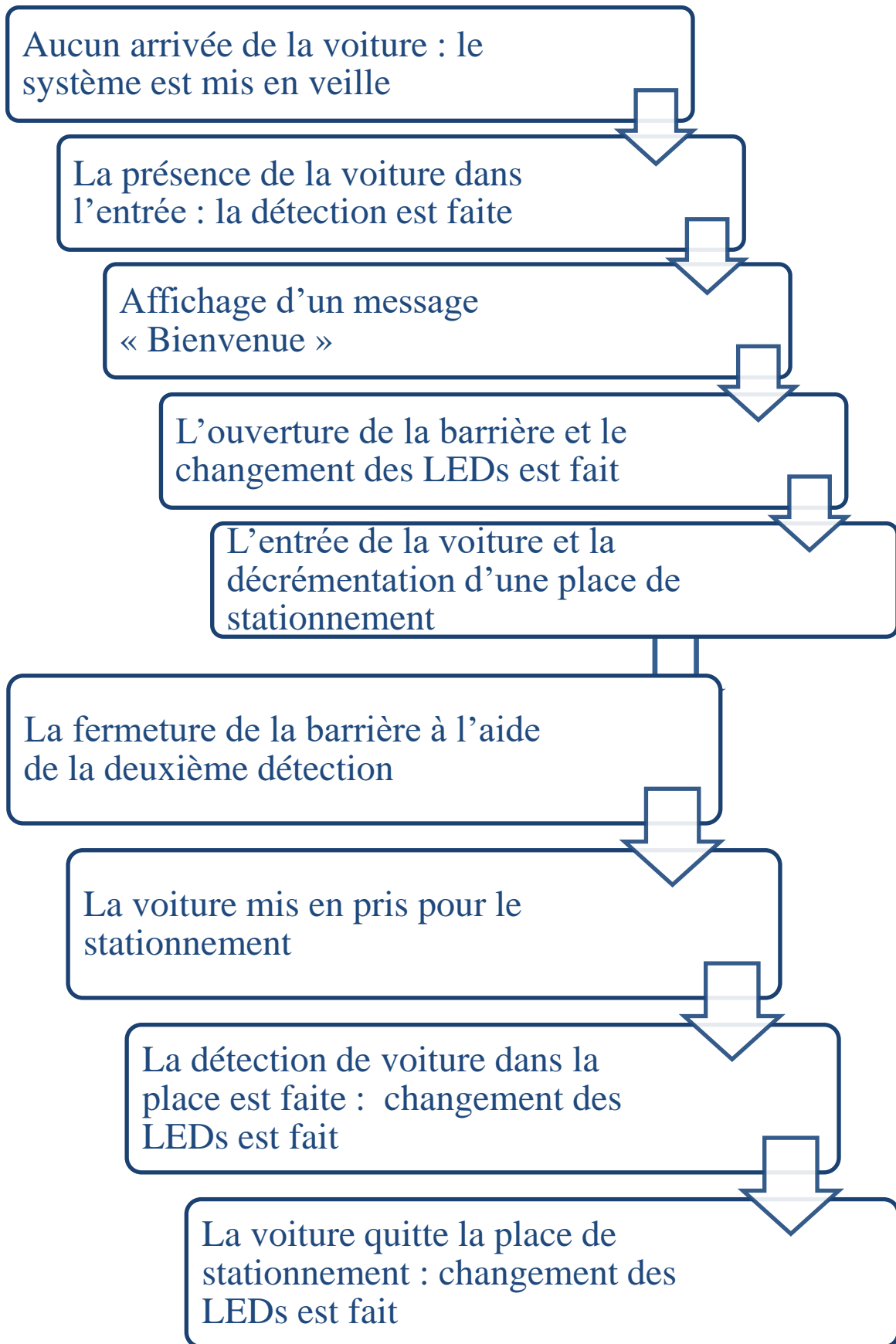


Figure 4.9 Schéma fonctionnel Proteus de la gestion de stationnement

4.5 LES DIFFERENTES ETAPES DE LA GESTION DE PARKING :



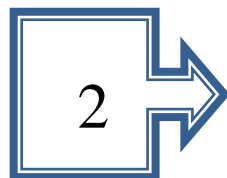
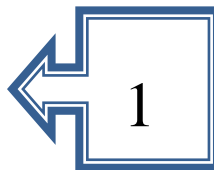
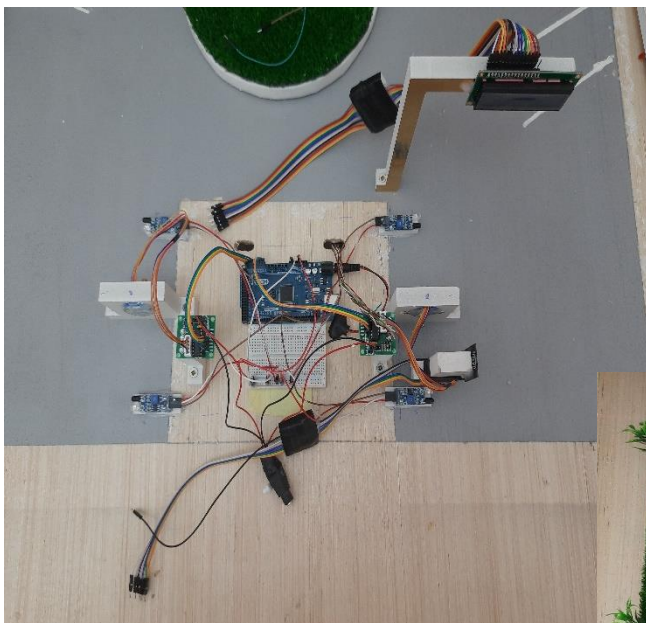
La présence de la voiture dans la sortie : la détection est faite

Affichage d'un message « Merci pour votre visite »

L'ouverture de la barrière et le changement des LEDs est fait

La sortie de la voiture et l'incrémentement d'une place de stationnement

La fermeture de la barrière à l'aide de la deuxième détection



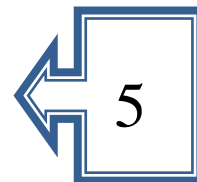
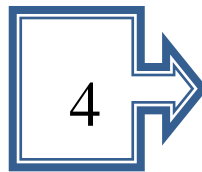
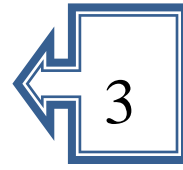


Figure 4.10 Images pour notre maquette

Conclusion générale et perspectives

Ce mémoire a but de réaliser une maquette d'une barrière levant automatique qui était exécutée à la base d'Arduino méga 2560.

Dans le premier chapitre, nous avons vu c'est quoi les systèmes automatisés et tout ce qui les concerne. Notamment, l'importance de l'automatisation consiste à confier d'un maximum de tâches répétitives et de peu de valeurs ajoutées à des systèmes informatiques plutôt qu'à des ressources humaines. Donc l'objectif est d'améliorer l'efficacité de la production et surtout renforcer la productivité, la fiabilité, la disponibilité et les performances.

Au deuxième chapitre, nous avons visé sur les barrières automatiques leurs mécanismes qui facilitent leur ouverture et leur fermeture. Elles sont également évoluées pour s'adapter à ces nouvelles technologies, elles jouent un rôle essentiel dans la prévention des intrusions non autorisées et la protection des personnes et des biens, et contrôlent le flux de circulation dans les rues et les parkings.

Dans le troisième et le quatrième chapitre, nous avons vu tous les composants et les logiciels qui nous avons choisi pour notre projet à côté d'une carte Arduino Mega. Ensuite, nous avons créé le programme qui comporte tous les composants pour gérer le fonctionnement de notre maquette.

La réalisation de cette barrière nous a permis d'acquérir des connaissances supplémentaires en conception et en développement, en fait, nous avons pu accéder à la maîtrise du langage d'Arduino.

On peut développer cette réalisation par plusieurs méthodes, parmi eux :

- Utilisation des capteurs magnétiques ou photocellules à la place des capteurs IR
- Utilisation un moteur pas à pas à la place de servomoteur
- Ajouter un module RFID pour la lecture des badges
- Ajouter une commande pour le servomoteur avec une télécommande
- Utilisation des caméras de surveillance
- Utilisation de reconnaissance faciale ou reconnaissance des matricules pour l'entrée
- Ajouter un lecteur des codes QR

Bibliographie

- [1] Dr. Herizi Abdelghafour, '*Cours automatismes industriels : Introduction aux systèmes automatisés*'. Page 1- 10, Université Msila.
- [2] C. Jossin, '*Automatisme : Buts de l'Automatisme*'. 2013 : Scribd.com
- [3] Joseph Aguilar Martin, '*Aujourd'hui-et-demain-au-laas : Une-petite-histoire-de-l-automatique-avec-en-fond-le-laas*'. 2008 : Universite Laas(laas.fr).
- [4] Dictionnaire, '*Définition : système*', Linternaute.fr, Consulter le : 12/06/2023
- [5] Antony Sciarretta, '*Systèmes automatisés*'. 2018 : Technologie-sciarretta.ovh
- [6] Charles Renouvier PRADES (Bac Pro MSPC), '*Cours : Les systèmes automatisés : Structure-générale d'un système automatisé de production*', Lycée Charles Renouvier (bpmei-prades.com), Consulter le : 11/06/2023
- [7] L. Bastien, '*Interface Homme-Machine : Tout savoir sur les IHM*'. 2018 : lebigdata.fr
- [8] L-K, '*Cours Interface Homme Machine*'. 2022 : physiquechimieathbiologie.com
- [9] JCG, '*Les Automatismes : Dialogue Homme / Machine HMI*'. 2012 : Lycée L. RASCOL (bts.crsa.rascol.free.fr)
- [10] JDoTec, '*Etude des systèmes : Chaîne fonctionnelle*', jdotec.net, Consulter le : 12/06/2023
- [11] Djarrallah Mohamed, '*Introduction aux systèmes automatisés*'. 2017 : Université batna2
- [12] Martin Ekuan, Gary Centric, '*Vue d'ensemble de l'automatisation : Objectifs de l'automatisation*'. 2023 : microsoft.com
- [13] Hassan Rayhane, '*Surveillance des systèmes de production automatisés : Détection et aide au diagnostic*'. 2004 : Institut National Polytechnique De Grenoble.
- [14] Science Maison, '*Technologies industrielles : Systèmes automatiques*', zpag.net, Consulter le : 10/06/2023

- [15] LBA Group, '*Comprendre les différents types de barrières levantes automatiques*'. 2018 : grouplba.com
- [16] Vincent, '*Technologie : Barrière de parking automatique, à quoi ça sert*'. 2017 : blog.interaces.net
- [17] Med Ali Haj Salah-prof info, '*Réalisation d'une barrière de parking automatique : Définition d'une barrière automatique*'. Robotique.tech, Consulter le : 17/05/2023
- [18] LBA Group, '*Comprendre les modèles de barrières levantes manuelles*'. 2018 : grouplba.com
- [19] Europages, '*Barriere manuelle levante BML Barrieres manuelles*', europages.fr, Consulter le : 17/05/2023
- [20] Frontier Pitts France, '*Nos produits : Barrières manuelles*', frontier-pitts.fr, Consulter le : 18/05/2023
- [21] Frontier Pitts France, '*Nos produits : Barrières automatiques*', frontier-pitts.fr, Consulter le : 18/05/2023
- [22] Hila Aicha, Ktibe Zayneb, '*ETUDE, CONCEPTION ET REALISATION D'UNE BARRIERE LEVANTE AUTOMATIQUE*'. 2021 : Mémoire de fin d'étude, Université Ahmed Draya d'ADRAR.
- [23] Rémi Britton, '*Dossier technique : Barrière automatique sympact*'. Page 11-18. 2005 : Lycée Ferry Versailles.
- [24] Toshiba.E, '*Dossier technique : Technologie : Barrière levante*'. 2016 : Lycée Ibn Zaydoun (fichier-pdf.fr)
- [25] Automatic Systems.E, '*Manuel de pose barrière Bl 229 : Barrière levante électrique*'. 2010 : Betafence.be
- [26] Société Acofase, '*les-indispensables-de-la-barrière-automatique*'. 2021 : acofase.fr
- [27] Chat. Gpt, '*Sources-d'alimentation-de-barrière-levante*'. Chat.openai.com, Consulter le : 18/06/2023
- [28] Damien Sauveron, '*Cours 4 : Introduction Arduino*'. Page 2-3. 2015 : Université de Limoges (Unilim.fr).

- [29] *'Présentation – Arduino : L'essentiel'*, arduino.blaisepascal.fr, Consulter le : 19/05/2023
- [30] Collège Gujan-Mestras, *'Technopc : fonction traitement de l'information carte de prototypage rapide Arduino Uno'*. collegegujan.fr, Consulter le : 26/05/2023
- [31] Admin7876, *'Les différentes cartes Arduino'*. 2019 : Domotics.fr
- [32] Zach DeMeyer, *'Electronic Components : Articles : Aperçu de la carte Arduino Mega 2560'*. 2017: Arrow Electronics Company (arrow.com).
- [33] Go Tronic (store), *'Cartes programmables: Arduino Mega 2560'*. Gotronic.fr, Consulter le: 18/06/2023
- [34] Chat. Gpt, *'Microcontrôleur-atmega2560-arduino-mega'*. Chat.openai.com, Consulter le : 20/06/2023
- [35] Shawn, *'ATmega2560 : Caractéristiques, comparaisons et Arduino Mega'*. 2019 : seedstudio.com
- [36] Components101(electronics design engineers), *'Microcontrôleurs : Microcontrôleur atmega2560'*. 2022 : Components101.com
- [37] Michel Deville, *'Hakken Robotto : Fiche technique : Servomoteur'*. 2021 : mcdinventor.fr
- [38] Mataucarre société, *'Arduino : Capteur de proximité Infrarouge – FC 51 – Arduino'*. 2017 : mataucarre.fr
- [39] Thebault, *'Détecteur infra-rouge FC-51 : Présentation'*. 2015 : gilles.thebault.free.fr
- [40] Richard Gauthier, *'Cours : Les afficheurs LCD'*. 2017 : Sciencesappliquees.com (Source : Lycée Le Dantec Lannion)
- [41] Utilisateur, *'Document Ressource : Le bouton poussoir'*. 2017 : Collège-lechaudron (portail.college-lechaudron.ac-reunion.fr).
- [42] Dzduino Société, *'buzzer-active-5v'*, dzduino.com, Consulter le : 30/05/2023
- [43] Volta Société, *'Composant Electronique : Buzzer-5v-actif'*. 2020 : Volta.ma
- [44] Lextronic Société, *'Composants électroniques : Led-diffusante-5mm'*, lextronic.fr, Consulter le : 28/05/2023

- [45] Olivier Lejeune, '*Electronique : Arduino: Qu'est-ce que Arduino*'. 2012 : positron-libre.com
- [46] Société Développez, '*cours-complet-Arduino*', Développez.com, Consulter le : 01/06/2023
- [47] Elektronique Société, '*Logiciels Electroniques : Proteus*', elektronique.fr, Consulter le : 04/06/2023
- [48] '*Fiches cpge : Proteus : Démarrage et interface de Proteus*'.2020 : cpge.frama.io
- [49] Kawser Ahmed, '*Topic on Flowchart: Introduction to computer studies*'. 2014: slideshare.net

Annexes

Annexe A

Annexe des matériels

La configuration de brochage pour le microcontrôleur ATmega2560 :

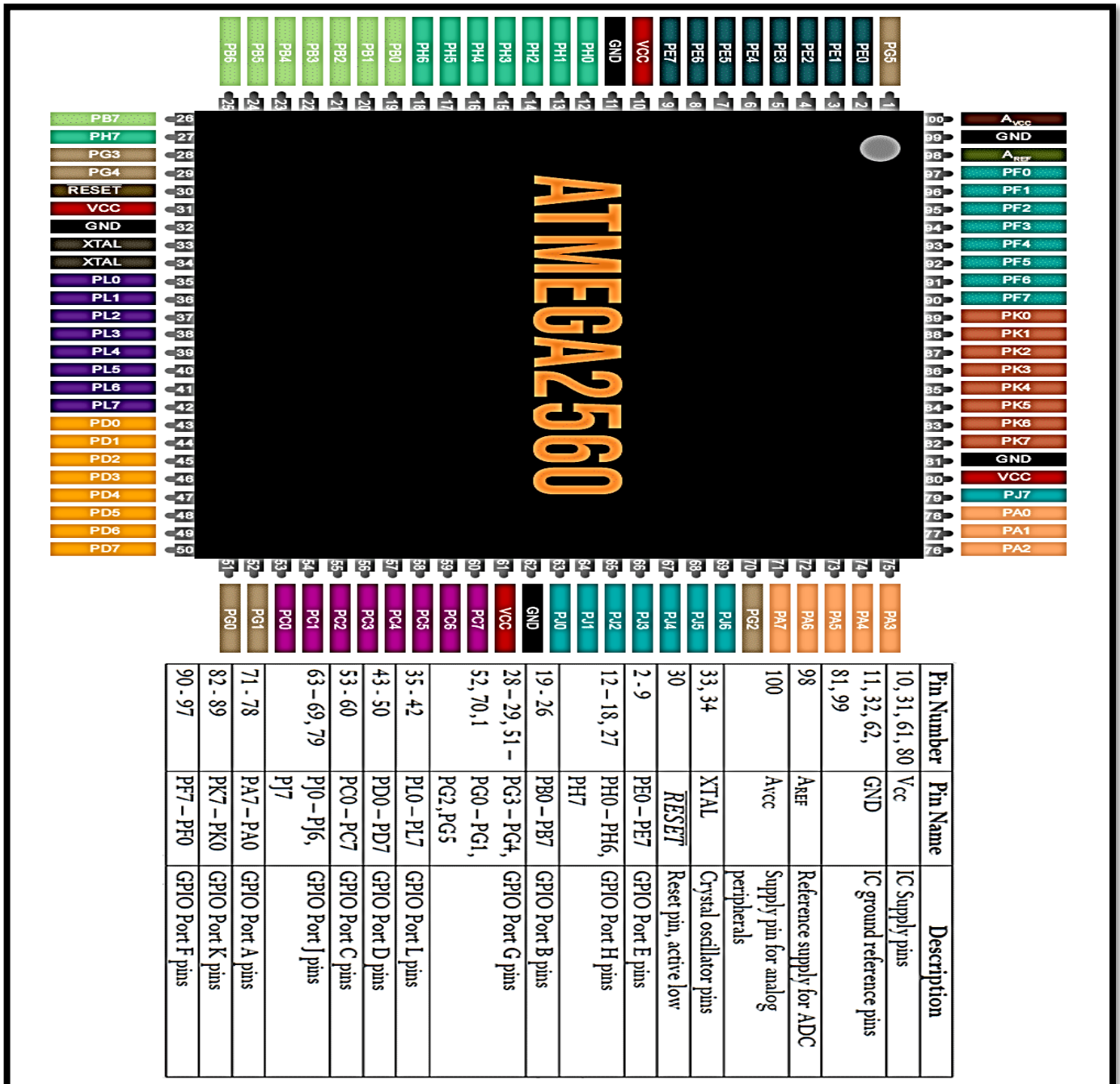


Figure A.1 Configuration de brochage pour le microcontrôleur ATmega2560

Description plus détaillée des principales caractéristiques du microcontrôleur ATmega2560 :

Architecture et performance :

Architecture RISC 8 bits : L'ATmega2560 utilise une architecture RISC (Reduced Instruction Set Computer) de 8 bits, ce qui signifie qu'il exécute des instructions de 8 bits à la fois.

Vitesse d'horloge : Il fonctionne avec une horloge interne ou externe, et sa vitesse d'horloge maximale est de 16 MHz. Cela signifie qu'il peut exécuter des instructions à une fréquence de 16 millions de fois par seconde. [34]

Mémoire :

Mémoire flash : Il dispose d'une mémoire flash de 256 kilooctets (Ko) qui est utilisée pour stocker le programme principal (code).

Mémoire SRAM : Il possède une mémoire SRAM (Static Random Access Memory) de 8 Ko qui est utilisée pour le stockage temporaire des données en cours de traitement.

Mémoire EEPROM : Il intègre également une mémoire EEPROM (Electrically Erasable Programmable Read-Only Memory) de 4 Ko. Cette mémoire est non volatile, ce qui signifie qu'elle conserve les données même lorsque l'alimentation est coupée. [34]

Broches d'E/S :

Entrées/Sorties (E/S) : L'ATmega2560 dispose de 86 broches d'entrée/sortie, dont 54 peuvent être configurées comme sorties et 86 comme entrées. Ces broches peuvent être utilisées pour interagir avec des composants externes tels que des capteurs, des actionneurs, des afficheurs, etc.

Fonctionnalités spéciales : Certaines broches peuvent être utilisées pour des fonctionnalités spéciales, telles que les interfaces série (UART), les interfaces de communication I2C et SPI, les entrées analogiques, etc. [34]

Interfaces de communication :

UART : Il dispose d'interfaces UART (Universal Asynchronous Receiver-Transmitter) qui permettent la communication série asynchrone.

I2C : Il prend en charge l'interface de communication I2C (Inter-Integrated Circuit) utilisée pour la communication avec d'autres périphériques en utilisant un bus à deux fils.

SPI : Il intègre une interface SPI (Serial Peripheral Interface) qui permet la communication série synchrone avec d'autres périphériques.

USB : Il offre également la possibilité de communication via l'interface USB (Universal Serial Bus). [34]

Fonctionnalités supplémentaires :

Convertisseurs analogique-numérique (CAN) : L'ATmega2560 dispose de 16 canaux d'entrée analogique qui permettent de convertir des signaux analogiques en valeurs numériques.

Timers/Counters : Il possède plusieurs timers/counters internes qui peuvent être utilisés pour la mesure de temps, la génération de signaux PWM, etc.

Interruptions matérielles : Il prend en charge les interruptions matérielles, permettant d'interrompre l'exécution normale. [34]

Le Servomoteur – Arduino :

La commande des servomoteurs :

La consigne envoyée au servomoteur n'est autre qu'un signal électronique de type PWM (Pulse Width Modulation ou Modulation de Largeur d'Impulsion). Il dispose cependant de deux caractéristiques indispensables pour que le servomoteur puisse comprendre ce qu'on lui demande.

A savoir : une fréquence fixe de valeur 50Hz et d'une durée à l'état HAUT elle aussi fixée dans certaines limites.

Comment fonctionne un servomoteur :

Le moteur est connecté au réducteur, le potentiomètre est attaché à l'axe de sortie et le circuit de contrôle alimente le moteur et lit la valeur du potentiomètre. Le circuit de contrôle reçoit également des informations du microcontrôleur qui lui indique la position à laquelle doit se placer l'axe.

Le moteur est alors alimenté et au travers du réducteur, l'axe tourne ainsi que le potentiomètre. En lisant régulièrement la valeur du potentiomètre, le circuit peut moduler la tension aux bornes du moteur pour placer exactement l'axe à la position souhaitée.

Cas d'utilisation :

- Mettre en mouvement un système facilement et pour un prix dérisoire

- Le mouvement doit être précis et ne dépasse pas la plage de 0-270°
- Le couple requis est relativement faible (~2kg.cm max)

Branchement avec Arduino :

- Rouge : pour l'alimentation positive (4.5V a 6V en général)
- Noir ou marron : pour la masse (0V)
- Orange, jaune, blanc, G : entrée du signal de commande

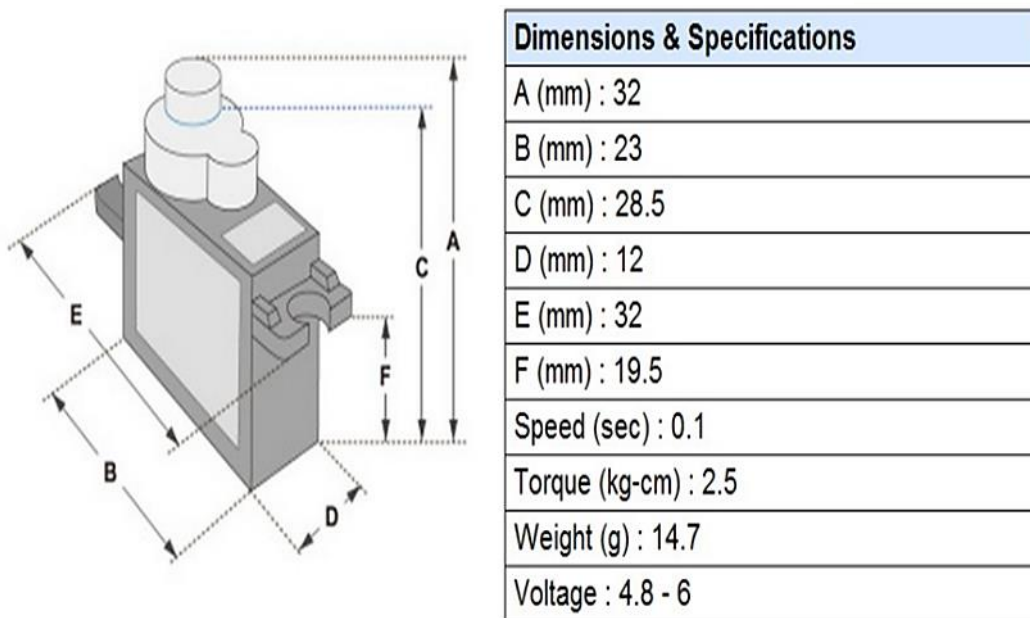
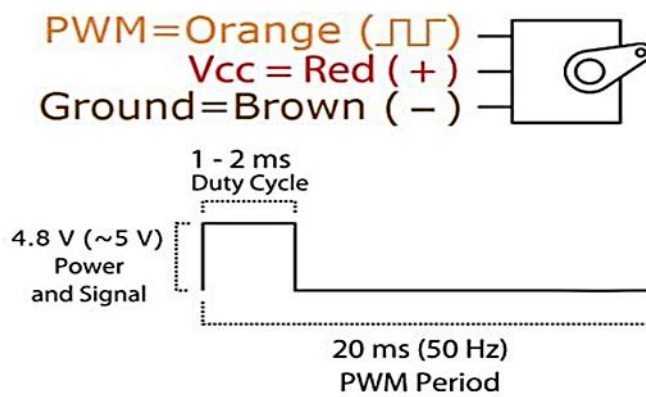


Figure A.2 Servomoteur datasheet

Capteur de proximité Infrarouge – FC 51 – Arduino :

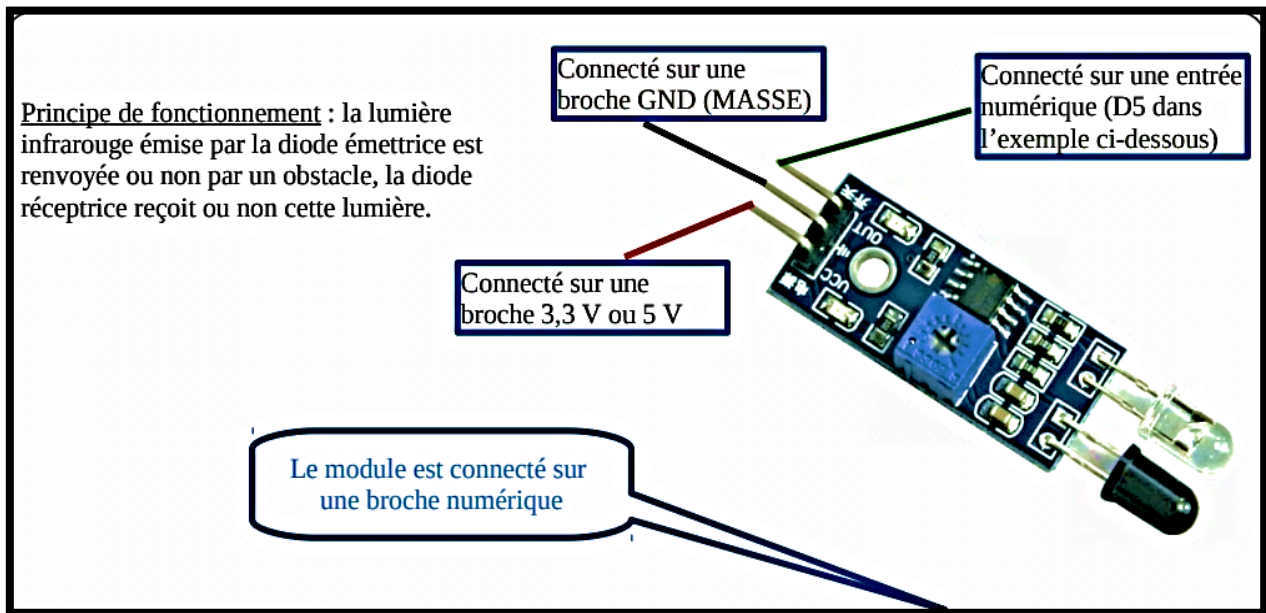


Figure A.3 Brochage d'un capteur de proximité Infrarouge – FC51

La distance de détection de ce module est de 2 à 30 cm (ou plus selon les modèles), l'angle de détection est de 35°.

Cette distance de détection peut être contrôlée par potentiomètre, il faut tourner le potentiomètre

Dans le sens horaire pour augmenter la distance de détection.



Dans le sens anti-horaire, la distance de détection diminue.

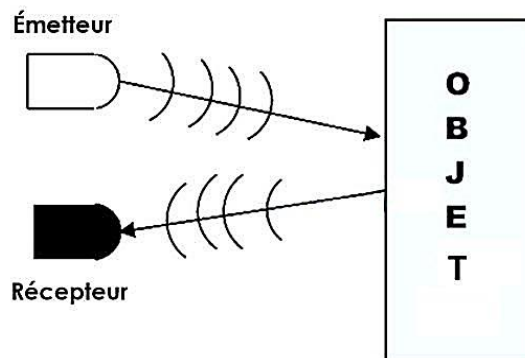


Figure A.4 Principe de fonctionnement

LED 5mm – Arduino :
Brochage avec Arduino :

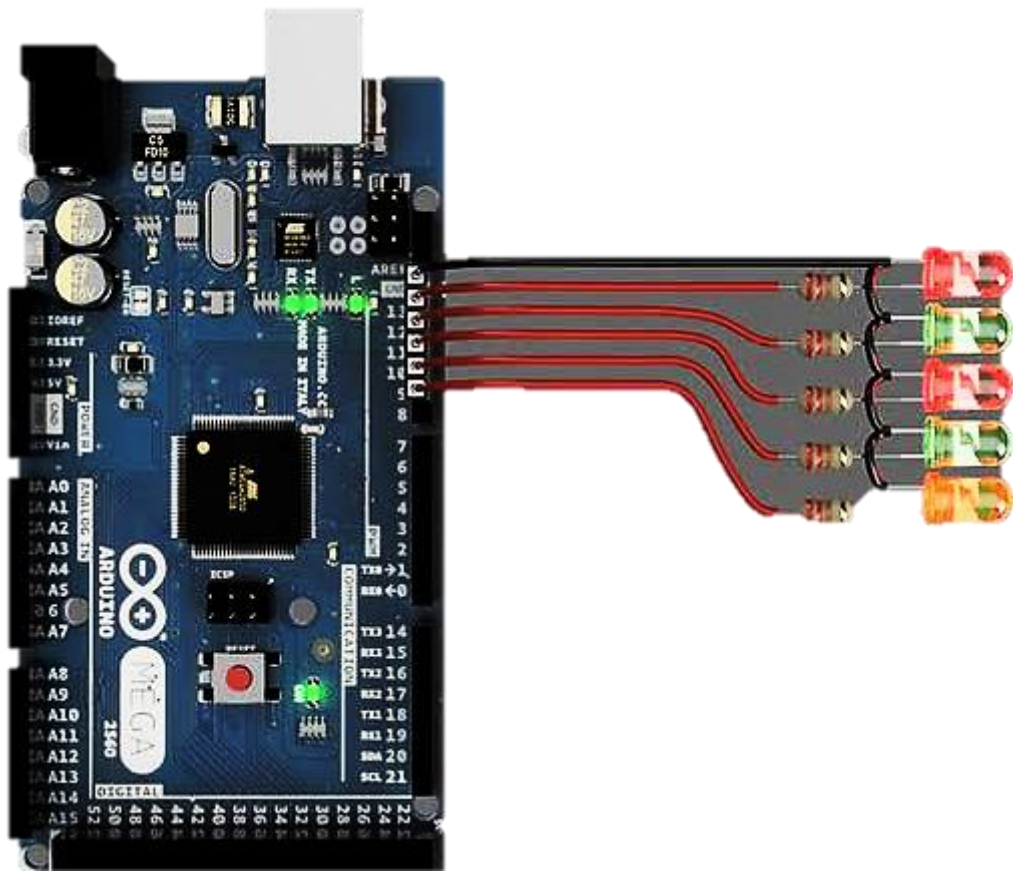


Figure A.5 Brochage de LED avec Arduino

Le fil noir est la masse branchée sur Arduino et la masse des leds.

Les fils rouges (les sorties) sont branchés sur chaque pins du port Arduino avec les leds lumineux.

Buzzer actif 5v - Arduino :
Branchement avec Arduino :

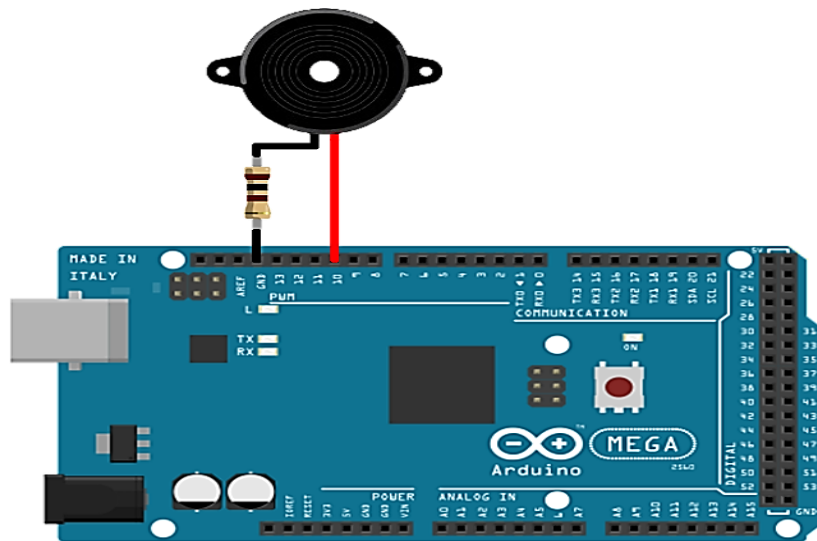


Figure A.6 Branchement de buzzer actif avec Arduino

Le fil noir est la masse branchée sur Arduino et la masse de buzzer.

Le fil rouge (la sortie) est branché sur un pin du port Arduino avec le buzzer.

Bouton poussoir 4 pattes – Arduino :
Branchement avec Arduino :

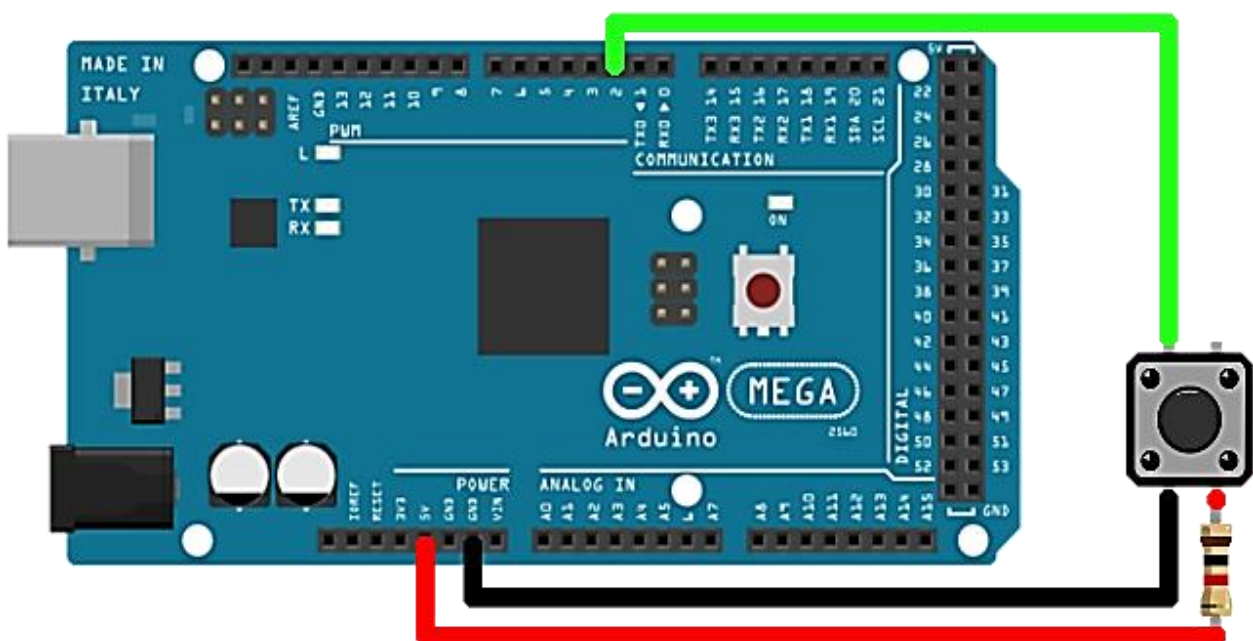


Figure A.7 Branchement de bouton poussoir avec Arduino

Ecran LCD – Arduino :

Composition d'écran LCD :

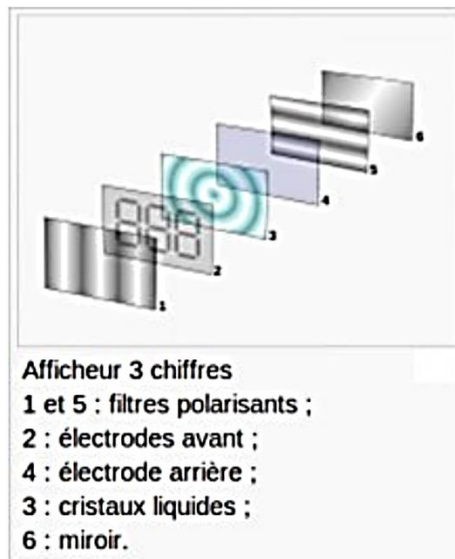


Figure A.8 Composition d'écran LCD

Branchement avec Arduino :

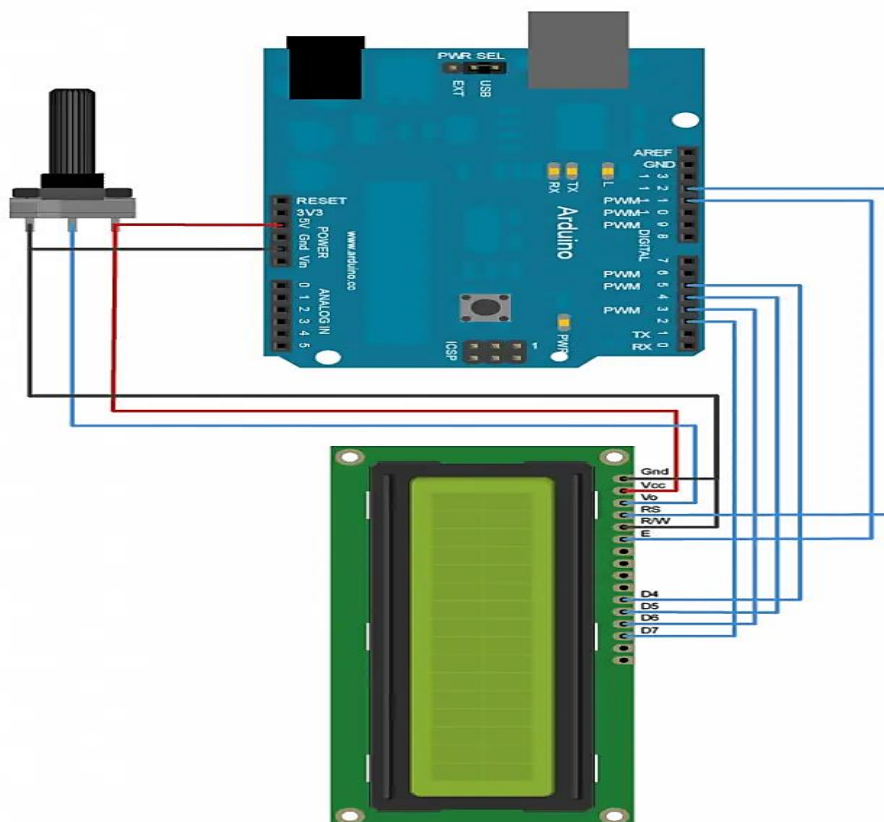


Figure A.9 Branchement d'écran LCD avec Arduino

Annexe B

Formatage du texte

```
#include <Servo.h>
#include <LiquidCrystal.h>
#define LED_VERTE 4
#define LED_JAUNE 3
#define LED_ROUGE 2
Servo moteur;
Servo moteur2;
int barriere;
int barriere2;
int LED1 = 26;
int LED11 = 27;
int LED2 = 28;
int LED22 = 29;
int LED3 = 30;
int LED33 = 31;
int LED4 = 32;
int LED44 = 33;
int capteur=7;
int capteur2 = 8;
int capteur3 = 9;
int capteur4 = 10;
int capteur5 = 22;
int capteur6 = 23;
int capteur7 = 24;
int capteur8 = 25;
int detection;
int detection2;
int detection3;
int detection4;
int detection5 = HIGH;
int detection6 = HIGH;
int detection7 = HIGH;
int detection8 = HIGH;
int barriereposition=0;
int barriere2position=0;
int BuzzPin = 13;
const int button1Pin = 34;
const int button2Pin = 35;
const int button3Pin = 36;
int button1State = 0;
int button2State = 0;
int button3State = 0;
const int rs = 12, en = 11, d4 = 40, d5 = 39, d6 = 38, d7 = 37;
```

```

LiquidCrystal lcd(rs, en, d4, d5, d6, d7);

void setup(){
  pinMode(capteur,INPUT);
  pinMode(capteur2,INPUT);
  pinMode(capteur3,INPUT);
  pinMode(capteur4,INPUT);
  pinMode(capteur5, INPUT);
  pinMode(capteur6, INPUT);
  pinMode(capteur7, INPUT);
  pinMode(capteur8, INPUT);
  pinMode(button1Pin, INPUT);
  pinMode(button2Pin, INPUT);
  pinMode(button3Pin, INPUT);
  pinMode(LED_VERTE, OUTPUT);
  pinMode(LED_JAUNE, OUTPUT);
  pinMode(LED_ROUGE, OUTPUT);
  pinMode(LED1, OUTPUT);
  pinMode(LED11, OUTPUT);
  pinMode(LED2, OUTPUT);
  pinMode(LED22, OUTPUT);
  pinMode(LED3, OUTPUT);
  pinMode(LED33, OUTPUT);
  pinMode(LED4, OUTPUT);
  pinMode(LED44, OUTPUT);
  pinMode(BuzzPin, OUTPUT);
  pinMode(LED_ROUGE, HIGH);
  pinMode(LED1, HIGH);
  pinMode(LED2, HIGH);
  pinMode(LED3, HIGH);
  pinMode(LED4, HIGH);
  lcd.begin(16, 2);
  lcd.print("CAR PARKING!");
  lcd.blink();
  Serial.begin(9600);
}

void loop(){
  detection = digitalRead(capteur);
  detection2 = digitalRead(capteur2);
  if ((detection == 1) && (detection2 == 1)){ // rien
    fermer();
    delay(1000);}
  else { // voiture
    lcd.print("BIENVENUE");
    delay(3000);
    lcd.clear();
    ouvrir();
    delay(1000);}
}

```

```

button1State = digitalRead(button1Pin);
button2State = digitalRead(button2Pin);
if (button1State == HIGH) {
    ouvrir();
}
if (button2State == HIGH) {
    fermer();
}
detection3 = digitalRead(capteur3);
detection4 = digitalRead(capteur4);
if ((detection3 == 1) && (detection4 == 1)){ // rien
    fermer2();
    delay(1000);}
else { // voiture
    lcd.setCursor(0,0);
    lcd.print("MERCI POUR");
    lcd.setCursor(0,1);
    lcd.print("votre visite");
    delay(3000);
    lcd.clear();
    ouvrir2();
    delay(1000);}
detection5 = digitalRead(capteur5);
if (detection5 == HIGH) {

    digitalWrite(LED1, HIGH);
    digitalWrite(LED11, LOW);
    delay(2000);
} else {

    digitalWrite(LED1, LOW);
    digitalWrite(LED11, HIGH);
    delay(2000);
}
delay(2000);
detection6 = digitalRead(capteur6);
if (detection6 == HIGH) {

    digitalWrite(LED2, HIGH);
    digitalWrite(LED22, LOW);
    delay(2000);
} else {

    digitalWrite(LED2, LOW);
    digitalWrite(LED22, HIGH);
    delay(2000);
}
delay(2000);
detection7 = digitalRead(capteur7);

```



```

if (detection7 == HIGH) {

    digitalWrite(LED3, HIGH);
    digitalWrite(LED33, LOW);
    delay(2000);
} else {

    digitalWrite(LED3, LOW);
    digitalWrite(LED33, HIGH);
    delay(2000);
}
delay(2000);
detection8 = digitalRead(capteur8);
if (detection8 == HIGH) {

    digitalWrite(LED4, HIGH);
    digitalWrite(LED44, LOW);
    delay(2000);
} else {

    digitalWrite(LED4, LOW);
    digitalWrite(LED44, HIGH);
    delay(2000);
}
delay(2000);
button3State = digitalRead(button3Pin);
    if (button3State == HIGH) {
        fermer();
        fermer2();
    }
}
void ouvrir(){
    if (barriereposition == 0){
        digitalWrite(LED_ROUGE, LOW);
        digitalWrite(LED_VERTE, HIGH);
        delay(500);
        moteur.attach(5);}
    for (barriere = 100; barriere >=10; barriere -=1)
        {moteur.write(barriere);
        delay(1000);
        tone(BuzzPin, 400);
        delay(2000);
        }
    barriereposition = 1;
    moteur.detach();
    noTone(BuzzPin);
}
void fermer(){
    if (barriereposition == 1){
        digitalWrite(LED_VERTE, LOW);

```

```

digitalWrite(LED_JAUNE, HIGH);
delay(1000);
digitalWrite(LED_JAUNE, LOW);
digitalWrite(LED_ROUGE, HIGH);
moteur.attach(5);}
for (barriere = 10; barriere <=100; barriere +=1)
  {moteur.write(barriere);
  delay(1000);
  tone(BuzzPin, 400);
  delay(2000);
  }
  barriereposition = 0;
  moteur.detach();
  noTone(BuzzPin);
}
void ouvrir2(){
  if (barriere2position == 0){
    digitalWrite(LED_ROUGE, LOW);
    digitalWrite(LED_VERTE, HIGH);
    delay(500);
    moteur2.attach(6);}
  for (barriere2 = 100; barriere2 >=10; barriere2 -=1)
    {moteur2.write(barriere2);
    delay(1000);
    tone(BuzzPin, 400);
    delay(1000);
    }
    barriere2position = 1;
    moteur2.detach();
    noTone(BuzzPin);
}
void fermer2(){
  if (barriere2position == 1){
    digitalWrite(LED_VERTE, LOW);
    digitalWrite(LED_JAUNE, HIGH);
    delay(1000);
    digitalWrite(LED_JAUNE, LOW);
    digitalWrite(LED_ROUGE, HIGH);
    moteur2.attach(6);}
  for (barriere2 = 10; barriere2 <=100; barriere2 +=1)
    {moteur2.write(barriere2);
    delay(1000);
    tone(BuzzPin, 400);
    delay(1000);
    }
    barriere2position = 0;
    moteur2.detach();
    noTone(BuzzPin);
}

```