



الجمهورية الجزائرية الديمقراطية الشعبية
République Algérienne Démocratique et Populaire
وزارة التعليم العالي والبحث العلمي
Ministère de l'Enseignement Supérieure et de la Recherche Scientifique

جامعة وهران 2 محمد بن أحمد
Université d'Oran 2 Mohamed Ben Ahmed

معهد الصيانة و الأمن الصناعي
Institut de Maintenance et de Sécurité Industrielle

Département De Maintenance En Instrumentation

MÉMOIRE

Pour l'obtention du diplôme de Master

Filière : Génie industriel
Spécialité : Génie industriel

Thème

**Conception et réalisation d'un système automatisé de distribution
des médicaments à base d'Arduino**

Présenté et soutenu publiquement par :

TABTI Asmaa

BOUZIDI Sarra

Devant le jury composé de :

Nom et Prénom	Grade	Etablissement	Qualité
AISSANI Nassima	Maître de Conférences A	Univ d'oran 2	Présidente
HACHEMI Khalid	Professeur	Univ d'oran 2	Encadreur
ADDA NEGGAZ Samir	Maître Assistant A	Univ d'oran 2	Examineur
RABHI Abderrahmane El Djalil	Ingénieur CDS	CDS , Oran	Co-encadreur

Année 2018/2019

Remerciements

En premier lieu, nous remercions Dieu le tout puissant de nous avoir donné la force et le courage pour réaliser ce modeste travail, sans lequel nous n'aurions pas pu progresser.

Nos plus vifs remerciements vont à notre encadreur Mr Hachemi Khalid pour nous avoir proposé ce sujet, et de nous avoir fait confiance.

Sa disponibilité dans les moments difficiles, et ses conseils judicieux ont été pour nous l'unique repère, puisse-t-il trouver dans ce travail le témoignage de nos profondes gratitude. Nous remercions vraiment l'encadreur Mr Rabhi Abderrahmane el Djalil, pour nous avoir encouragé, conseillé, guidé et aidé le long de cette difficile période, veuillez accepter l'expression de nos profonds respects, que Dieu vous protège.

Nous tenons à citer dans ces remerciements les membres du jury qui ont bien voulu examiner et juger notre travail.

Nous remercions tous nos professeurs de l'IMSI qui ont contribué à notre formation.

Un merci particulier à Mr Iles Amine Directeur plateforme Technologique Robotique de Services S.M.R.I.S, Enseignant chercheur en mécanique et robotique -université des Sciences et de la technologie USTO MB - pour son aide durant la conception d'un convoyeur.

Nous vous exprimons notre profonde gratitude et un immense respect.

Nous tenons à remercier sincèrement Mr Ibrahim Fenni et l'équipe de decograde, spécialisée dans la gravure numérique qui nous a aidés pour la construction de la maquette convoyeur.

La concrétisation de ce travail n'aurait jamais vu le jour sans la précieuse collaboration et sans une volonté d'une équipe dynamique, nos familles, nos amis (es) et nos proches, trouvent l'expression de notre profonde gratitude.

Dédicaces

Je dédie ce modeste travail À ;

ALLAH le tout-puissant à qui je dois tout.

À mes Chers Parents;

*Pour leur amour, leurs encouragements et soutien sans failles,
Je les remercie de m'avoir accompagné tout au long de mon parcours*

À mes soeurs;

Hanane, Nacera, Zahra, et à mes frères Said et Bachir

*Qui ont beaucoup apporté à sa réalisation et sans lesquelles je ne
serais pas la personne que je suis aujourd'hui.*

*De faire de cette réalisation ce qu'elle est un moment de ma vie,
gravé dans mon coeur. Merci pour votre amour. Je ne serais pas là
sans vous.*

À toute ma Famille.

A mon binôme : Asmaa

*À tous mes camarades de promotion, mes meilleurs amis et tous
ceux qui ont contribué de près ou de loin à ce travail. À ceux qui
se sont toujours dévoués et sacrifiés pour moi; qui ont toujours
été là dans mes moments de détresse.*

SARRA



Dédicaces

Que ce travail témoigne de mes respects :

À mes parents : Grâce à leurs tendres encouragements et leurs grands sacrifices, ils ont pu créer le climat affectueux et propice à la poursuite de mes études. Aucune dédicace ne pourrait exprimer mon respect, ma considération et mes profonds sentiments envers eux. Je prie le bon Dieu de les bénir, de veiller sur eux, en espérant qu'ils seront toujours fiers de moi.

À mes frères : Mohamed, Tayeb, Abd Illah et Abd El Chakir.

À tous mes professeurs : Leur générosité et leur soutien m'oblige de leurs témoigner mon profond respect et ma loyale considération.

À mon binôme : Sarra

À tous ceux qui ont contribué de près ou de loin pour que ce projet soit possible, je vous dis merci.

À tous mes amis et mes collègues : Khadidja, Hayat, Yassin, Amar ; Ils vont trouver ici le témoignage d'une fidélité et d'une amitié infinie.

Asmaa



Résumé :

Le but de ce travail est la conception et la réalisation d'un système automatisé de distribution des médicaments de 16 types, Ce système s'est inspiré de l'automate français 'ApotéKa' qui utilise le Flow-rack AS/RS à chute libre (FF-flow-rack AS/RS).

Notre système est basé sur les servomoteurs pour provoquer la chute libre des médicaments vers un convoyeur qui les transportent. Les servomoteurs sont pilotés par un module PCA968 commandé par une carte Arduino mega2560 par un PC, la présence ou non des boîtes dans les casiers est détectée par un capteur d'obstacle infrarouge.

Mots clés: Flow-rack AS/RS, chute libre, distribution de médicaments, servomoteur, Arduino, commandé.

Abstract

The aim of this work is the conception and the realization of an automated drug dispensing system, of 16 types, this system was inspired by the French automaton 'ApotéKa' which uses the Free Fall flow-rack AS/RS (FF-flow-rack AS / RS).

Our system is based on servomotors to provoke the free fall of the drugs towards a conveyor that transport them. The servomotors are piloted by a PCA968 module which is controlled by Mega2560 Arduino board by PC, the presence or absence of the boxes in the bins is detected by an infrared obstacle sensor.

Keywords: flow-rack AS/RS, Free Fall, drug dispensing, Servomotors, Arduino, controlled.

Figure I.1. Système Hybride Robomat.....05
Figure I.2. Système de convoyage pneumatique Airtec de Tecnilab.....07
Figure I.3. Tapis roulant avec système d'aiguillage.....08
Figure I.4. Exemple de glissière.....09
Figure I.5. L'ApotéKa.....11
Figure I.6. Le MoveTec.....11
Figure I.7. L'Ariana.....13
Figure I.8. Le Pharmatrack.....14
Figure I.9. Intérieur du Rowa.....15
Figure I.10. Bras articulé du Rowa.....15
Figure I.11. L'Unguator.....16
Figure I.12. Accessoires de l'Unguator.....17

Figures du Chapitre II

Figure II.1. Système automatisé de Stockage/Déstockage (AS/RS).....23
Figure II.2. Principaux constituants d'un AS/RS.....24
Figure II.3. Racks de stockage.....25
Figure II.4. Machine de stockage/déstockage.....26
Figure II.5. Station dépôt/ livraison.....27
Figure II.6. Système de contrôle28
Figure II.7. Photo réelle d'un flow-rack AS/RS à chute libre « APOTEKA ».....30
Figure II.8. Constitution typique d'un flow-rack AS/RS à chute libre.....31
Figure II.9. Côté remplissage de l'ApotéKa.....33
Figure II.10. Fonctionnement du système d'éjection des produits dans un flow-rack AS/RS
à chute libre.....34
Figure II.11. Représentation des mouvements du produit dans la face de déstockage (face discrète)...34

Figures du Chapitre III

Figure III.1. Photo réelle d'Arduino Méga 2560.....	40
Figure III.2. Brochage de la carte Méga 2560.....	41
Figure III.3. Logiciel de programmation de la carte ARDUINO.....	46
Figure III.4. Arduino 1.8.1.....	46
Figure III.5. Les servomoteurs.....	49
Figure III.6. Architecture d'un servomoteur.....	49
Figure III.7. Caractéristiques du servomoteur à rotation angulaire (Micro-Servo).....	50
Figure III.8. Câble de commande pour servomoteur.....	50
Figure III.9. Principe d'un signal de commande par PWM.....	51
Figure III.10. Exemples des signaux PWM.....	51
Figure III.11. La commande à l'aide Arduino.....	52
Figure III.12. Le PCA9685.....	53
Figure III.13. Brochage de PCA9685.....	54
Figure III.14. Alimentation de PCA9685.....	55
Figure III.15. Branchement d'un servomoteur (couleurs Graupner) sur PCA9685.....	55
Figure III.16. Dimensions de module émetteur et de récepteur.....	56
Figure III.17. Composition et branchement de Capteur d'obstacle infrarouge à 4 canaux.....	56
Figure III.18. Principe de fonctionnement d'un capteur infrarouge.....	57
Figure III.19. Fonctionnement de machine à courant continu.....	58
Figure III.20. Description de la machine à courant continu.....	58
Figure III.21. Principe de fonctionnement d'un moteur à courant continu.....	59
Figure III.22. Les types des relais.....	60
Figure III.23. Le relais SPDT.....	61

Figures du Chapitre IV

Figure IV.1. Maquette en 3D	63
Figure IV.2. Vue de face de la maquette.....	64
Figure IV.3. vue arrière de la maquette.....	64
Figure IV.4. Phase du montage de maquette.....	65

Figure IV.5. Convoyeur en 3D.....	65
Figure IV.6. composants de convoyeur	66
Figure IV.7. Photo réelle de convoyeur fabriqué en PVC avant et après le montage.....	66
Figure IV.8. photo réelle de la carte Arduino mega2560	67
Figure IV.9. photo réelle du module PCA9685	68
Figure IV.10. photo réelle d'alimentation AC 100-240v 50/60hz.....	68
Figure IV.11. photo réelle d'un servomoteur sg90.....	69
Figure IV.12. DC moteur.....	70
Figure IV.13. Photo réelle de Module relais 5v 1 canal.....	71
Figure IV.14. Phase de préparation des câbles.....	71
Figure IV.15. câbles avec jumpers.....	72
Figure IV.16. Montage d'Arduino méga et module PCA9685 avec les servomoteurs.....	73
Figure IV.17. Montage d'Arduino méga avec capteur 4 canal infrarouge.....	74
Figure IV.18. Montage d'Arduino méga avec relais et DC moteur.....	75
Figure IV.19. Simulation d'Arduino méga et module PCA9685 avec les servomoteurs.....	76
Figure IV.20. Simulation d'Arduino méga avec capteur 4 canal infrarouge.....	77
Figure IV.21. Simulation d'Arduino méga avec DC moteur et relais.....	77
Figure IV.22. Montage réel d'Arduino méga et module PCA9685 avec les servomoteurs.....	81
Figure IV.23. Test réel du capteur d'obstacle infrarouge a 4 canaux.....	82
Figure IV.24. Photo réelle de système globale.....	83

Tableau II.1. Quelques caractéristiques d'ApotéKa	31
Tableau III.1. Synthèses de caractéristiques.....	42
Tableau III.2. Barre d'actions du logiciel IDE.....	48
Tableau III.3. la programmation de servomoteur sur l'Arduino.....	52
Tableau III.4. Raccordement sue Arduino.....	54
Tableau IV.1. Branchement de module émetteur / récepteur.....	74
Tableau IV.2. Branchement de module récepteur/ Arduino.....	75
Tableau IV.3. Le branchement de relais avec Arduino.....	76

AS/RS : (Automated Storage/Retrieval System) , système automatisé de stockage/déstockage

D/L : station de dépôts/livraison

AGV: véhicule à guidage automatique

FF-flow-rack AS/RS: Free Fall flow-rack AS/RS

PWM : (Pulse Width Modulation) ,Modulation de largeur d'impulsion.

UART: Universal Asynchronous Receiver Transmitter

ICSP: In-Circuit Serial Programming

SRAM: Static Random Access Memory

EEPROM: electrically erasable programmable read only memory

FTDI: Future Technology Devices International

RX : Receiver

TX : Transaction

SPI : (Serial Peripheral Interface Interface) Série Périphérique.

MOSI: Master Out Slave In.

MISO: Master in Slave Out.

IDE : Integrated Development Environment.

RCO: Rapport Cyclique d'Ouverture.

SPST: Single Pole Single Throw.

SPDT: Single Pole Double Throw.

DPST: Double Pole Single Throw.

DPDT: Double Pole Double Throw.

NO: normally open.

NC: normally closed.

RGBA: Red/Green/Amber

IR: rayonnement infrarouge

DIN: Deutsche Industrie Normen

VAC : Volt Alternatif Courant

VDC : Volt Directionnel Courant

DC : Direct current (courant continu)

Remerciement**Dédicace**

Résumé	i
Liste des figures	ii
Liste des tableaux	v
Abréviation	vi

Introduction Générale

Introduction générale	1
-----------------------------	---

Chapitre I: Introduction aux systèmes automatisés de distribution des médicaments

I.1. Introduction	2
I.2. Historique	2
I.3. Présentation globale du système	2
I.3.1. Description	2
I.3.2. Composition	3
I.3.3. Les différents types de machines	3
I.3.4. L'installation	6
I.3.5. Les systèmes de convoyage	6
I.4. Présentation des différents automates	9
I.4.1. Systeme Pharmamat	9
I.4.2. Systeme Apotéka	10
I.4.3. Systeme Movetec	11
I.4.4. Systeme Ariana	12
I.4.5. Systeme Pharmatrack	14
I.4.6. Système Rowa	15
I.4.7. Unguator	16
I.4.8. Rhomeo	17
I.5. Intérêts des automates	18
I.6. Conclusion	21

Chapitre II: Flow rack AS/RS a chute libre 'Apoteka'

II.1. introduction	22
II.2. SECTION 01 : généralités sur les systèmes automatisés de stockage/déstockage.....	22

II.2.1.	Définition	22
II.2.2.	Fonction	22
II.2.3.	Composition	23
II.2.4.	Conception d'un AS/RS	28
II.2.5.	Types d'AS/RS.....	29
II.3.	SECTION 02: Le Flow rack AS/RS a chute libre "Apoteka"	29
II.3.1.	Présentation.....	29
II.3.2.	Composition	30
II.3.3.	Principe de fonctionnement.....	32
II.3.4.	Modélisation discrète du temps moyen de déstockage	34
II.3.5.	Avantages	36
II.3.6.	Inconvénients	36
II.3.7.	Différences entre le FF-flow-rack et flow rack classique	36
II.3.2.	Conclusion	37

Chapitre III: Système de commande Arduino

III.1.	Introduction	38
III.2.	Présentation d'Arduino :	38
III.2.1.	Définition	38
III.2.2.	Bonnes raisons de choisir Arduino	38
III.2.3.	Outils Arduino	39
III.2.4.	Les gammes de la carte Arduino	39
III.2.5.	La carte Méga	40
III.3.	Présentation d'Arduino Méga 2560	40
III.3.1.	Caractéristiques de l'Arduino Méga 2560	41
III.3.2.	Synthèse des caractéristiques	42
III.3.3.	Alimentation	42
III.3.4.	Mémoire	43
III.3.5.	Entrées et sorties numériques	43
III.3.6.	Broches analogiques	44
III.3.7.	Autres broches	44
III.3.8.	Communications	45
III.3.9.	Principaux fonctionnements	45
III.4.	Étude de la partie logicielle.....	45
III.4.1.	Plateforme de programmation Arduino	45
III.4.2.	Proteus Professional	48

III.5.	Les servomoteurs	48
III.6.	Le module PCA9685.....	52
III.7.	Capteur d'obstacle infrarouge à 4 canaux	56
III.8.	Moteurs à courant continu.....	57
III.9.	Relais.....	60
III.10.	Conclusion	61

Chapitre IV: Conception et réalisation

IV.1.	Introduction	62
IV.2.	Les étapes de la réalisation	62
IV.3.	Description du système	62
IV.3.1.	Dispositifs matériels et technologie	62
IV.3.2.	Maquette de système	63
IV.3.3.	Convoyeur	65
IV.4.	Choix des composants et justification	67
IV.5.	Simulation et test avec Arduino.....	72
IV.5.1.	Branchement	72
IV.5.2.	Simulation	76
IV.5.3.	Programmation avec l'Arduino	78
IV.6.	Test et réalisation finale	81
IV.7.	Réalisation finale du système	82
IV.8.	Conclusion.....	83
	Conclusion générale	84
	Références Bibliographiques	85

Annexe

Introduction générale :

L'automatisation se développe de plus en plus au sein des officines. En effet, malgré une période "grise" pour la profession, durant laquelle la réduction des frais généraux sont les principaux soucis des titulaires, les sociétés d'automatisation savent tirer leur épingle du jeu en proposant des solutions adaptées à toutes les typologies d'officines, petites, moyennes, grandes, ou de centre commercial (avec leur grande largeur de stock), ils ont su sur certains points s'adapter aux besoins croissants des pharmaciens en allant vers des systèmes plus autonomes (rangement des boîtes automatiques), plus compactes et moins coûteuses, pour cela plusieurs solutions technologiques d'AS/RS sont apparues afin d'automatiser la dispensation des médicaments. Parmi ces solutions le système automatisé « APOTEKA » qui est basé sur le Flow-rack AS/RS à chute libre ou bien FF-flow-rack AS/RS.

Le flow rack AS/RS à chute libre (FF-flow-rack AS/RS) est une nouvelle variante du flow-rack AS/RS, il utilise un opérateur où une seule machine S/D pour les opérations de stockage, une combinaison entre le mouvement de chute libre et un convoyeur de transport pour les opérations de déstockage. Ce système automatique a été mis en place non seulement pour éliminer des erreurs de cueillette et gagner du temps dans la préparation des ordonnances, mais également pour créer un déroulement des opérations plus efficace de pharmacie, en plus ce système à haute densité d'automation élimine le besoin de rayonnage statique.

Notre but dans ce projet est la conception et la réalisation d'un système automatisé de distribution de médicaments, commandé par un microcontrôleur ATmega256 (Arduino mega2560), ce qui est moins coûteux qu'un automate programmable. Ce système est de type 'flow rack à chute libre' (casiers inclinés) où des servomoteurs provoquent la chute libre des produits (boîtes de médicaments), et les capteurs d'obstacle infrarouges indiquent la présence ou non des boîtes.

Ce mémoire est composé de quatre chapitres :

Le premier chapitre présente une introduction aux systèmes automatisés de distribution de médicaments, on passant en revue la plupart des systèmes qui existent.

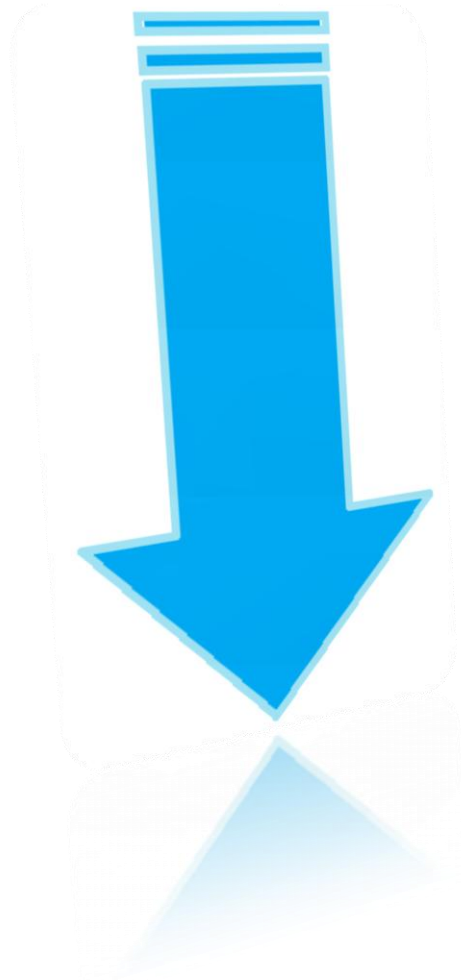
Le deuxième chapitre se compose de deux grandes parties. Nous réservons la première partie à la présentation générale des systèmes automatisés de stockage/déstockage. La deuxième partie est consacrée à la présentation du FF-flow-rack AS/RS plus précisément.

Le troisième chapitre est une étude approfondie sur le Système de commande Arduino en focalisant sur l'Arduino mega2560 utilisé pour commander notre système.

Le quatrième chapitre est consacré aux différentes parties de notre système (mécanique, électronique, électrique) ainsi que les étapes de notre réalisation.

Ce mémoire s'achève par une conclusion générale qui présente le bilan de ce travail ainsi que les perspectives envisagées.

Chapitre I



**Introduction aux systèmes automatisés de
distribution des médicaments**

I.1. Introduction :

Les officinaux, à l'affût de tout progrès technologique visant à améliorer la gestion et la qualité de leur outil de travail, se tournent actuellement vers une gestion des stocks entièrement automatisée.

Ce chapitre concerne les systèmes de distribution des médicaments et nous allons voir aussi les différents automates et les robots qui permettent de nos jours la collecte et le convoyage des spécialités jusqu'au comptoir, limitant ainsi les déplacements du personnel et permettant un meilleur contact avec le patient.

I.2. Historique :

Le premier automate à apparaître sur le marché français au début des années 1990 a été le Pharmamat. Il était développé par la société PlusInfos, avec à son capital la société autrichienne PEEM qui fournissait déjà à l'époque des automates pour les grossistes-répartiteurs. Ce sera le seul automate sur le marché français jusqu'à la fin des années 90 avec une trentaine de pharmacies équipées. Mais malgré la qualité du produit, le problème du coût a été un frein, le retour sur investissement n'étant pas évident pour les pharmaciens.

Mais tout va véritablement commencer pour le marché de l'automatisation hexagonal en 1999 quand deux français, Jean-Louis Connier et Michel Poux, fondent la société Mekapharm et sortent leur produit phare l'Apoteka. Leur idée a été de créer un automate facilement démontable et transportable, contrairement aux grosses installations destinées à la base au milieu industriel de l'autrichien PEEM.

Le marché va s'accroître de 2000 à 2007, d'autres fournisseurs toujours présents vont apparaître comme Mach4 ou Pharmax. On comptera jusqu'à une douzaine de vendeurs pendant ces années fastes de l'automatisation, avec Mekapharm qui profite de son avance en se taillant environ près de 40% de parts de marché. A partir de 2008 la crise va ralentir le marché, le soufflé retombe et à coup de retrait ou de rachat cinq fournisseurs représentent la quasi-totalité du marché : Mekapharm, ARX, Mach4, Pharmax et Tecnilab.

Aujourd'hui encore, le marché est marqué par l'antagonisme entre robot et automate, malgré l'offre de systèmes combinés chez quasiment tous les fournisseurs. Les automates sont historiquement français avec les fabricants Mekapharm et Pharmax. Les fabricants de robots sont pour la plupart allemands et se sont adaptés au marché français : ARX et Mach4. Parti de l'automate en 2000, le fabricant italien Tecnilab s'est lancé ensuite dans la robotisation. Il vante une gamme complète adaptée à chaque pharmacie. [1]

I.3. Présentation globale du système :

I.3.1. Description :

Les automates de la répartition sont des machines industrielles très solides et très lourdes. Les officines n'ont pas toutes un sol renforcé permettant d'accueillir de telles machines.

Tous les automates d'officine (ou presque) présentent des plateaux inclinés, chaque module est divisé en colonnes : il n'y a pas de plateau. Les colonnes constituent deux pans inclinés qui forment un « V » inversé. De chaque côté de l'automate, des cartons de boîtes placés en vis-à-vis servent à remplir l'automate au fur et à mesure de son fonctionnement. Les boîtes sont empilées les unes sur les autres et c'est la boîte du dessous qui est éjectée par une courroie. Nous le verrons, que se soit en officine ou dans la répartition, c'est toujours la gravité qui entraîne les conditionnements au plus près du tapis roulant.

L'automate est réservé aux moyennes rotations, les grosses rotations sont toujours traitées manuellement. Le remplissage de l'automate s'effectue également manuellement à partir d'un stock se trouvant en vis-à-vis des colonnes de l'automate à remplir. Après éjection, les boîtes sont convoyées par un tapis central jusqu'à une caisse destinée à la pharmacie ayant passé commande.

I.3.2. Composition :

Le système pour fonctionner requiert différents éléments :

- ✓ La « machine », où sont stockés les médicaments, chargée d'éjecter les boîtes

Commandées :

- ✓ Le système de convoyage : tapis roulant, ascenseur, toboggan, pneumatique ...
- ✓ Un ou des points(s) de réception ;
- ✓ Un système informatique gérant l'automate et servant d'interface avec le logiciel utilisé par la pharmacie au comptoir.

I.3.3. Les différents types de machines :

Parmi les différentes machines sur le marché trois types coexistent: les robots, les automates et des systèmes mixtes (robot /automate).

a. Systèmes à automates :

Description :

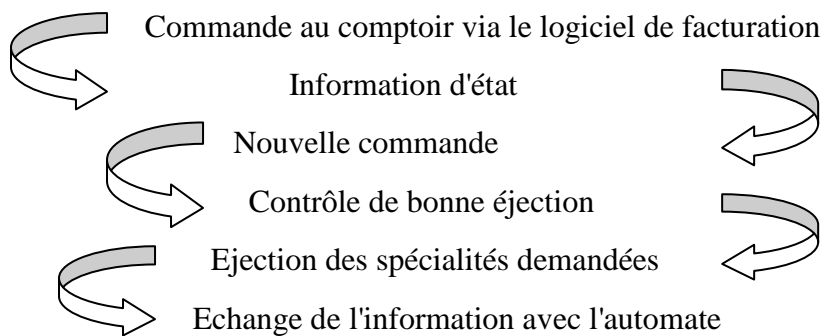
Un automate se présente comme un assemblage de plusieurs «armoires» dont les « étagères » sont inclinées. Chaque « armoire » est appelée module. Chaque module comprend plusieurs «étagères », ou plateaux, divisées en « goulottes » appelées canaux.

Chaque canal correspond à un médicament et sa largeur peut varier : chaque rangée peut donc avoir un nombre de canaux différent. Plusieurs modules peuvent être assemblés et ils peuvent être disposés côte à côte ou en vis-à-vis, entre les deux vient alors se placer un tapis roulant. Les boîtes de médicaments sont introduites d'un côté du module et tombent par gravité vers le tapis central. Une petite butée retient les boîtes dans l'appareil. Cette butée est

franchie pour permettre l'éjection des boîtes par différents moyens. L'éjecteur est un levier actionné par l'air comprimé ou un électro-aimant : il pousse la boîte vers le haut pour franchir la butée. Celle-ci peut aussi être vaincue par un « doigt » soulevant la boîte.

Le « doigt » correspond au système de « picking » utilisé par certains robots.

Schéma de principe :



Automates contre robots :

Les médecins français prescrivant plusieurs boîtes et parce que les premiers fabricants français s'étaient lancés dans le développement d'automate, c'est ce type d'installation qui était recherchés pour leur rapidité. Mais la tendance s'inverse notamment grâce à la baisse des prix des robots.

Mais chaque fabricant défend sa vision et sa machine fer de lance, automate pour les uns et robots pour les autres. Les allemands, traditionnellement tournés vers le robot, voient dans la baisse du prix de leur machine et la réduction du nombre de lignes prescrites sur les ordonnances une fin progressive de l'épiphénomène du marché de l'automate français.

Pour les automaticiens, l'avenir est dans les grosses pharmacies ayant une capacité de délivrance rapide permise par les automates. D'autant plus que les automates permettent des installations à des prix relativement bas, en n'automatisant qu'une partie du stock contrairement aux lourds investissements des robots auprès desquels la gestion de la totalité du stock est souvent entièrement déléguée. [1]

De l'avis de tous les fabricants, la décision de s'automatiser par un robot ou un automate doit avant tout reposer sur un véritable audit et une bonne définition.

b. Systèmes à robots :

Les robots sont composés d'étagères horizontales et d'un ou deux bras articulés qui se déplacent sur les axes X et Y. La délivrance et le rangement des médicaments se font par l'intermédiaire de la tête du robot. Celle-ci se déplace sur un rail vertical et un rail horizontal.

Le rail vertical permet le déplacement de la tête du robot à différentes hauteurs et le rail horizontal permet le déplacement du rail vertical.

Le rangement est semi-automatique ou automatique : les boîtes sont soit scannées manuellement une à une avant d'être déposées sur un tapis puis rangées, soit déposées sur des étagères ou sur un tapis roulant avant d'être scannées automatiquement par le robot. La délivrance est plus lente qu'avec un automate, le robot gère avec sa pince ou ses pinces selon le fabricant les boîtes une par une. Pour augmenter la vitesse de rangement et de délivrance, le multipicking a été créé. Le robot est capable de transporter plusieurs boîtes à la fois, identiques ou non selon les fabricants, au lieu d'en ranger ou délivrer une à la fois.

c. Systèmes « mixtes » :

Les fabricants allemands ont été les premiers à développer des solutions mixtes, couplant robot et automate. C'est à dire qu'un automate a été greffé au robot, le robot rangeant les boîtes dans l'automate. La délivrance se fait soit grâce au robot pour les faibles et moyennes rotations, soit grâce à l'automate pour les fortes rotations.

Ces appareils permettent d'ajouter aux avantages des robots, ceux des automates, à l'optimisation de l'espace du robot, s'ajoute la rapidité de l'automate. L'officine est alors proche de l'automatisation totale (Figure I.1)



Figure I.1. Système Hybride Robomat.

I.3.4.L'installation :

a. Surface nécessaire :

La surface occupée par l'automate est fonction du nombre de modules dont va s'équiper la pharmacie et de l'automate retenu, mais pour un système moyen, la surface au sol est d'au moins 10 à 20 m². Ceci ne tient compte que des armoires composant l'automate ; à ces valeurs il faudra ajouter la place prise par le système de convoyage (tapis par exemple) et le quai de chargement. Il convient même de doubler cette valeur si on veut tenir compte des étagères à mettre vis-à-vis des modules, afin de stocker des boîtes supplémentaires ne pouvant être contenues dans l'automate. En effet, le remplissage se fera d'autant plus aisément si on les met en regard de l'emplacement prévu dans l'automate. Ceci s'applique bien sûr pour les grosses rotations ; pour les références moins demandées, le canal de l'automate pourra aisément contenir toutes les boîtes, sans avoir à en stocker ailleurs.

Il est également possible d'allouer plusieurs canaux pour le même médicament. Cette technique est utile pour les gros conditionnements et pour les très grosses rotations.

b. Lieu d'installation :

L'automate ou le robot peut se trouver à la cave, au rez-de-chaussée ou même à l'étage. Le système de convoyage s'adaptera alors en fonction de son emplacement. Il existe plusieurs moyens de convoyage, pouvant être combinés.

I.3.5. Les systèmes de convoyage :

Que l'on soit équipé d'un robot ou d'un automate, un système de convoyage est quasiment tout le temps nécessaire afin d'acheminer les médicaments près des comptoirs. C'est un élément très important dans un projet d'automatisation d'une officine, puisqu'il aura un rôle essentiel dans l'efficacité globale du projet.

Pour avoir le minimum de problèmes, le système de convoyage doit être le plus court et le plus simple possible.

Il existe différents systèmes de convoyage tel que :

a. Les ascenseurs :

Ils permettent de monter au rez-de-chaussée lorsque l'automate est au sous-sol. Différents systèmes existent. TECNILAB propose un ascenseur avec son robot qui permet d'acheminer rapidement des commandes même assez lourdes. L'ascenseur est mis en bout de tapis et amène en moins de 5 secondes la commande au rez-de-chaussée.

b. Les systèmes pneumatiques :

Les systèmes pneumatiques fonctionnent ainsi : les médicaments sortant de l'automate ou du robot arrivent dans un réceptacle. Ils sont pris en charge par un piston. Ce type de système implique un compresseur qui envoie le piston au travers d'un tube de pvc ou de plexiglas d'environ 20 cm de diamètre. Celui-ci glisse le long du tube pour déposer les médicaments un à un, à un ou plusieurs points de délivrance. Le piston est ensuite aspiré pour reprendre sa place à sa station d'accueil. [2] La vitesse est de 5 à 8 m/s pour un système pneumatique contre 1 m/s pour un tapis roulant mais il faut compter un délai supplémentaire lié au déclenchement du compresseur.

Un des freins à l'installation de ce système de convoyage est le bruit ajouté à un coût plus important que les autres transitiques. Ce système de convoyage est aussi en perte de vitesse car de plus en plus de grosses boîtes ne peuvent pas passer dans les tubes pour des questions de taille. [3](Figure I.2)



Figure I.2. Système de convoyage pneumatique Airtec de Tecnilab [4].

c. Les nacelles et chariots :

Ce sont des petits paniers qui serviront à collecter la commande et qui les amèneront au point de récupération en suivant un rail.

- **Avantages :**
- ✓ Convoyage d'un grand nombre de médicaments à la fois, même lourds.
- **Inconvénients :**

- ✓ Fiabilité.
- ✓ Lenteur car les chariots doivent sans cesse faire l'aller-retour.

d. Les tapis roulants :

Quand l'automate ou le robot est situé au rez-de-chaussée, les tapis roulants sont utilisés pour permettre le transport horizontal. Un même tapis peut acheminer les ordonnances à différents postes de travail. Pour cela, des aiguillages sont placés sur le parcours ou on utilise des tapis à différentes voies d'acheminement. Ainsi, avec un seul tapis, on achemine les médicaments à différents comptoirs. [2]

Les tapis peuvent fonctionner dans l'absolu à 3 m/s mais dans la réalité la vitesse se situe entre 0,60 et 1 m/s. Dix mètres de tapis équivaudront donc à un temps d'acheminement de 10 à 17 secondes. Une vitesse d'éjection de quelques secondes peut ainsi être largement pénalisée par les temporisations d'acheminement (délai nécessaire entre deux ordonnances pour éviter les mélanges). [5] (Figure I.3)

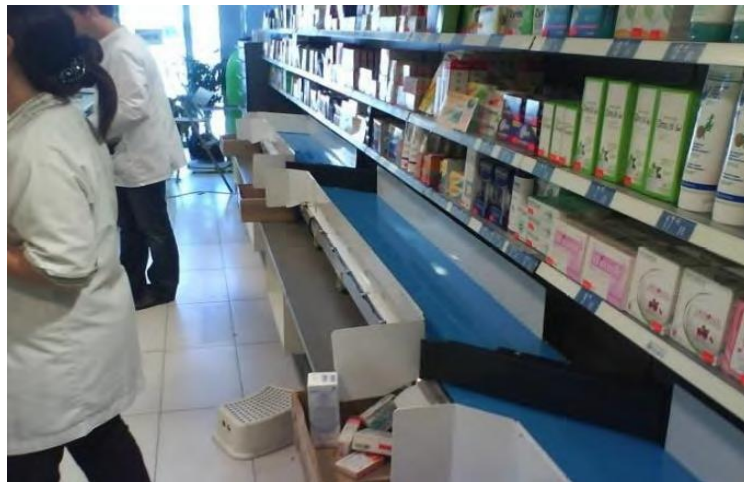


Figure I.3. Tapis roulant avec système d'aiguillage [6].

e. Les glissières :

Les glissières sont le système de transitique idéal lorsque le robot est situé à l'étage. Les glissières ont une forme de toboggan hélicoïdal sur lesquelles les boîtes de médicaments glissent tout simplement par gravité jusqu'au comptoir. (Figure I.4)



Figure I.4. Exemple de glissière [6].

Avantages :

- Simplicité.
- Coût peu élevé.
- Fiable (pas de panne de gravité).

Inconvénients :

- Le stock doit se trouver à un étage supérieur : il faut une personne pour envoyer les produits demandés.

I.4. Présentation des différents automates :

I.4.1. Système Pharmamat :

- **Présentation :**

Le Pharmamat est commercialisé par PlusInfo-PEEM, société autrichienne née en 1990 par la fusion du groupe PEEM et de PlusInfo qui avait depuis le début des années 80 commencé à développer des automates pour les répartiteurs. Le premier modèle de cet automate fut installé pour la première fois en France en 1993, chez M. Yves COMTE à Digne-les-Bains. Depuis, une seconde génération plus évoluée et plus légère l'a remplacé.

- **Les différents modèles :**

Il existe deux modèles:

- Le Pharmamat LW-PEMATI est un robot destiné aux petites pharmacies ayant jusqu'à 3 postes de vente, ou aux produits de moyenne à faible rotation. Il convient si la pharmacie ne traite pas plus de 50 ordonnances à l'heure. Les produits sont récupérés au fur et à mesure par

le robot par un système de « picking » et sont déposés vers un point de transfert (MEDIGLISSI, tapis, ...) [7]. Le robot est un concept qui ne semble plus avoir les faveurs de son constructeur qui conseille plutôt l'automate.

- Le Pharmamat classique : c'est un automate. Il est destiné aux pharmacies qui ont de très gros débits, avec des pics journaliers, ou pour les médicaments de forte à moyenne rotation. Les produits sont éjectés sur un tapis collecteur qui va éventuellement acheminer la commande vers un deuxième tapis fixé au faux plafond (selon la configuration). Le tapis piloté envoie alors les produits vers un convoyeur (toboggan, ascenseur, autre tapis ...), en fonction du poste de vente qui a passé la commande.

- **Avantages et Inconvénients :**

Avantages :

- Système fiable et prouvé auprès des répartiteurs pharmaceutiques et grossistes.
- Déviateurs amortissant les chutes.
- Les 5 niveaux de sécurité.
- Système d'aide au remplissage (surtout pour l'automate).

Inconvénients :

- Le prix élevé.
- Système d'éjection électropneumatique ce qui implique l'utilisation d'un compresseur d'air (à fonctionnement plus ou moins silencieux, mais que l'on peut placer à l'extérieur).
- Déflecteurs pouvant empêcher de prélever manuellement les médicaments en cas de panne.
- Lenteur du robot qui dépose un par un les médicaments commandés (50 ordonnances/heure seulement contre 500 pour l'automate).
- Fiabilité d'éjection du robot : alignement parfait du système d'éjection nécessaire.

I.4.2. Système Apotéka :

- **Présentation :**

L'ApotéKa est le seul automate français. Il est produit par MEKAPHARM et distribué par Phi-CONCEPT. Il est l'archétype même de l'automate et ressemble à son grand frère le Pharmamat (Figure I.5).



Figure I.5. L'ApotéKa [8].

Dans le deuxième chapitre nous allons voir le fonctionnement d'Apotéka, son remplissage et sa gestion.

I.4.3. Système Movetec :

- **Présentation**



Figure I.6. Le MoveTec [9].

Le MoveTec est un robot commercialisé par la société TECNILAB, dont le siège social se trouve en Italie et qui présente des filiales en Espagne, Allemagne et France. Il équipe déjà 14 pharmacies en France.

Il se présente comme une armoire (Figure I.6), des portes transparentes et pliantes à l'avant permettent de détecter et réparer tout problème (comme un produit coincé).

En théorie, on peut assembler jusqu'à huit modules MoveTec ensemble. En pratique, quatre modules ont été installés. Un module « Maître » appelé « Master » pilote jusqu'à sept autres modules « Esclaves » ou « Slaves ». Les modules peuvent être placés en ligne ou face à face. Le module « Master » est équipé d'un ordinateur avec écran vidéo, connecté au logiciel de gestion de la pharmacie. Cet ordinateur va gérer l'activité des différents modules, surveiller la situation interne du système et indiquer les quantités commandées et la nécessité de chargement. A noter que le MoveTec utilise le même logiciel d'échange de données avec le programme utilisé au comptoir que le Pharmamat et qu'il peut être paramétré à distance par modem comme l'Apotéka. Le module « Master » dispose également du bouton d'arrêt d'urgence et c'est lui qui contient le compresseur.

Tous les modules sont en aluminium anodisé, ce qui allège notablement l'ensemble.

- **Avantages et Inconvénients :**

Avantages :

- L'esthétique.
- Certification ISO 9001.
- Tapis mobile, se levant à hauteur des médicaments.
- Largeur de chaque canal très modulable (pas de 12 mm).
- Structure en aluminium anodisé limitant sa masse à 350 kg/m chargé.
- Système de diodes facilitant le chargement.
- L'alerte donnée en cas de seuil de remplissage minimal atteint.

Inconvénients :

- C'est un robot: il est donc plus lent qu'un automate.
- Fiabilité de l'éjection: alignement parfait du système d'éjection nécessaire.
- L'alignement des tapis des différents modules ralentissant l'éjection des boîtes hors du système.

I.4.4. Système Ariana :

- **Présentation :**

L'Ariane distribué par FAHRENBERGER TECHNOLOGIE, a commencé sa carrière sous le nom d'Oscar produit par STIRNBERG AUTOMATION. Développé par Stefan STIRNBERG, jeune ingénieur allemand, l'Ariana est un robot. Il est conçu pour s'adapter à tout type de pharmacie même les plus exiguës. D'ailleurs, il est entièrement monté sur place, il n'y a donc aucun problème pour l'installer dans l'officine.

Chaque module ne fait que 50 cm de largeur et présente une capacité de 400 canaux par mètre de rayonnage. Chaque canal fait un mètre de long et est constitué de deux rails opposés ajustés au mieux en fonction de la largeur des boîtes. Il n'y a pas de plateaux.

Tout en aluminium il ne pèse pas plus de 200 kg à vide, ce qui fait de lui le plus léger de tous les systèmes. La partie stockage et la partie robotique sont totalement indépendantes.

Il se distingue par un rangement vertical breveté permettant de ranger même des produits cylindriques et par l'absence de tapis roulant (Figure I.7). [10][11][12]



Figure I.7. L'Ariana [13].

- **Avantages et Inconvénients :**

- **Avantages :**

- Nuisance sonore réduite car absence de tapis roulant.
- Pas de chute de produit: on peut y placer n'importe quel type de produit.
- Structure très légère, bonne solidité et peu de pièces d'usure, donc grande fiabilité.
- Robot « intelligent ».

- **Inconvénients :**

- C'est un robot : il est plus lent.
- Réglage très technique : les changements d'emplacements sont difficiles.

I.4.5. Système Pharmatrack :

- **Présentation :**

C'est le système le plus récemment mis sur le marché. Il est commercialisé par TECNYFARMA, une société espagnole. Il a été présenté pour la première fois au salon Pharmagora 200 1 (Figure I.8) [14].

Présenté comme compact : 1,5 mètre de largeur sur 2,3 mètres de long soit 3,45 m² occupés au sol, son constructeur préconise de le placer derrière les comptoirs pour éviter l'« utilisation coûteuse de bandes transporteuses » [15], sans libérer de l'espace pour la surface de vente en délocalisant le stock ! Le fabricant précise tout de même que « L'automate peut être installé au premier étage ... » [15], les médicaments tomberont donc par gravité au rez-de-chaussée.



Figure I.8. Le Pharmatrack [16].

- **Avantages et Inconvénients :**

Avantages :

- Le prélèvement des boîtes se fait par soulèvement, ce qui économise l'utilisation d'un compresseur bruyant ou de bobines électromagnétiques.
- Peu de pièces d'usure : un moteur pour le déplacement horizontal et vertical, un moteur pour soulever les médicaments.

Inconvénients :

- Pas d'ordinateur dans le robot, ni de compteur optique pour détecter une panne ou une boîte coincée.
- Pas de système de transport des médicaments en cas de stock délocalisé.
- C'est un robot : il est plus lent qu'un automate.
- Système monobloc et non modulaire comme ses concurrents : capacité limitée.

I.4.6. Système Rowa :

- **Présentation :**

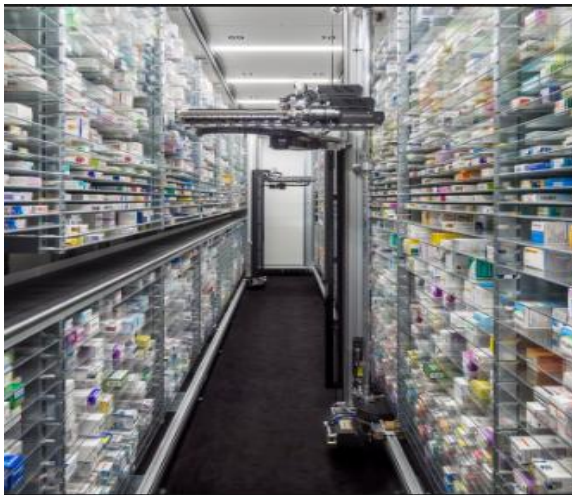


Figure I.9. Intérieur du Rowa [17].



Figure I.10. Bras articulé du Rowa [17].

Rowa est encore un robot allemand, distribué en France par la société ARX. Il a été présenté à Pharmagora 2001, où tout le monde a pu admirer son perfectionnement [18].

Se présente comme une grande armoire avec des étagères des deux côtés et un robot au milieu (Figure I.9). On peut entrer à l'intérieur de l'enceinte par une porte située à une extrémité. Le robot peut faire 2, 2,5 ou 3 mètres de haut et les étagères peuvent aller jusqu'à 9,5 mètres de long (8,5 mètres si la hauteur est de 3 mètres), ce qui permet le stockage de 800 à 1600 articles pour un mètre. La profondeur de Rowa est de 1,30 mètre, mais son installation nécessite 1 mètre supplémentaire [19].

- **Avantages et Inconvénients :**

Avantage :

- Un bijou de technologie : passionnant à voir à l'œuvre.

- S'occupe des produits réfrigérés.
- Nombre de références pouvant être stockées illimité.
- Caméra dans l'enceinte reliée au constructeur, utile en cas de panne.
- Optimisation maximale de la place disponible.

Inconvénients :

- Prix très élevé.
- Semble beaucoup trop lent pour les grosses et moyennes rotations.
- Mode de rangement aléatoire.

I.4.7. Unguator :

- **Présentation**

Unguator est un petit robot permettant de faire des pommades et des mélanges d'onguents facilement (Figure I.11) [20]. Il est commercialisé en France par la société ARX. L'avantage principal est de travailler en système fermé: il suffit de peser les composants à mélanger directement dans les pots spéciaux, puis ils sont mélangés, déjà avec leur couvercle définitif, de façon homogène. Les pots Unguator sont remis directement au patient munis d'un applicateur ou d'une canule (Figure I.12).



Figure I.11. L'Unguator [21].

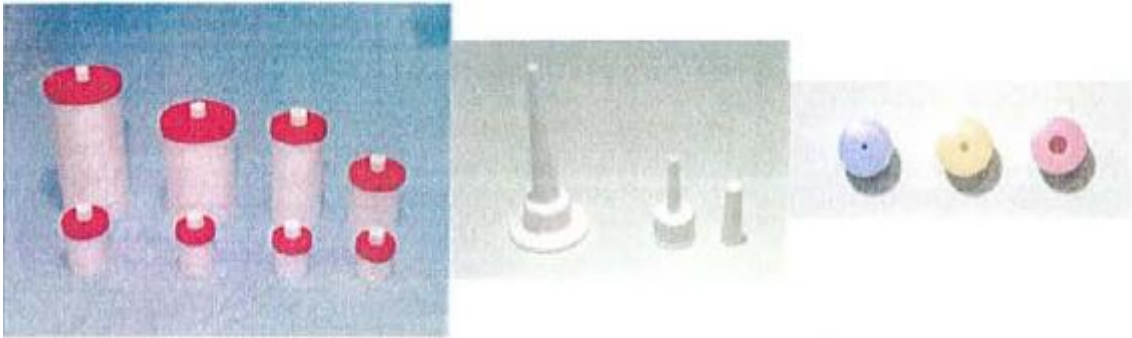


Figure I.12. Accessoires de l'Unguator [22].

- **Avantages et Inconvénients :**

Avantages :

- Gain de temps : pour le mélange, pas de nettoyage.
- Mélanges homogènes : vitesse de rotation standardisée.
- Système clos : pas de dégagements toxiques pour les préparateurs.
- Pas de contamination du produit.

Inconvénients :

- L'aspect de certains mélanges laisse à désirer.
- Utilisation uniquement des pots Unguator.

I.4.8. Rhomeo :

- **Présentation :**

Rhoméo est le premier robot destiné au rangement des tubes granules et des doses homéopathiques. Le prototype a été présenté au salon Pharmagora en 2001 [18]. Il pouvait contenir 2000 tubes dont 2/3 de tubes granules et 1/3 de doses. Ses dimensions réduites 60*62 *120 cm, semblables à une armoire de rangement, permettent de le placer près des comptoirs. Le rangement se fait par lecture des codes barres : l'automate propose alors un emplacement libre où placer le tube. A noter qu'un pavé numérique permet d'entrer le nombre de tubes de la même référence que l'on veut ranger en même temps dans le robot.

Les tubes se fixent tout simplement sur des clips carrés ou ronds selon le modèle choisi. Le système fonctionne avec trois chaînes indexées, type chaîne de vélo de dix mètres chacune, présentant 600 positions tubes ou 800 positions doses : plusieurs combinaisons sont donc possibles. Un autre modèle, plus grand avec quatre chaînes, permettra d'accueillir plus de tubes [4]. Si ce n'est pas suffisant pour les très grosses officines, il y aura la possibilité également de relier deux Rhoméo ensemble. Les deux modèles se branchent par un simple câble à l'informatique de l'officine.

L'intérêt est évident, car la préparation d'une ordonnance homéopathique et le rangement des commandes sont souvent longs et fastidieux. On estime à 10 à 15 minutes en moyenne le temps de préparation d'une ordonnance de 7 références représentant 25 tubes [10]. Pour le rangement il faudra encore 10 minutes de plus. Le temps de ces deux opérations combinées, passe à 5 minutes: on scanne, on range, on scanne et on range ... , sans se soucier du bon emplacement ou de la dilution. Au comptoir, il suffit de taper les références désirées et de venir chercher sa commande ; rien de plus simple et rapide : les tubes sont tombés dans un compartiment. L'appareil permet aussi de mieux gérer informatiquement son stock d'homéopathie, pour l'heure actuelle peu ou pas géré par les officinaux. Les commandes pourront donc être directement télétransmises au fournisseur comme pour l'allopathie. Le gain de temps est appréciable aussi pour le laboratoire qui voit ainsi ses factures téléphoniques diminuer.

- **Avantages et Inconvénients :**

Avantages :

- Gain de temps non négligeable.
- Meilleure gestion du stock.
- Simplicité d'utilisation.
- Grande capacité.

Inconvénients :

- Le remplissage est encore manuel.
- Pas d'acheminement jusqu'au comptoir.

I.5. Intérêts des automates :

- **Arguments commerciaux :**

Les arguments commerciaux développés par les fabricants sont :

- ✓ Baisse périodique des marges.
- ✓ Stagnation du chiffre d'affaires.
- ✓ Concurrence accrue des grandes surfaces et magasins spécialisés.
- ✓ Hors Contexte
- ✓ Augmentation du nombre de références.

L'automate se propose donc d'augmenter la productivité du personnel tout en augmentant leur confort de travail, d'agrandir l'espace de vente, de fidéliser la clientèle... [23][24][25][26][19][20]

- **Rentabilisation du personnel :**

L'automate, et c'est indéniable, permet d'économiser des trajets. D'après une étude menée par M. Yves COMTE dans son officine pour PlusInfo-PEEM [11], chaque membre du personnel parcourt en moyenne 10 km/jour, soit 2 heures de perdues sur 8 heures de travail, donc 20% du temps de travail. L'économie des trajets arrive d'ailleurs en tête des critères de choix (ex-æquo avec l'augmentation de l'espace de vente) d'après une enquête réalisée par M. Philippe COMTE [27]. D'après M. TCHILINGUIRIAN, pharmacien à Contes, l'équipe peut s'occuper du patient et mettre en œuvre ses connaissances pleinement au comptoir. Le pharmacien est à l'origine le spécialiste du médicament et non un manutentionnaire payé pour chercher des boîtes sur une étagère ou dans un tiroir.

L'image auprès de la clientèle s'en trouve donc rehaussée.

Les remplaçants et les nouveaux employés n'ont plus besoin de «déranger» leurs collègues pour savoir où est placé tel médicament, le stress est donc également diminué [11] et le nouveau personnel est tout de suite rentable.

L'automate a permis à M. TCHILINGUIRIAN de réduire les frais de personnel, tout en assurant le service pour 400 clients/jour.

Même en cas d'effectif réduit par arrêt maladie ou vacances, le personnel restant fait face plus aisément aux pics journaliers.

L'automate est souvent présenté comme une amélioration des conditions de travail pour le personnel. Pour mémoire, l'informatique s'est heurtée également à de telles réticences et aujourd'hui elle est pourvoyeuse d'emploi et est parfaitement acceptée. Les automates permettent au pharmacien de se remettre en valeur et de dégager du temps pour le comptoir et des tâches administratives de plus en plus lourdes.

▪ **Rangement des commandes simplifié :**

Le temps de chargement est amélioré de 50% en théorie par rapport au temps de remplissage de tiroirs. En effet, le personnel remplissant l'automate n'est pas dérangé par les vendeurs car tous les automates et robots peuvent être chargés sans arrêter les délivrances. De plus, avec l'apparition de robots « rangeurs » (Apotéka, et dans le futur l'Ariana et certainement la concurrence...) le rangement et la vérification des livraisons est très rapide. De plus, certains appareils, comme le MoveTec, proposent des aides au remplissage très performantes.

Cependant, des systèmes ne simplifient pas vraiment le rangement, comme l'ApotéKa sans robot, et alors le rangement devient fastidieux pour le personnel.

L'autre avantage du rangement dans ces machines est que la première boîte introduite est la première éjectée. Cela permet d'avoir un bon renouvellement du stock et d'éviter au maximum les péremptions.

- **Augmentation de la qualité du service :**

Nous l'avons vu, le personnel reste en permanence au comptoir avec le client, ce qui favorise les échanges et donc la fidélisation. De plus, les patients sont intrigués de la façon dont les médicaments arrivent au comptoir. Parfois même, il vaut mieux garder une part de mystère et laisser libre cours à leur imagination. En effet, d'après M. Philippe COMTE, le fait d'avoir placé une caméra filmant leur Pharmamat était mal perçu par la clientèle [11].

Il vaut mieux les laisser s'imaginer du personnel travaillant à l'arrière pour expédier les boîtes ou un système complexe, pleins d'écrous...

L'automate augmente aussi la rapidité de la délivrance et sa qualité. En effet, le personnel peut être plus concentré sur l'ordonnance et le fait de la saisir sur l'ordinateur permet déjà de bien vérifier la prescription. Il y a un risque moindre de confusion entre deux dosages d'une même spécialité présentant le même conditionnement. Cependant, il faut rester vigilant, car une boîte a pu être mal rangée dans l'automate : il ne faut pas avoir une confiance aveugle dans les produits obtenus [28].

Les renouvellements d'ordonnance sont ultra-simplifiés : il suffit d'appuyer sur une touche et les mêmes médicaments que ceux délivrés le mois précédent parviennent au comptoir en moins de la secondes.

Le temps gagné permet de ne pas perdre les clients trop pressés qui quittent l'officine s'ils estiment le temps d'attente trop long, surtout lorsqu'ils viennent pour des produits hors ordonnance ... [11] [26].

- **Diminution du vol et des pertes :**

L'automate permet également d'être plus vigilant sur le vol à l'étalage des produits de parapharmacie : la présence continue de personnel au comptoir permet de dissuader les éventuels candidats.

D'un autre côté, l'éventuel détournement par le personnel est également diminué : une fois dans l'automate les médicaments sont difficilement accessibles. Certains pharmaciens isolent même l'automate dans une pièce fermée à clef où seul le personnel affecté au remplissage a accès. La fraude qui consiste à prendre une boîte dans un tiroir et à changer le stock, pour que celui-ci soit juste, est impossible. En effet, on peut savoir par recoupement avec le programme de l'automate le nombre précis de boîtes éjectées et faire la comparaison avec le nombre de boîtes vendues.

L'automate évite aussi les erreurs de facturation: le nombre de boîtes délivrées est forcément le même que celui qui est facturé. En effet l'éjection des boîtes ne se fait qu'après validation de l'ordonnance sur le logiciel de facturation.

- **Impact sur le chiffre d'affaires et la marge :**

Le premier intérêt de l'automate est de pouvoir délocaliser le stock et donc de gagner de la place pour l'espace de vente. La surface créée permet d'augmenter le nombre de références

présentées en parapharmacie ou d'améliorer sa présentation afin d'augmenter le pourcentage d'achats impulsifs [29] : 78% des achats sont influencés par l'agencement et l'argumentation du pharmacien. Grâce à l'automatisation, le pharmacien est plus présent pour argumenter. De plus, 75% des pharmaciens s'équipant d'automates réalisent des travaux d'agencement au même moment [11]. On en déduit donc que l'impact sur le chiffre d'affaires ne peut être que positif car un nouvel espace permet également de développer une nouvelle activité qui stimulera le client et permettra d'en attirer de nouveaux [30].

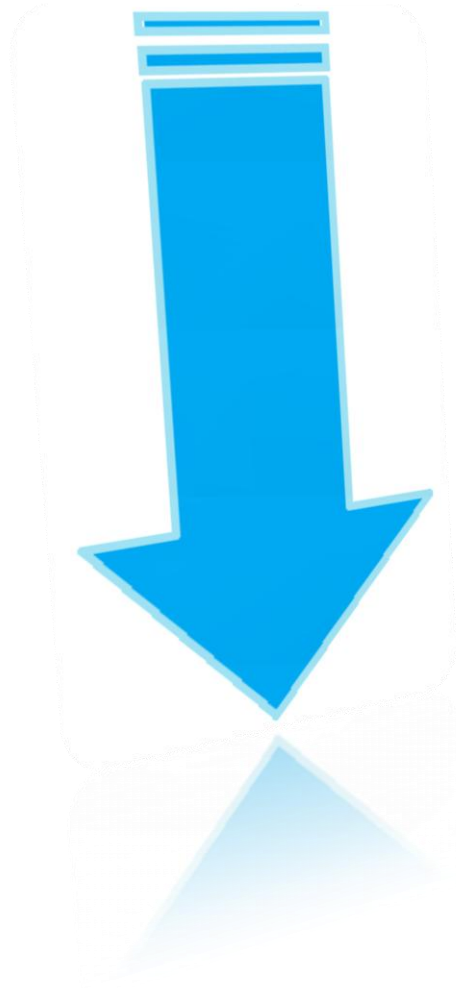
D'après l'enquête de M. Philippe COMTE [26], 6 pharmaciens automatisés sur la ont vu leur chiffre d'affaires augmenter plus que la moyenne nationale en 1999 et 25% d'entre eux ont eu une augmentation de marge.

I.6. Conclusion :

Finalement, Les systèmes de distribution des médicaments permettent à l'équipe officinale de se consacrer entièrement au patient, à sa pathologie et aux conseils associés à son traitement.

Le pharmacien voit son image revalorisée et ses compétences en matière de Santé Publique mises en évidence auprès du public. C'est pourquoi, les officinaux ayant déjà franchi le pas plébiscitent ce nouveau concept et lui promettent un bel avenir.

Chapitre II :



Flow rack AS/RS a chute libre

‘ApotéKa

II.1. introduction :

Les systèmes automatisés de stockage/déstockage (AS/RS) sont de plus en plus souvent adoptés dans les systèmes de production moderne et dans la logistique pour diverses raisons. Ils ont d'abord vu le jour dans les années 1960, en se concentrant initialement sur les lourdes charges de palettes, mais avec l'évolution de la technique, les charges traitées sont devenues plus petites.

Dans les années récentes, plusieurs solutions technologiques d'AS/RS sont apparues pour l'automatisation de la dispensation des médicaments qui est devenue un défi pour répondre aux besoins croissants des pharmacies. Parmi ces solutions le système automatisé « APOTEKA » qui est basé sur le Flow-rack AS/RS à chute libre ou bien FF-flow-rack AS/RS, ceci est relatif au mouvement de chute libre des produits qui le caractérise.

L'objectif de ce deuxième chapitre est de présenter le Flow rack AS/RS à chute libre 'Apoteka' dans lequel s'inscrit notre travail. Nous commencerons, alors, par des généralités sur les systèmes automatisés de stockage/déstockage : définitions, fonction, composition et conception d'un système de stockage AS/RS ainsi que les différents types. Ensuite, le flow rack AS/RS à chute libre est exposé à la section 2.

La section 2 traitera une présentation de système « apoteka » : composition, principe de fonctionnement, remplissage et déstockage ainsi qu'une modélisation discrète du temps moyen de déstockage. Enfin les avantages et les inconvénients du FF-flow-rack et sa comparaison avec le flow rack classique sont présentés.

II.2. SECTION 01 : généralités sur les systèmes automatisés de stockage/déstockage

II.2.1. Définition :

Un AS/RS peut être défini comme une combinaison de matériel et de contrôles qui gèrent automatiquement, stockent et récupèrent des produits à grande vitesse et précision, sans manipulation directe par un travailleur humain. [31] [32] [33]

Selon le Material Handling Institute, un système automatisé de stockage déstockage est défini comme : Une combinaison d'équipements et de systèmes de contrôle qui prend en charge, stocke et déstocke, des produits avec précision, exactitude et célérité sous un certain degré d'automatisation. [34]

II.2.2. Fonction :

L'AS/RS a été conçu dans le but de diminuer l'intervention humaine au maximum dans les opérations de manutention des produits, pièces de rechanges, palettes, matières premières ou même outillage, mis en stock. Permettant de cette manière de réduire considérablement le risque d'erreurs, de manipulations maladroites, ou les interventions humaines en milieu

dangereux. En fonctionnement, un AS/RS reçoit des entités à stocker à un point de dépôt, les identifie et les transporte à l'endroit où ils doivent être stockés, ou bien les déstocke et les livre à une station de livraison. Toutes ces opérations sont contrôlées à l'aide d'un système à base de PLC ou micro-ordinateurs avec des logiciels adéquats.

II.2.3. Composition :

Généralement, un système automatisé de stockage/déstockage inclut les principaux composants suivants :

- Des étagères de stockage (Racks)
- Des allées, avec des supports multi-niveaux donnant accès aux racks.
- Un transstockeur (appelé aussi machine de stockage/déstockage ou machine S/R).
- une station d'entrée/sortie (I/O station), (D/L, P/D station ou quais).
- Des convoyeurs à accumulation
- un système central de surveillance informatique et de communication (système de contrôle). [35] [36]



Figure II.1. Système automatisé de Stockage/Déstockage (AS/RS)

Les racks sont généralement en acier ou en aluminium avec des cellules de stockage qui peuvent accueillir les charges qui doivent être stockées. Les transstockeurs sont des machines de stockage/déstockage entièrement automatisées qui peuvent se déplacer de manière autonome, stocker et déstocker des produits. Les allées sont formées par les espaces vides entre les racks, où les transstockeurs peuvent se déplacer. La station de Dépôt/Livraison D/L

est l'endroit où les Charges récupérées sont déposées, et où les charges entrantes sont prises en charge pour le stockage. [37]

La figure (II.2) Illustre la structure générique et les principaux constituants d'un AS/RS.

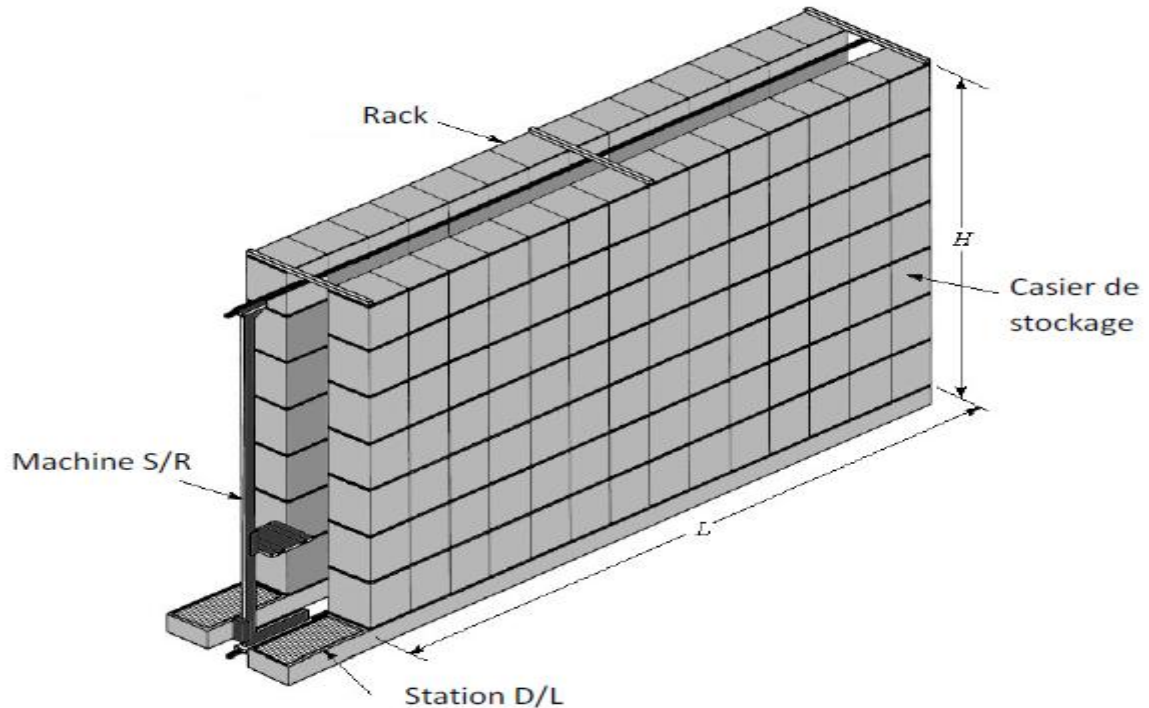


Figure II.2. Principaux constituants d'un AS/RS

a. Rack de stockage :

Un rack est formé d'un ensemble de casiers ou cellules. Dans chaque cellule on peut stocker une ou plusieurs charges. Ces cellules mises l'une à côté de l'autre forment une ligne et l'une au dessus de l'autre forment une colonne (voir figure II.3)

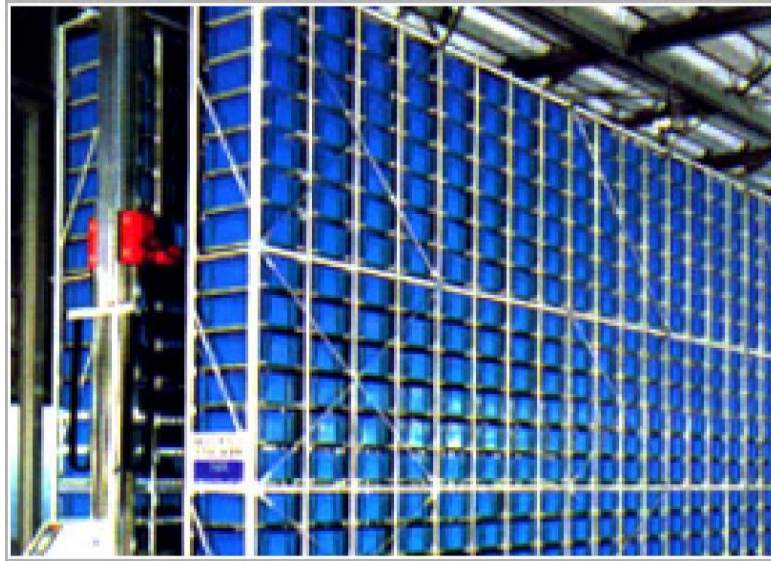


Figure II.3. Racks de stockage

b. Machine de stockage/déstockage (S/R) :

La machine de stockage/déstockage est une structure mobile servant à stocker/déstocker les produits (voir figure II.4). C'est en général un portique en forme de rectangle creux, les cotés horizontaux glissent sur des rails fixés le long des allées (l'un au sol et l'autre au plafond). Tandis que sur les cotés verticaux coulissent un plateau glissant à deux degrés de liberté (linéaires, vertical et horizontal perpendiculaire à l'allée) permettant de charger et décharger les palettes.

L'emploi d'un transstockeur automatique (machine S/R) garantit une grande fiabilité et un haut rendement des opérations de stockage/déstockage. Les machines S/R sont utilisées pour réaliser des opérations de stockage et de déstockage qui peuvent être longues ou même dangereuses pour l'intervention humaine. Ces machines peuvent se déplacer simultanément dans le sens horizontal et vertical. Pour réaliser une opération de stockage, la machine S/R récupère le produit à stocker du point de dépôt/livraison et se déplace vers l'emplacement de stockage où elle dépose ce produit et retourne à son point de repos.

Pour réaliser une opération de déstockage la machine S/R se déplace de la station dépôt/livraison vers le point où se trouve le produit à déstocker, le récupère et le dépose au point de livraison. Dans l'industrie nous pouvons trouver : des transstockeurs à palettes, des transstockeurs pour tiroirs, cartons etc.



Figure II.4.Machine de stockage/déstockage

c. Station dépôt/livraison P/D :

La station de dépôt/livraison appelée aussi station d'entrée/sortie est le point par lequel passent tous les produits entrants ou sortants du système (voir figure II.5). C'est l'interface entre la machine S/R et le système de transfert externe à l'AS/RS. Le transfert des produits de l'AS/RS aux autres systèmes de production se fait soit manuellement soit en utilisant des systèmes de transfert tel que : les AGV, les convoyeurs.

En général, on utilise des convoyeurs à accumulation (ce type de convoyeur autorise un mouvement du produit par rapport au convoyeur c à d si on bloque un produit sur le convoyeur, l'ensemble des autres produits continuent à se déplacer sauf s'ils sont arrêtés dans leurs progression par un produit arrêté).



Figure II.5. Station dépôt/ livraison

d. Système de contrôle :

Le système de contrôle a pour mission de coordonner et superviser le fonctionnement de tous les équipements en temps réel. Nous pouvons par exemple voir sa nécessité dans le stockage aléatoire car il indique à la machine S/R l'emplacement libre pour le stockage, et la guide vers l'emplacement du produit à déstocker. Le système de contrôle est composé de trois entités principales :

- Les calculateurs, sont des ordinateurs à base de microprocesseurs dotés de mémoire et de Puissances de calculs plus ou moins grandes.
- Les logiciels, développés pour chaque type d'opération, sont le coeur du système de contrôle. De leurs puissances dépend le degré de performances du système.
- Les bases de données, les systèmes de transmission d'information, les capteurs...représentent la troisième entité. (Voir figure II.6)



Figure II.6. Système de contrôle

II.2.4. Conception d’un AS/RS :

Le Material Handling System Institute ainsi que Bozer et White [38] dressent une liste de paramètres qui doivent être pris en considération pour une bonne conception d’un système AS/RS. Parmi les plus importants paramètres, nous citons :

- Dimensions de la charge : largeur, hauteur, profondeur, poids
- Dimensions du casier : largeur, hauteur, profondeur
- Nombre total de casiers
- Structure de l’étagère : charge utile, charge admissible,...
- Rendement : le nombre de stockage/déstockage par unité de temps
- Durée de cycle : cycle simple, cycle double, cycle multi-adresses,...
- Taux d’utilisation des machines.

En concevant un AS/RS, deux principes fondamentaux doivent être considérés [39] : Le premier est que l’AS/RS doit fonctionner, au maximum, à 85% de ses capacités maximales. Le second est qu’il faut considérer chaque cas de conception d’AS/RS comme un problème unique.

II.2.5. Types d’AS/RS:

Il existe dans la littérature une multitude d’AS/RS, évoqués dans les différents travaux effectués par la communauté scientifique, et appuyés par l’établissement effectif de ces systèmes dans l’industrie d’aujourd’hui. Cette pluralité de systèmes est due à leur différence architecturale, structurelle et à leurs domaines d’applications. Dans ce qui suit nous allons en recenser les plus connus :

- Les AS/RS à charge unitaire
- Les AS/RS multi allées
- Les AS/RS à racks glissants
- Les AS/RS à mini charge
- Les AS/RS à carrousel
- Les AS/RS à personne embarquée
- Les AS/RS à étagères profondes
- Les AS/RS à convoyeur gravitationnel
- Les AVS/RS
- L’AS/RS à convoyeur gravitationnel à une seule machine
- L’AS/RS à chariots gravitationnels bidirectionnels ou le bi-directional flow-rack AS/RS

II.3. SECTION 02: Le Flow rack AS/RS a chute libre “Apoteka”

II.3.1. Présentation:

L'ApotéKa est le seul automate français. Il est produit par MEKAPHARM et distribué par Phi-CONCEPT. Il est l'archétype même de l'automate et ressemble à son grand frère le Pharmamat, il est utilisé pour l'automatisation de la distribution des médicaments dans les pharmacies Ce système automatique a été mis en place afin de permettre aux pharmacies de gagner du temps dans la préparation des ordonnances et de venir chercher les commandes par la suite. Aussi il présente également l'avantage de donner aux pharmaciens plus de temps à consacrer aux conseils pour l'usage et la prise des médicaments prescrits. Figure II.7.



Figure II.7. Photo réelle d’un flow-rack AS/RS à chute libre « APOTEKA ».

II.3.2. Composition :

APOTEKA consiste en un flow-rack AS/RS à Chute Libre il constituait de colonnes inertes où collecteurs inclinés (casiers inclinés), pour que la force de gravité permette le glissement et la chute des boîtes (médicaments). Ces colonnes sont placées dans des armoires qui peuvent être associées, et chaque armoire est appelée un module. Chaque colonne regroupant un ensemble d’emplacements pouvant contenir plusieurs produits identiques placés les uns à la suite des autres (les mêmes SKU: Stock Keeping Unit). Une station de dépôt situé sur la face de stockage, et une station de livraison située sur la face de déstockage. La figure (II.8) présente la constitution détaillée d’un flow-rack AS/RS à chute libre.

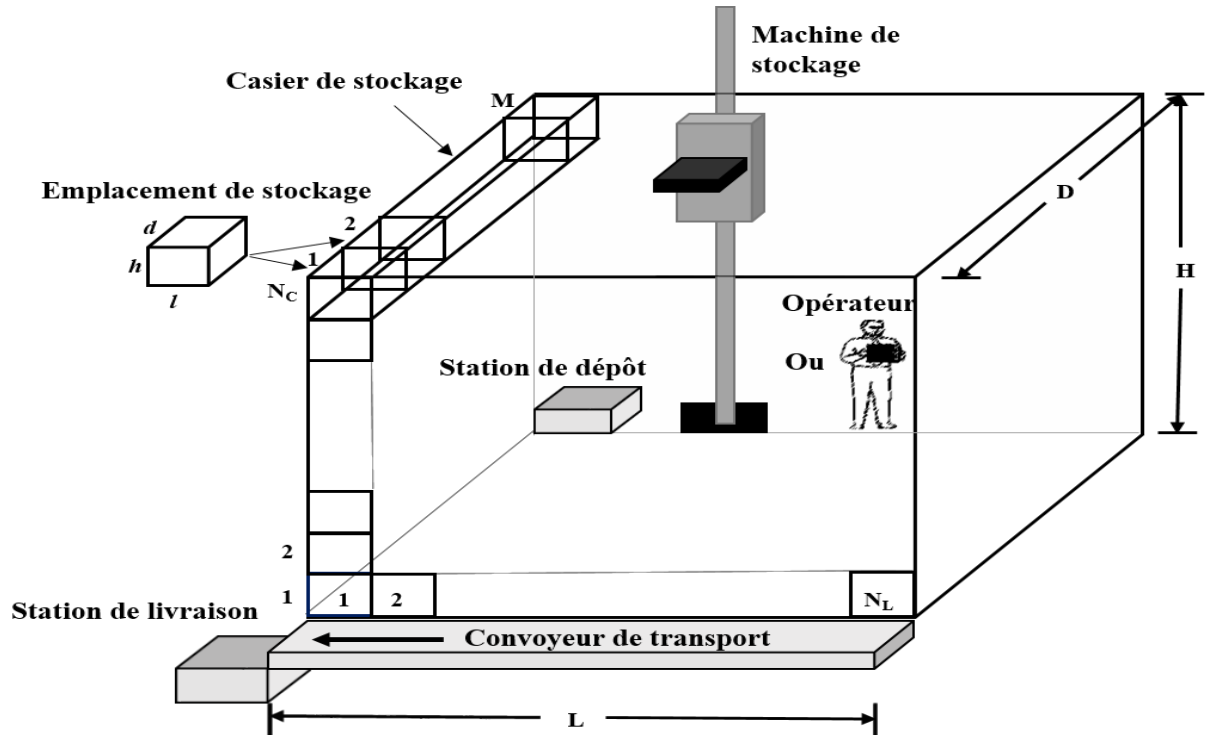


Figure II.8. Constitution typique d’un flow-rack AS/RS à chute libre

L’automate ApotéKa présente une version réfrigérée de sa société Mekapharm, le tableau (II.1) présente quelques caractéristiques :

Fournisseur	Mekapharm
Modèle(s)	ApotéKa
Stockage	A plat
Hauteur	Jusqu’a 2.80 m
Longueur	1.125 m
Profondeur	0.5 ou 0.9 m
Particularités	Version réfrigérée

Tableau II.1. Quelques caractéristiques d’ApotéKa

II.3.3. Principe de fonctionnement:

C'est sans doute le plus simple des automates dans son fonctionnement, ce qui en fait certainement un des plus fiables. L'automate se présente sous forme de modules à assembler et peut se placer n'importe où dans la pharmacie. Il n'a pas de déflecteurs pour amortir les chutes des boîtes, ce qui permet un accès aisé par l'avant aux boîtes en cas de panne. Il faut également penser à mettre les produits lourds et fragiles au plus près du tapis de convoyage, car la chute est rude. (Les tapis utilisés font 40 cm de largeur et sont disposés de telle façon à ne pas entraver la circulation dans les endroits qu'ils traversent. [40])

Son principal point fort, par rapport aux autres appareils, est le système d'éjection électromagnétique qui dispense d'installer un compresseur (bruyant). Quant aux moteurs entraînant les tapis, ils sont carénés pour réduire leur bruit en fonctionnement.

L'automate est couplé au logiciel de vente via le port série: on échange code CIP, quantité, etc. Un onduleur permet jusqu'à 4 heures d'autonomie à l'automate en cas de coupure électrique. Cependant, les tapis seront paralysés par la coupure de l'alimentation, à moins d'avoir un générateur...

Les canaux sont composés de « cornières » particulières permettant un bon calage des spécialités et évitant tout accordéon dans une file.

a. Remplissage :

Le canal à remplir s'affiche uniquement sur l'écran de l'ordinateur de contrôle de l'ApotéKa qui fournit les coordonnées du canal correspondant à chaque référence. Les spécialités sont rangées par ordre alphabétique en deux groupes : les produits lourds ou fragiles et les produits « normaux ». Ce classement permet de retrouver facilement les boîtes en cas de panne.

L'aide visuelle au remplissage manuel sur les armoires de l'automate est inexistante pour l'instant, mais un robot rangeur est disponible. Autant dire que l'achat de ce robot rangeur devient presque obligatoire, car sans lui le rangement d'une commande devient fastidieux pour quelqu'un de non aguerri. Il faut donc bien mettre les spécialités en face de leur emplacement pour pouvoir remplir plus aisément l'Apotéka" au fur et à mesure qu'il se vide (Figure II.10).



Figure II.9. Côté remplissage de l'ApotéKa

Le principe du robot rangeur est le suivant : un module supplémentaire pouvant contenir environ 800 boîtes va servir de stock tampon. Le déballage et la vérification des commandes passées aux grossistes sont d'autant plus aisés: il suffit de scanner une boîte, une diode s'allume devant un canal libre du stock tampon (choisit au hasard des disponibilités par le système de rangement). Une fois le bon de livraison vérifié, le robot va se charger de remplir l'automate à partir du stock tampon. Raffinement supplémentaire, le robot est équipé d'un télémètre qui permet de faire l'inventaire des produits présents à l'intérieur de l'automate. Toutes les caractéristiques des Boîtes étant connues par l'automate, le robot calcule grâce à la longueur d'une boîte le taux d'occupation de chaque canal.

Enfin, Phi-CONCEPT annonce aussi un nouveau scanner portable d'aide au remplissage ainsi que des étiquettes et des photos pour repérer plus vite l'emplacement des médicaments.[41]

b. Déstockage :

L'éjection se fait par l'excitation de l'électro-aimant du casier contenant le produit demandé. Ceci provoque la descente en chute libre du produit.

La sortie d'un produit va induire le glissement progressif par gravité de tous les autres produits dans le casier. L'électro-aimant revient à sa position initiale à chaque éjection. Il est à noter, que la détection d'éjection du produit peut se faire à l'aide d'un capteur de sortie, comme indique la Figure II.11. [42]

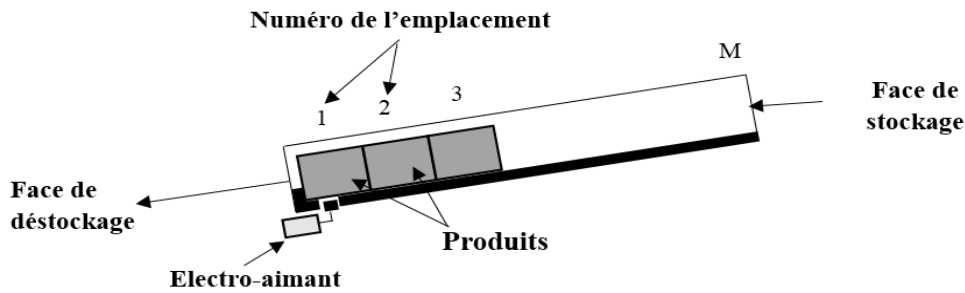


Figure II.10 .Fonctionnement du système d'éjection des produits dans un flow-rack AS/RS à chute libre.

II.3.4. Modélisation discrète du temps moyen de déstockage :

Le déplacement d'un produit dans un flow-rack à chute libre est différent du déplacement d'une machine de stockage/déstockage. En effet, le déplacement résulte d'un déplacement vertical occasionné par la chute commandée du produit, et d'un déplacement horizontal effectué par un convoyeur de transport (Voir Figure II.12). La vitesse de déplacement du convoyeur est supposée constante. En revanche le déplacement vertical n'est pas un mouvement uniforme mais un mouvement uniformément accéléré. [43]

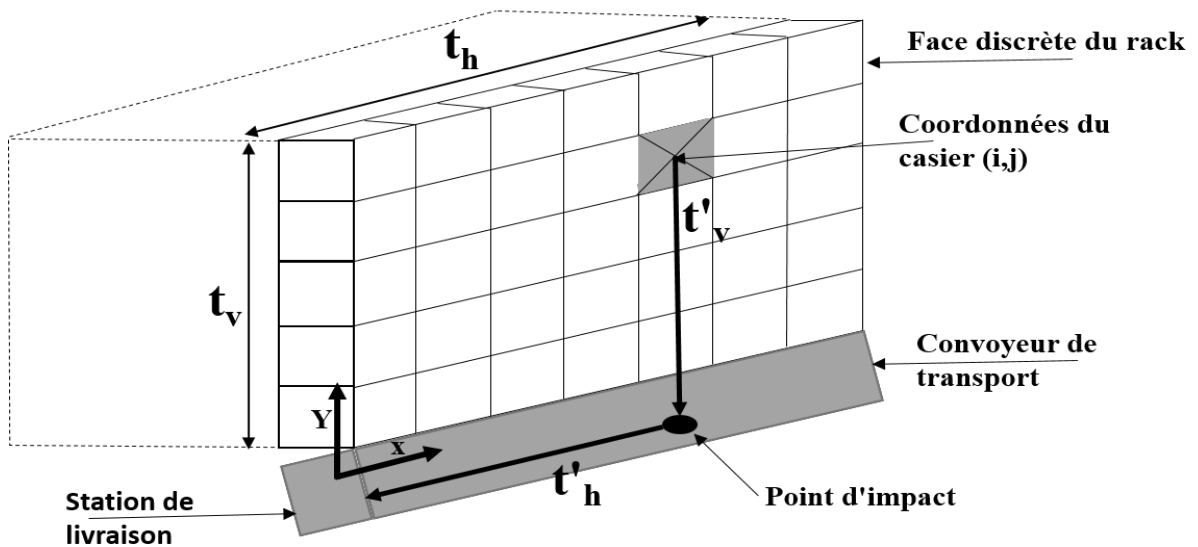


Figure II.11. Représentation des mouvements du produit dans la face de déstockage (face discrète).

Les notations utilisés:

N_V : Nombre de casiers par ligne.

N_H : Nombre de casiers par colonne.

N : Nombre total de casiers. $N = N_V * N_H$

g : L'accélération de la pesanteur. $g = 9.81 \text{ m/s}^2$.

V_c : La vitesse du convoyeur.

t_v : Le temps de déplacement vertical

t_h : Le temps de déplacement horizontal

t_g : Le temps de déplacement global

A_1 : Temps moyen de transport (Moyenne arithmétique).

Le temps global de déstockage d'un produit est le temps nécessaire pour aller du casier de déstockage vers la station de livraison. C'est-à-dire la somme du temps vertical (noté t_v) et du temps de déplacement horizontal (noté t_h).

Considérons le casier de coordonnées (i, j) . Le temps vertical (t_v) que met le produit durant sa chute libre, pour atteindre le convoyeur peut être calculé de l'équation du mouvement de chute libre suivante:

$$\text{D'où : } j = \frac{1}{2} \cdot g \cdot t_v^2$$

$$t_v = \sqrt{j \cdot \frac{2}{g}}$$

Le temps horizontal (t_h) nécessaire au convoyeur pour transporter ensuite le produit vers la station de livraison est :

$$t_h = \frac{i}{V_c}$$

Le temps global de déstockage d'un produit est la somme du temps de déplacement horizontal (t_h) et du temps vertical (t_v).

$$t_g = \frac{i}{V_c} + \sqrt{j \cdot \frac{2}{g}}$$

Si on veut calculer un temps moyen de déstockage pour tous les casiers, on obtient l'expression discrète suivante :

$$A_1 = \frac{1}{NH.Nv} \sum_{i=1}^{NH} \sum_{j=1}^{Nv} \left[\left(\frac{i}{Vc} \right) + \sqrt{j \cdot \frac{2}{g}} \right]$$

II.3.5. Avantages :

En plus des divers avantages offerts par les AS/RS tels que : l'amélioration du débit de sortie, la capacité, la gestion et le contrôle des stocks et la sécurité. Le FF-flow-rack AS/RS a des avantages spéciaux tels que :

- ✓ Simplicité
- ✓ Fiabilité
- ✓ Ejection électromagnétique
- ✓ C'est le plus rapide des automates : jusqu'à une boîte par seconde par éjecteur
- ✓ la possibilité de déstockage simultané de plusieurs produits
- ✓ Réduction du temps d'attente des clients
- ✓ Réduction des erreurs de distribution
- ✓ Réduction de la taille des files d'attente en officine, ce qui laisse plus de temps pour les pharmaciens de conseiller leurs clients.

II.3.6. Inconvénients :

- ✓ Les boîtes placées en haut de l'automate font une chute de plus de deux mètres (l'utilisation du flow-rack AS/RS à chute libre n'est pas adéquate dans le cas de produits volumineux et/ou fragile, car la chute libre du produit peut les endommager).
- ✓ Remplissage fastidieux sans robot rangeur.

II.3.7. Différences entre le FF-flow-rack et flow rack classique :

Par rapport au Flow rack classique AS / RS présenté par Sari et al [14], le Flow rack à chute libre AS / RS offre de nombreux avantages. Les différences entre elles sont résumées dans l'étude décrite dans Metahri et Hachemi [45].

- le débit du FF flow rack peut atteindre une amélioration de 90% par rapport au flow rack classique.
- Avec le FF-flow-rack, la commande de chaque casier est indépendant des autres; par conséquent, de nombreux articles peuvent être récupérés en même temps.

- Dans le flow rack classique, tous les casiers sont inaccessibles une fois la machine S / R en panne. Cependant, dans le FF flow rack, si l'électroaimant d'un casier est hors service, seul ce casier est affecté.
- Le FF flow rack offre une réduction de 40% de l'investissement initial en raison du remplacement de la machine S / R par la combinaison du mouvement en chute libre et du convoyeur de transport (selon Rosenblatt et al ; cela représente environ 40% ou plus selon le coût du système [46]).
- Avec le flow-rack classique, la machine S / R fait des va-et-vient pour récupérer un article. Alors que, avec le FF flow rack, le déstockage d'un produit se fait par la combinaison trop rapide du mouvement en chute libre et du convoyeur. Par conséquent, le nouveau FF flow rack offre une meilleure rapidité de livraison [45]

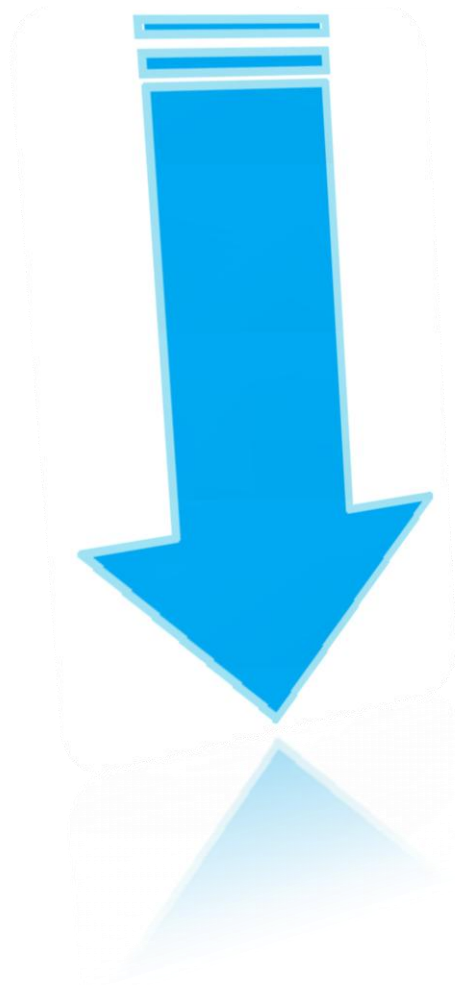
En particulier dans la distribution pharmaceutique, ce système offre une grande rapidité dans la préparation et la livraison des commandes clients (réduction du temps de préparation). De plus, ce système présente l'avantage de donner plus de temps aux pharmaciens pour conseiller leurs clients lors de la prise du médicament prescrit [47].

II.3.2. Conclusion :

Dans ce chapitre, nous avons présenté des généralités sur les systèmes automatisés de stockage/déstockage, ensuite nous nous sommes focalisés sur le FF flow rack qui constitue la base de la solution commerciale Apoteka.

L'Apoteka par rapport aux autres systèmes présente plusieurs avantages aux pharmaciens, comme le montre l'étude comparative présentée.

Chapitre III



Systeme de commande Arduino

III.1. Introduction :

Depuis que l'électronique existe, sa croissance est fulgurante et continue encore aujourd'hui. On va en effet parler d'électronique embarquée qui est un sous-domaine de l'électronique et qui a l'habileté d'unir la puissance de la programmation à la puissance de l'électronique. Nous allons, dans un premier temps, voir ce qu'est le système Arduino et la programmation. Puis nous enchaînerons sur la prise en main de certains composants électroniques compatibles avec l'Arduino.

III.2. Présentation d'Arduino :

III.2.1. Définition :

Arduino est une plate-forme de prototypage d'objets interactifs à usage créatif constituée d'une carte électronique et d'un environnement de programmation. Sans tout connaître ni tout comprendre de l'électronique, cet environnement matériel et logiciel permet à l'utilisateur de formuler ses projets par l'expérimentation directe avec l'aide de nombreuses ressources disponibles en ligne [48].

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.

III.2.2. Bonnes raisons de choisir Arduino :

Il existe pourtant dans le commerce, une multitude de plateformes qui permettent de faire la même chose. Notamment les microcontrôleurs « PIC » du fabricant Micro chip. Nous allons voir pourquoi choisir l'Arduino :

a. Le prix :

En vue des performances qu'elles offrent, les cartes Arduino sont relativement peu coûteuses, ce qui est un critère majeur pour le débutant.

b. La liberté :

Elle constitue en elle-même deux choses : Le logiciel: gratuit et open source, développé en Java, dont la simplicité d'utilisation relève du savoir cliquer sur la souris. Le matériel : cartes électroniques dont les schémas sont en libre circulation sur internet. Cette liberté a une condition : le nom « Arduino » ne doit être employé que pour les cartes « officielles ». En somme, vous ne pouvez pas fabriquer votre propre carte sur le modèle Arduino et lui assigner

le nom « Arduino ». Les cartes non officielles, on peut les trouver et les acheter sur Internet et sont pour la quasi totalité compatibles avec les cartes officielles Arduino.

c. La compatibilité :

Le logiciel, tout comme la carte, est compatible sous les plateformes les plus courantes (Windows, Linux et Mac), contrairement aux autres outils de programmation du commerce qui ne sont, en général, compatibles qu'avec Windows.

d. La communauté :

La communauté Arduino est impressionnante et le nombre de ressources à son sujet est en constante évolution sur internet. De plus, on trouve les références du langage Arduino ainsi qu'une page complète de tutoriels sur le site arduino.cc (en anglais) et arduino.cc (en français).

III.2.3. Outils Arduino :

A présent, rapprochons-nous de « l'utilisation » du système Arduino et voyons comment il se présente. Il est composé de deux choses principales, qui sont : le matériel et le logiciel. Ces deux outils réunis, il nous sera possible de faire n'importe quelle réalisation.

Le matériel : Il s'agit d'une carte électronique basée autour d'un microcontrôleur Atmega du fabricant Atmel, dont le prix est relativement bas pour l'étendue possible des applications.

Le logiciel : Le logiciel va nous permettre de programmer la carte Arduino. Il nous offre une multitude de fonctionnalités.

III.2.4. Les gammes de la carte Arduino :

Actuellement, il existe plus de 20 versions de module Arduino, nous citons quelques-uns afin d'éclaircir l'évaluation de ce produit scientifique et académique : [49]

- Le NG d'Arduino, avec une interface USB pour programmer et usage d'un ATmega8.
- L'extrémité d'Arduino, avec une interface USB pour programmer et usage d'un Microcontrôleur ATmega8.
- L'Arduino Mini, une version miniature de l'Arduino en utilisant un microcontrôleur ATmega168.
- L'Arduino Nano, une petite carte programme à l'aide Porte USB cette version utilisant un microcontrôleur ATmega168 (ATmega328 pour une plus nouvelle version).
- Le Lily Pad Arduino, une conception de minimaliste pour l'application wearable en utilisant un microcontrôleur ATmega168.
- Le NG d'Arduino plus, avec une interface USB pour programmer et usage d'un Tmega168.
- L'Arduino Bluetooth, avec une interface de Bluetooth pour programmer en utilisant un microcontrôleur ATmega168.

- L'Arduino Diecimila, avec une interface USB et utilise un microcontrôleur ATmega168.
- L'Arduino Duemilanove ("2009"), en utilisant un microcontrôleur l'ATmega168 (ATmega328 pour une plus nouvelle version) et actionné par l'intermédiaire de la puissance d'USB/DC.
- L'Arduino Méga, en utilisant un microcontrôleur ATmega1280 pour I/O additionnel et mémoire.
- L'Arduino UNO, utilisations microcontrôleur ATmega328
- L'Arduino Mega2560, utilisations un microcontrôleur ATmega2560, et possède toute la mémoire à 256 KBS. Elle incorpore également le nouvel ATmega8U2 (ATmega16U2 dans le jeu de puces d'USB de révision 3).
- L'Arduino Leonardo, avec un morceau ATmega3U4 qui élimine le besoin de raccordement d'USB et peut être employé comme clavier.
- L'Arduino Esplora : ressemblant à un contrôleur visuel de jeu, avec un manche et des sondes intégrées pour le bruit, la lumière, la température, et l'accélération.

Parmi ces types, nous avons choisi une carte Arduino mega car il a plus d'options en plus.

III.2.5. La carte Méga :

La carte Arduino Méga est une autre carte qui offre toutes les fonctionnalités des précédentes, mais avec des options en plus. On retrouve notamment un nombre d'entrées et de sorties plus importantes ainsi que plusieurs liaisons séries. La figure (III.1) présente la carte Arduino méga.

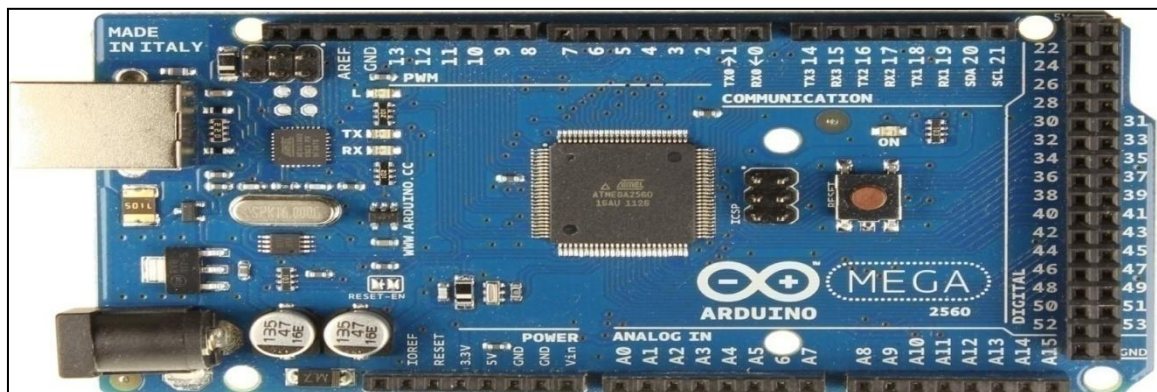


Figure III.1. Photo réelle d'Arduino Méga 2560

III.3. Présentation d'Arduino Méga 2560 :

La carte Arduino repose sur un circuit intégré (un mini ordinateur appelé également microcontrôleur) associée à des entrées et sorties qui permettent à l'utilisateur de brancher différents types d'éléments externes :

- ✓ Côté entrées, des capteurs qui collectent des informations sur leur environnement comme la variation de température via une sonde thermique, le mouvement via un détecteur de présence ou un accéléromètre, le contact via un bouton-poussoir, etc.
- ✓ Côté sorties, des actionneurs qui agissent sur le monde physique telle une petite lampe qui produit de la lumière, un moteur qui actionne un bras articulé, etc.

III.3.1. Caractéristiques de l'Arduino Méga 2560 :

Cette carte dispose :

- 54 broches numériques d'entrées/sorties (dont 14 peuvent être utilisées en sorties PWM (largeur d'impulsion modulée) .
- 16 entrées analogiques (qui peuvent également être utilisées en broches entrées/sorties numériques).
- 4 UART (port série matériel).
- Un quartz 16Mhz.
- Une connexion USB.
- Un connecteur d'alimentation jack.
- Un connecteur ICSP (programmation "in-circuit").
- Un bouton de réinitialisation (reset).

Elle contient tout ce qui est nécessaire pour le fonctionnement du microcontrôleur; pour pouvoir l'utiliser et se lancer, il suffit simplement de la connecter à un ordinateur à l'aide d'un câble USB (ou de l'alimenter avec un adaptateur secteur ou une pile, mais ceci n'est pas indispensable, l'alimentation étant fournie par le port USB). La figure (III.1) présente le brochage de la carte Méga2560.

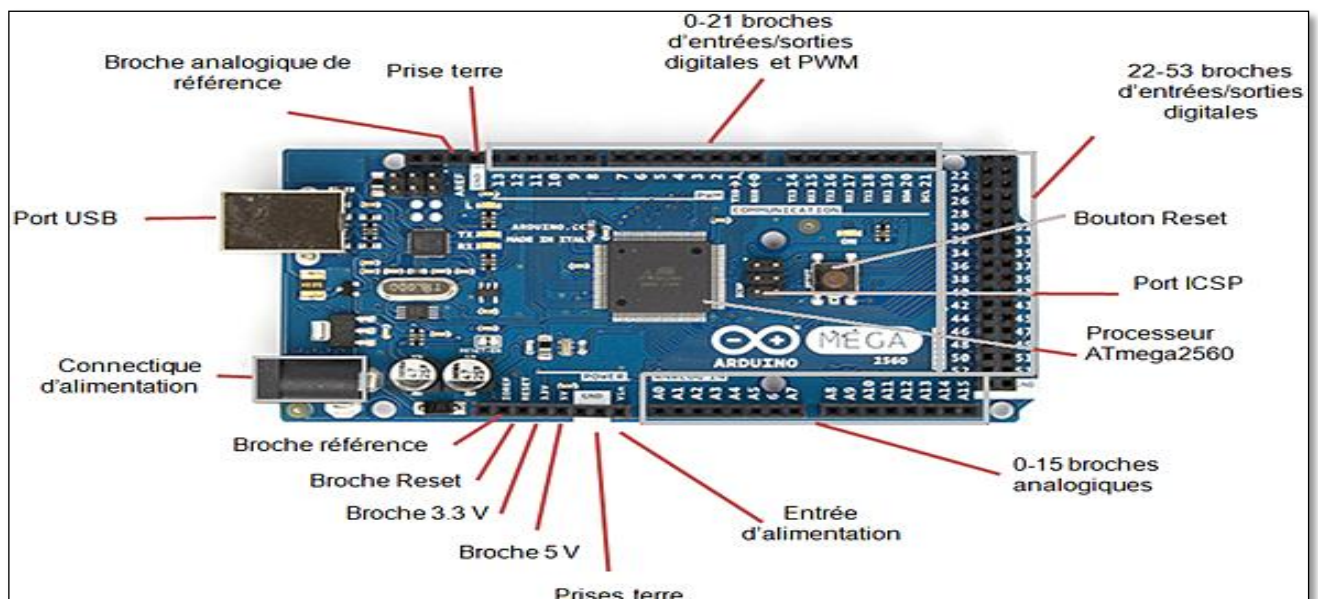


Figure III.2. Brochage de la carte Méga 2560.

III.3.2. Synthèse des caractéristiques :

Le Tableau(III.1) présente les caractéristiques de la carte Arduino méga 2560.

Microcontrôleur	AT Méga2560
Tension de fonctionnement	5V
Tension d'alimentation (recommandée)	7-12V
Tension d'alimentation (limites)	6-20V
Broches E/S numérique	54 (dont 14 disposent d'une sortie PWM)
Broches d'entrées analogiques	16 (utilisables en broches E/S numériques)
Intensité maxi disponible par broche E/S (5V)	40 mA (ATTENTION : 200mA cumulé pour l'ensemble des broches E/S)
Intensité maxi disponible pour la sortie 3.3V	50 Ma
Intensité maxi disponible pour la sortie 5V	Fonction de l'alimentation utilisée - 500 mA max si port USB utilisé seul
Mémoire Programme Flash	256 KB dont 8 KB sont utilisés par le boot loader
Mémoire SRAM (mémoire volatile)	8 KB
Mémoire EEPROM (mémoire non volatile)	4 KB
Vitesse d'horloge	16 MHz

Tableau III.1.Synthèse des caractéristiques

III.3.3. Alimentation :

La carte Arduino Méga 2560 peut-être alimentée soit via la connexion USB (qui fournit 5V jusqu'à 500mA) ou à l'aide d'une alimentation externe. La source d'alimentation est

sélectionnée automatiquement par la carte. L'alimentation externe (non-USB) peut être soit un adaptateur secteur (pouvant fournir typiquement de 3V à 12V sous 500mA) ou des piles (ou des accus). L'adaptateur secteur peut être connecté en branchant une prise 2.1mm positif au centre dans le connecteur jack de la carte. Les fils en provenance d'un bloc de piles ou d'accus peuvent être insérés dans les connecteurs des broches de la carte appelées Gnd (masse ou 0V) et Vin (Tension positive en entrée) du connecteur d'alimentation. La carte peut fonctionner avec une alimentation externe de 6 à 20 volts. Cependant, si la carte est alimentée avec moins de 7V, la broche 5V pourrait fournir moins de 5V et la carte pourrait être instable. Si on utilise plus de 12V, le régulateur de tension de la carte pourrait chauffer et endommager la carte. Aussi, la plage idéale recommandée pour alimenter la carte Uno est entre 7V et 12V.

La carte Arduino Méga 2560 diffère de toutes les cartes précédentes car elle n'utilise par le circuit intégré FTDI USB-vers-série. A la place, elle utilise un Atmega8U2 programmé en convertisseur USB-vers-série. Les broches d'alimentation sont les suivantes :

- **VIN** : la tension d'entrée positive lorsque la carte Arduino est utilisée avec une source de tension externe (à distinguer du 5V de la connexion USB ou autre source 5V régulée). On peut alimenter la carte à l'aide de cette broche, ou, si l'alimentation est fournie par le jack d'alimentation, accéder à la tension d'alimentation sur cette broche.
- **5V** : la tension régulée utilisée pour faire fonctionner le microcontrôleur et les autres composants de la carte (les circuits électroniques numériques nécessitent une tension d'alimentation parfaitement stable dite "tension régulée" obtenue à l'aide d'un composant appelé un régulateur et qui est intégré à la carte Arduino). Le 5V régulé fourni par cette broche peut donc provenir soit de la tension d'alimentation VIN via le régulateur de la carte, ou bien de la connexion USB (qui fournit du 5V régulé) ou de tout autre source d'alimentation régulée.
- **3V3** : une alimentation de 3.3V fournie par le circuit intégré FTDI (circuit intégré faisant l'adaptation du signal entre le port USB de votre ordinateur et le port série de l'ATmega) de la carte est disponible : ceci est intéressant pour certains circuits externes nécessitant cette tension au lieu du 5V). L'intensité maximale disponible sur cette broche est de 50mA.
- **GND**: broche de masse (ou 0V).

III.3.4. Mémoire :

L'ATMéga 2560 à 256Ko de mémoire FLASH pour stocker le programme (dont 8Ko également utilisés par le boot loader). L'ATMéga 2560 a également 8 ko de mémoire SRAM (volatile) et 4Ko d'EEPROM (non volatile - mémoire qui peut être lue à l'aide de la librairie EEPROM).

Le boot loader est un programme préprogrammé une fois pour toute dans l'ATMéga et qui permet la communication entre l'ATMéga et le logiciel Arduino via le port USB, notamment lors de chaque programmation de la carte.

III.3.5. Entrées et sorties numériques :

Chacune des 54 broches numériques de la carte Méga peut être utilisée soit comme une entrée numérique, soit comme une sortie numérique, en utilisant les instructions pin Mode (), digital Write() et digital Read() du langage Arduino. Ces broches fonctionnent en 5V. Chaque

broche peut fournir ou recevoir un maximum de 40mA d'intensité et dispose d'une résistance interne de "rappel au plus" (pull-up) (déconnectée par défaut) de 20-50 KOhms. Cette résistance interne s'active sur une broche en entrée à l'aide de l'instruction `digitalWrite` (broche, HIGH) [50].

De plus, certaines broches ont des fonctions spécialisées :

- **Communication Série :** port Série Serial : 0 (RX) and 1 (TX); Port Série Serial 1: 19 (RX) and 18 (TX); Port Série Serial 2: 17 (RX) and 16 (TX); Port Série Serial 3: 15 (RX) and 14 (TX). Utilisées pour recevoir (RX) et transmettre (TX) les données séries de niveau TTL. Les broches 0 (RX) and 1 (TX) sont connectées aux broches correspondantes du circuit intégré ATmega8U2 programmé en convertisseur USB-vers-série de la carte, composant qui assure l'interface entre les niveaux TTL et le port USB de l'ordinateur.
- **Interruptions Externes:** Broches 2 (interrupt 0), 3 (interrupt 1), 18 (interrupt 5), 19 (interrupt 4), 20 (interrupt 3), et 21 (interrupt 2). Ces broches peuvent être configurées pour déclencher une interruption sur une valeur basse, sur un front montant ou descendant, ou sur un changement de valeur. Voir l'instruction `attachInterrupt()` pour plus de détails.
- **Impulsion PWM (largeur d'impulsion modulée):** Broches 0 à 13, fournissent une impulsion PWM 8-bits à l'aide de l'instruction `analogWrite ()`.
- **SPI (Interface Série Périphérique):** Broches 50 (MISO), 51 (MOSI), 52 (SCK), 53 (SS) ; ces broches supportent la communication SPI (Interface Série Périphérique) disponible avec la librairie pour communication SPI. Les broches SPI sont également connectées sur le connecteur ICSP qui est mécaniquement compatible avec les cartes Uno, Duemilanove et Diecimila.
- **I2C:** Broches 20 (SDA) et 21 (SCL) ; supportent les communications de protocole I2C (ou interface TWI (TwoWire Interface - Interface "2 fils"), disponible en utilisant la librairie `Wire/I2C` (ou TWI - Two-Wire interface - interface "2 fils") . Noter que ces broches n'ont pas le même emplacement que sur les cartes Uno, Duemilanove ou Diecimila.
- **LED:** Broche 13 ; il y a une LED incluse dans la carte connectée à la broche 13. Lorsque la broche est au niveau HAUT, la LED est allumée, lorsque la broche est au niveau BAS, la LED est éteinte.

III.3.6. Broches analogiques :

La carte Mega2560 dispose de 16 entrées analogiques, chacune pouvant fournir une mesure d'une résolution de 10 bits (sur 1024 niveaux soit de 0 à 1023) à l'aide de la très utile fonction `analogRead()` du langage Arduino. Par défaut, ces broches mesurent entre le 0V (valeur 0) et le 5V (valeur 1023), mais il est possible de modifier la référence supérieure de la plage de mesure en utilisant la broche AREF et l'instruction `analogReference()` du langage Arduino. Les broches analogiques peuvent être utilisées en tant que broches numériques.

III.3.7. Autres broches :

Il y a deux autres broches disponibles sur la carte :

- **AREF :** Tension de référence pour les entrées analogiques (si différent du 5V). Utilisée avec l'instruction `analogReference()`.

- **Reset** : Mettre cette broche au niveau BAS entraîne la réinitialisation (= le redémarrage) du microcontrôleur. Typiquement, cette broche est utilisée pour ajouter un bouton de réinitialisation sur le circuit qui bloque celui présent sur la carte.

III.3.8. Communications :

La carte Arduino Mega2560 dispose de toute une série de facilités pour communiquer avec un ordinateur, une autre carte Arduino, ou avec d'autres microcontrôleurs.

L'ATmega2560 dispose de quatre UARTs (Universal Asynchronous Receiver Transmitter ou émetteur-récepteur asynchrone universel en français) pour communication série de niveau TTL (5V) et qui est disponible sur les broches 0 (RX) et 1 (TX). Un circuit intégré ATmega8U2 sur la carte assure la connexion entre cette communication série de l'un des ports série de l'ATmega 2560 vers le port USB de l'ordinateur qui apparaît comme un port COM virtuel pour les logiciels de l'ordinateur. Le code utilisé pour programmer l'ATmega8U2 utilise le driver standard USB COM, et aucun autre driver externe n'est nécessaire.

Le logiciel Arduino inclut une fenêtre terminal série (ou moniteur série) sur l'ordinateur et qui permet d'envoyer des textes simples depuis et vers la carte Arduino. Les LEDs RX et TX sur la carte clignote lorsque les données sont transmises via le circuit intégré ATmega8U2 utilisé en convertisseur USB-vers-série et la connexion USB vers l'ordinateur (mais pas pour les communications série sur les broches 0 et 1).

III.3.9. Principaux fonctionnements :

Les principales fonctionnalités de l'interface de l'application Arduino vous permet de créer et éditer un programme (appelé sketch) qui sera compilé puis téléversé sur la carte Arduino. Ainsi, lorsque vous apportez des changements sur le code, ces changements ne seront effectifs qu'une fois le programme téléversé sur la carte. Il est à noter que ce manuel fait référence à la version en anglais de ce logiciel puisqu'elle comporte habituellement des mises à jour plus récentes que la version en français.

III.4. Étude de la partie logicielle

Une carte d'acquisition que sa construction se base sur un microcontrôleur doit être dotée d'une interface de programmation comme est le cas de notre carte. L'environnement de programmation open-source pour Arduino peut être téléchargé gratuitement (pour Mac OS X, Windows, et Linux).

III.4.1. Plateforme de programmation Arduino :

Le logiciel de programmation (Figure II.3) de la carte ARDUINO sert d'éditeur de code (Langage proche du C). Une fois, le programme tapé ou modifié au clavier, il sera

Transféré et mémorisé dans la carte à travers de la liaison USB. Le câble USB alimente à la fois en énergie la carte et transporte aussi l'information ce programme appelé IDE ARDUINO [50]. (Voir Figure III.3)



Figure III.3.Logiciel de programmation de la carte ARDUINO

La fenêtre de l'application Arduino comporte les éléments suivants [51] :



Figure III.4.Arduino 1.8.1.

1. un menu.
2. une barre d'actions.
3. un ou plusieurs onglets correspondant aux sketches.
4. une fenêtre de programmation.
5. une console qui affiche les informations et erreurs de compilation et de téléversement du programme.

✓ **Éléments du menu :**

Les différents éléments du menu vous permettent de créer de nouveaux sketches, de les Sauvegarder, de gérer les préférences du logiciel et les paramètres de communication avec votre carte Arduino.

✓ **Dossier de travail :**

Dans les préférences (File > Préférences), il est possible de spécifier le dossier de travail. Il s'agit du dossier où seront sauvegardés par défaut les programmes et les bibliothèques qui pourront y être associées. Lorsqu'un programme est sauvegardé, un dossier portant le nom du programme est créé. Celui-ci contient le fichier du programme portant le nom que vous lui aurez donné suivi de l'extension .pde ainsi qu'un dossier intitulé applet qui contient les différents éléments créés et nécessaires lors du processus de compilation du programme et de téléversement vers la carte.

✓ **Outils de configuration (Tools) :**

Dans le menu Tools, il est possible et essentiel de spécifier le type de carte Arduino que vous utiliserez. Sous Tools > Board, il faut spécifier pour quel type de carte vous compilez et téléversez le programme. Le type de carte est généralement inscrit sur la carte elle-même. Il est également nécessaire lorsqu'on branche une nouvelle carte Arduino ou que l'on change de carte de spécifier le port sériel virtuel qui sera utilisé pour la communication et le téléversement du programme.

Pour ce faire, il faut aller sous Tools > Serial Port et choisir le port approprié. Sous Windows, il s'agit la plupart du temps du port ayant un numéro supérieur à 3. Sous Mac OS X, il s'agit habituellement du premier élément de la liste. Une bonne technique pour déterminer quel port correspond à la carte Arduino consiste à débrancher celui-ci, attendre un peu et de prendre note des ports déjà présents.

✓ **Barre d'actions :**

Le tableau (III.2) présente la barre d'actions de logiciel.






	Bouton « Verify » (Vérifier) ; il permet de compiler votre programme et de vérifier si des erreurs s'y trouvent. Cette procédure prend un certain temps d'exécution et lorsque est terminée, elle affiche un message de type « Binary sketch size : ... » indiquant la taille du sketch téléversé.
	Pour transmettre le sketch compilé avec succès sur la carte Arduino dans le microcontrôleur.
	Bouton « New » (Nouveau) ; ce bouton permet de créer un nouveau sketch.
	Bouton « Open » (Ouvrir) ; il fait apparaître un menu qui permet d'ouvrir un sketch qui figure dans votre dossier de travail ou des exemples de sketches intégrés au logiciel.
	Bouton « Save » (Sauvegarder) ; il permet de sauvegarder votre sketch.

Tableau III.2.Barre d'actions

III.4.2. Proteus Professional :

C'est une suite logicielle destinée à l'électronique. Développé par la société Labcenter Electronics, les logiciels incluent dans Proteus Professional permettent la CAO (Construction Assistée par Ordinateur) dans le domaine électronique. Deux logiciels principaux composent cette suite logicielle: (ISIS, ARES, PROSPICE) et VSM. 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 Professional possède d'autres avantages

- 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.

III.5. Les servomoteurs :

Les servomoteurs sont des actionneurs. Très utilisés en modélisme et dans l'industrie, ils ont comme caractéristique principale leur « couple », c'est-à-dire la force de rotation qu'ils peuvent exercer. Plus un servomoteur aura de couple et plus il pourra actionner des « membres » lourds comme déplacer un bras qui porte une charge (Figure III.5)[52].



Figure III.5. Les servomoteurs [52].

Un servomoteur est comme son nom l'indique un moteur mais avec quelques spécificités en plus. Contrairement à un moteur classique qui est utilisé pour tourner avec une vitesse proportionnelle à un courant ou à une tension, pour obtenir une position. Il effectue une rotation suivant un angle déterminé en fonction du signal envoyé. Le plus souvent l'angle est compris entre 0 et 180° puis il garde cette position.

III.5.1. Partie mécanique de servomoteur :

Un servomoteur comprend (Figure III.6):

- Un moteur électrique (continu), généralement assez petit.
- Des engrenages réducteur en sortie du ce moteur (pour avoir moins de vitesse et plus de couple ou de force).
- Un capteur type "potentiomètre" raccordé sur la sortie.
- Il s'agit donc d'une résistance qui varie en fonction de l'angle, ce qui permet de mesurer l'angle de rotation sur l'axe de sortie.
- Un asservissement électronique pour contrôler la position/rotation, de cet axe de sortie pour le maintenir à la bonne position.
- Un dispositif électrique d'asservissement.
- Un axe dépassant hors du boîtier avec différents bras ou roues fixation.

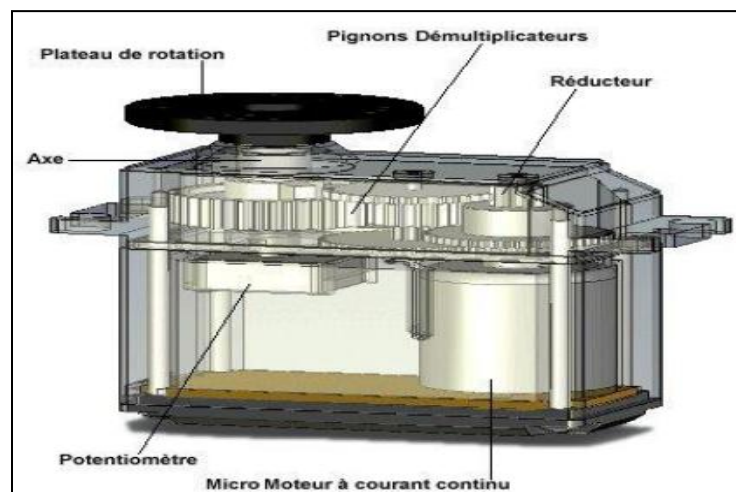


Figure III.6. Architecture d'un servomoteur [53].

III.5.2. Les caractéristiques du SG90 sont les suivantes :

Les différentes caractéristiques du servomoteur sont présentées dans la figure ci-dessous (Figure III.7) [54].

Servomoteur à rotation angulaire (Micro-Servo)

Caractéristiques : <ul style="list-style-type: none">• Alimentation : 4,8 à 6V• Angle de rotation : 180°• Couple : 1,3 kg.cm• Vitesse : 0,12 sec/ 60°• Dimensions : 23,2 X 12,5 X 22 mm• Poids : 9g

Figure III.7. Caractéristiques du servomoteur à rotation angulaire (Micro-Servo) [54].

III.5.3. Connecteur du servomoteur :

Un servomoteur se pilote par l'intermédiaire d'un câble à (03) trois fils (Figure III.8). Ce câble permet à la fois de l'alimenter et de lui transmettre des consignes de position par le fil de signal :

- Le noir ou marron : La masse
- Le rouge : La tension d'alimentation continue (+)
- Le jaune, orange, blanc ou bleu : Le signal de commande PWM

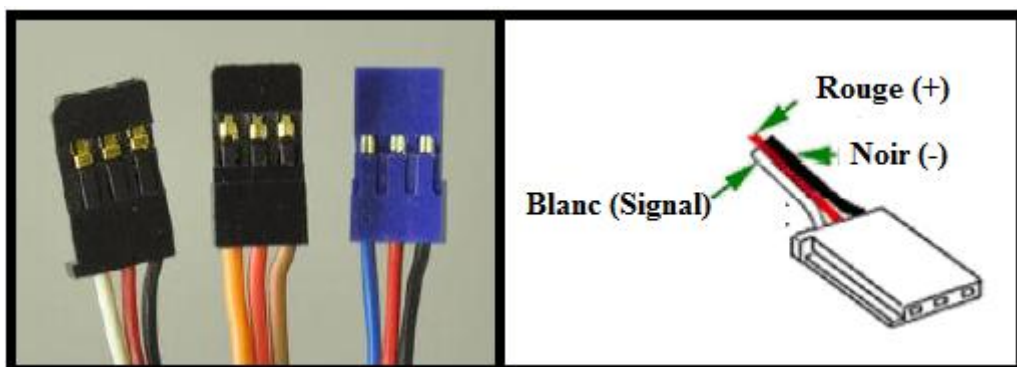


Figure III.8. Câble de commande pour servomoteur

III.5.4. Fonctionnement :

Le servomoteur est commandé par l'intermédiaire d'un câble électrique à 3 fils qui permettent d'alimenter le moteur et de lui transmettre des ordres de positions sous forme d'un signal codé en largeur d'impulsion plus communément appelés **PWM** ou **RCO**. Cela signifie que c'est la durée des impulsions qui détermine l'angle absolu de l'axe de sortie et donc la position du bras de commande du servomoteur (Figure III.9). Le signal est répété périodiquement, en général toutes les 20 millisecondes, ce qui permet de contrôler et de corriger continuellement la position angulaire de l'axe de sortie, cette dernière étant mesurée par le potentiomètre (Figure III.10) [54].

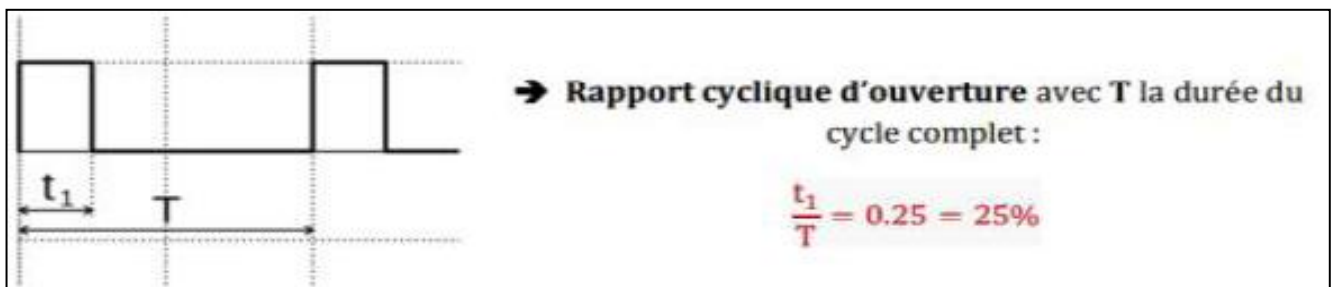


Figure III.9. Principe d'un signal de commande par PWM

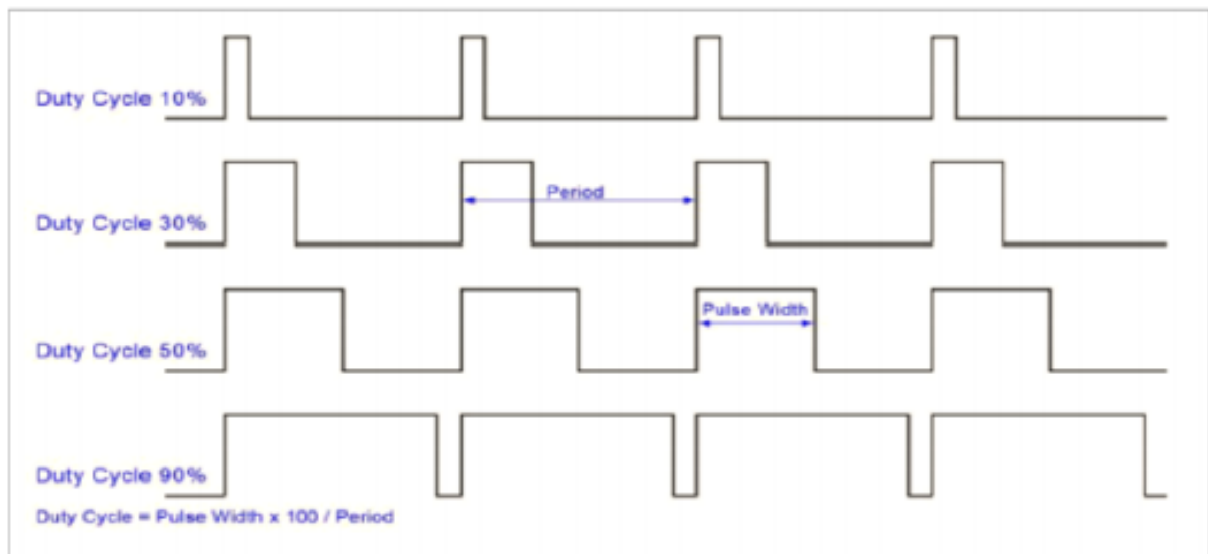


Figure III.10. Exemples des signaux PWM

III.5.5. Piloter un servomoteur avec l'Arduino :

Le mode de commande d'un servomoteur est standardisé : on envoie sur son fil de commande une impulsion dont la durée correspond à l'angle désiré (Figure III.11). Historiquement, cette impulsion était délivrée par un circuit oscillateur.

Avec la programmation de l'Arduino, ce circuit n'est plus nécessaire. Une bibliothèque (library) dédiée, la bibliothèque « servo », permet de piloter un servomoteur en lui transmettant simplement l'angle sur lequel il souhaite se positionner.

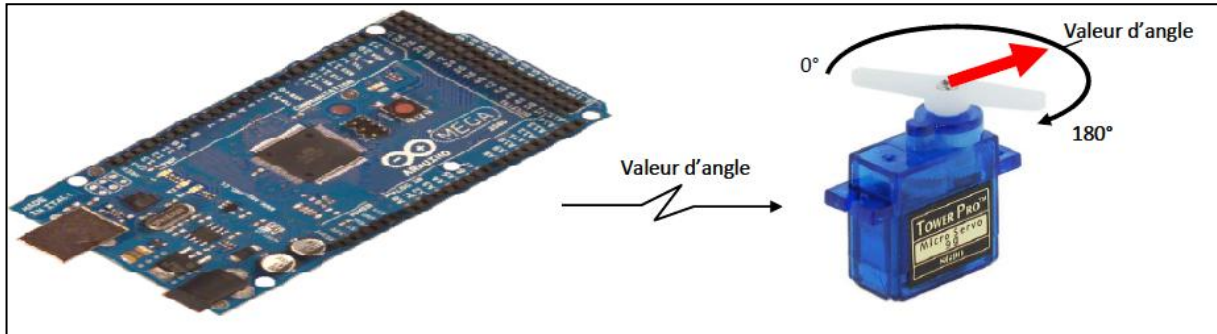


Figure III.11. La commande à l'aide Arduino [52].

Ce qui permet de réaliser simplement des mouvements qui peuvent être complexes et de les automatiser.

Le tableau suivant présente les connaissances nécessaires au pilotage d'un servomoteur :

<code>#include <Servo.h></code>	L'utilisation de la librairie <code>servo.h</code> qui prend en charge la communication de bas niveau entre l'Arduino et le servomoteur. Les commandes précédées par <code>#</code> sont des ordres particuliers donnés au compilateur
<code>Servo myservo;</code>	La création d'un objet de type <code>Servo</code> , appelé ici <code>myservo</code>
<code>myservo.attach(9);</code>	Le fil de commande de ce servo sera connecté au PIN 9 et l'objet <code>myservo</code> pilotera ce PIN
<code>myservo.write(90);</code> <code>myservo.write(Angle);</code>	Demander au servomoteur de se déplacer à l'angle désiré, soit de façon absolue en lui indiquant une valeur entière (90°) dans le premier cas, soit en lui passant le contenu d'une variable (<code>Angle</code>) compris entre 0 et 180, ce qui peut être utile par exemple pour donner une progressivité au déplacement en faisant varier l'angle d'un pas fixe (quelques degrés) par une boucle
<code>myservo.read(Angle);</code>	Pour lire la valeur de l'angle du servomoteur

Tableau III.3. la programmation de servomoteur sur l'Arduino [52].

III.6. Le module PCA9685:

Le PCA9685 remplaçant du 9635 est un prévu à l'origine pour commander en mode modulation de largeur d'impulsion (PWM) des retro-éclairages et des panneaux de Leds

RGBA avec 4 canaux par couleurs soit 16 canaux individuels. Ce circuit peut bien sur être utilisé avec n'importe quel autre type d'application utilisant des commandes PWM comme des servomoteurs par exemple [55].

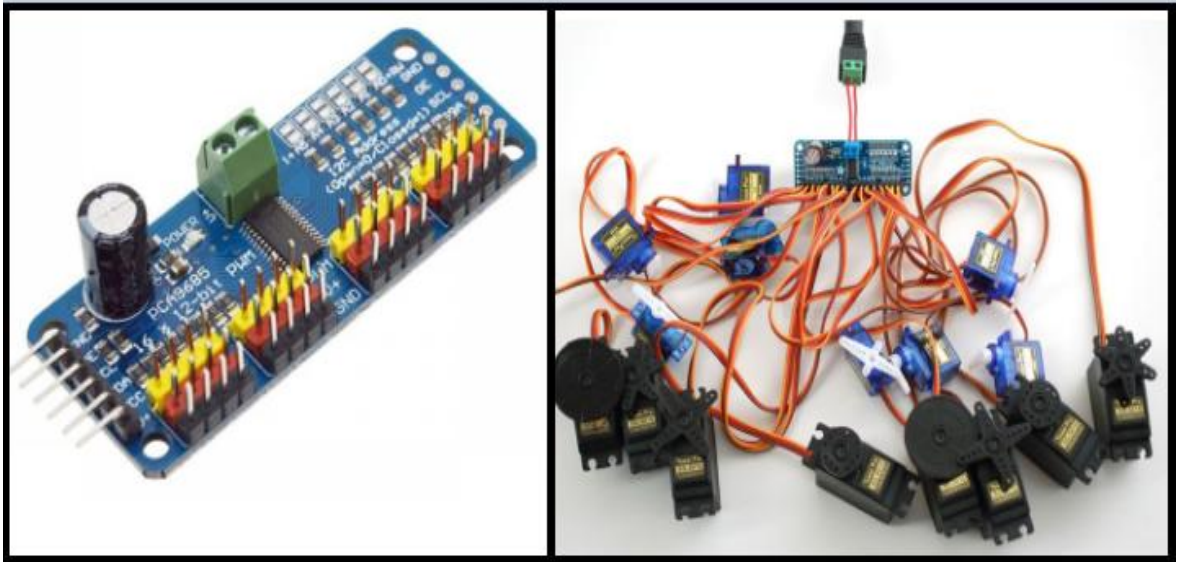


Figure III.12. Le PCA9685.

III.6.1. Caractéristiques :

- C'est un pilote PWM contrôlé par i2c avec une horloge intégrée.
- Il est conforme à 5V, ce qui signifie que vous pouvez le contrôler à partir d'un microcontrôleur 3.3V et toujours en toute sécurité jusqu'à 6V.
- 6 broches de sélection d'adresse afin que vous puissiez transférer jusqu'à 62 d'entre elles sur un seul bus i2c, un total de 992 sorties - c'est beaucoup de servomoteurs ou de LED.
- PWM à fréquence réglable jusqu'à environ 1,6 KHz.
- Résolution 12 bits pour chaque sortie - pour les servomoteurs, cela signifie une résolution de 4 μ s au taux de mise à jour de 60 Hz.
- Sortie configurable push-pull ou open-drain.
- La broche de validation de sortie permet de désactiver rapidement toutes les sorties.
- Bloc de jonction pour l'entrée d'alimentation.
- Protection contre les inversions de polarité sur l'entrée du bloc terminal.
- Connecteurs à 3 broches en groupes de 4 afin de pouvoir brancher 16 servomoteurs à la fois.
- Conception «en chaîne»
- Un grand condensateur sur la ligne V +

- Résistances série de 220 ohm sur toutes les lignes de sortie pour les protéger et pour rendre les LED de conduite triviales.

III.6.2. Détail sur PCA9685:

Ce module pilote jusqu'à 16 servomoteurs sur I2C avec seulement 2 broches.

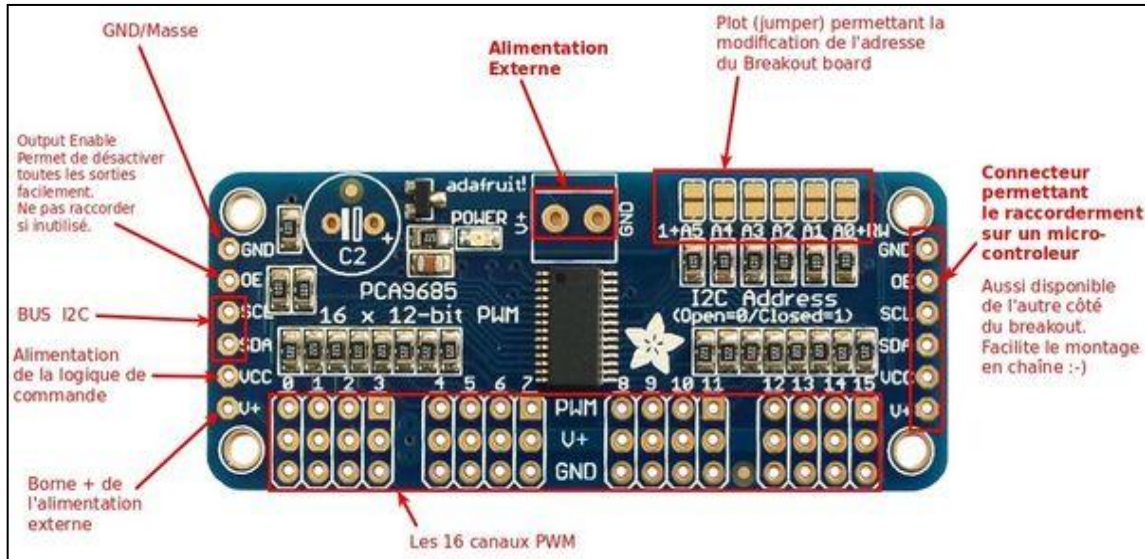


Figure III.13. Brochage de PCA9685.

III.6.3. Branchement sur un Arduino :

Le PCA9685 se raccorde sur un Arduino Mega2650 comme suit:

Breakout	Arduino
GND	GND
VCC	5V
SCL	Numérique 21
SDA	Numérique 20
OE	(Output Enabled) Non utilisé
V+	(le '+' du bornier d'alimentation externe) Non utilisé

Tableau III.4.Raccordement sur Arduino.

III.6.4. Alimentation externe :

Le bornier bleu sur le module PCA9685. Ce dernier permet de raccorder une source d'alimentation externe utilisé pour fournir la puissance nécessaire à 16 servomoteurs. Si un servomoteur ne réclame pas énormément d'énergie, il n'est absolument pas de même lorsqu'il faut en commander 16. Le régulateur de tension pauvre Arduino ne tiendrait

probablement pas le coup bien longtemps. Il faut donc utiliser une source d'énergie externe... la raison de ce connecteur. Selon la fiche produite, vous pouvez y raccorder une source de tension allant jusqu'à 6 Volts (ce qui est le maximum en usage pour les Servomoteur) [56].

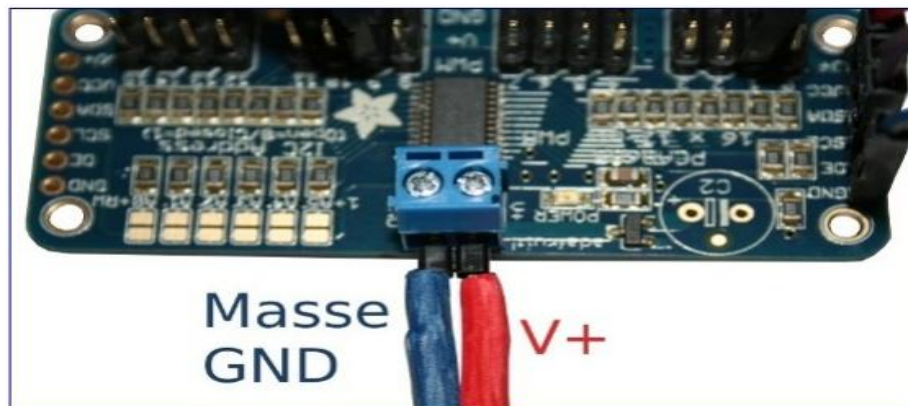


Figure III.14. Alimentation de PCA9685.

III.6.5. Branchement d'un servomoteur sur PCA9685 :

Les pinHeader (connecteurs) ne disposent pas de détrompeurs, il est donc possible de raccorder les servomoteurs dans le mauvais sens par erreur.

Inverser le sens de raccordement du servo reviendrait à appliquer +5v sur la broche de commande du PCA9685. [56]

PCA9685 à cependant penser à sécurisé les sorties PWM, en effet une résistance de 220 Ohms est montée en série sur chaque sortie.

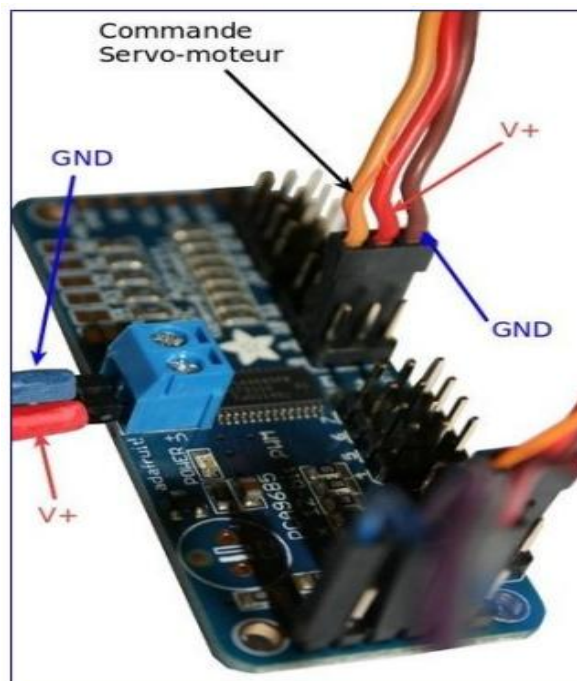


Figure III.15. Branchement d'un servomoteur sur PCA9685

III.7. Capteur d'obstacle infrarouge à 4 canaux :

Un capteur infrarouge permet de détecter un mouvement dans son champ de vision en se basant sur l'infrarouge, il hautement sensibles, sélectifs et stable. Leur durée de vie est relativement longue et il est insensible aux modifications de l'environnement. Il est t constitués d'un ensemble émetteur/récepteur fonctionnant avec des radiations non visibles, dont la longueur d'onde est juste inférieure à celle du rouge visible.

Le capteur d'obstacle infrarouge à 4 canaux est adaptable à la lumière ambiante. Il a une paire de tube de transmission et de réception infrarouge.

La figure (III. 16.) Représente les dimensions de module émetteur et de récepteur, d'un Capteur d'obstacle infrarouge.

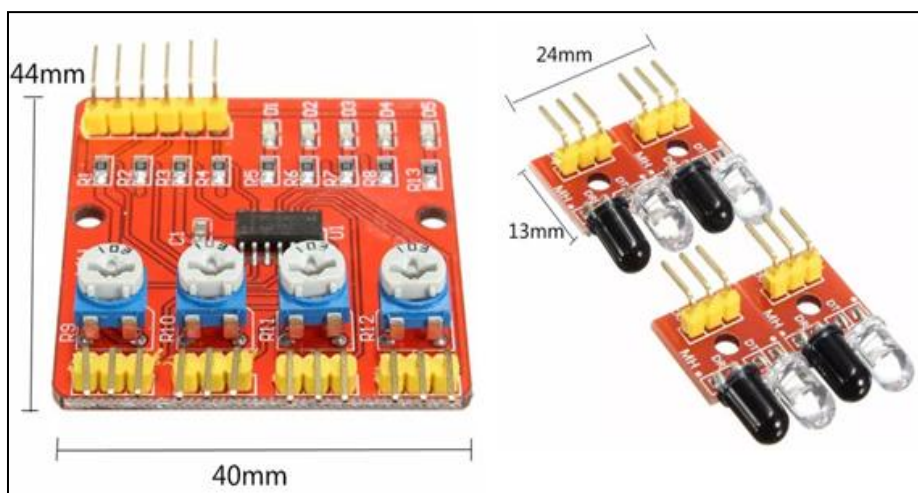


Figure III.16. Dimensions de module émetteur et de récepteur

La figure (III.17) Représente la composition et le branchement d'un Capteur d'obstacle infrarouge à 4 canaux avec l'Arduino

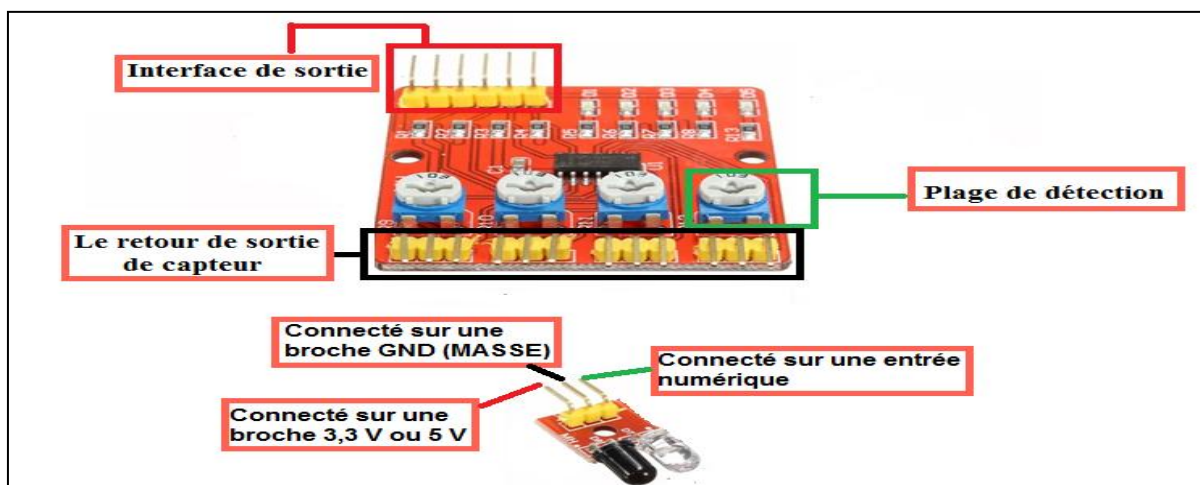


Figure III.17. Composition et branchement de Capteur d'obstacle infrarouge à 4 canaux

III.7.1. Fonctionnement :

Lorsque le module détecte un obstacle, Le tube émetteur émet une certaine fréquence IR qui sera réfléchiée et reçue par le tube récepteur. Après le traitement avec le circuit de comparateur, l'interface de sortie de signal sort un signal numérique (un signal de niveau de basse tension).

La distance de détection de ce module est de 2 à 30 cm, l'angle de détection est de 35°. Cette distance de détection est réglable peut être contrôlée par le 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.

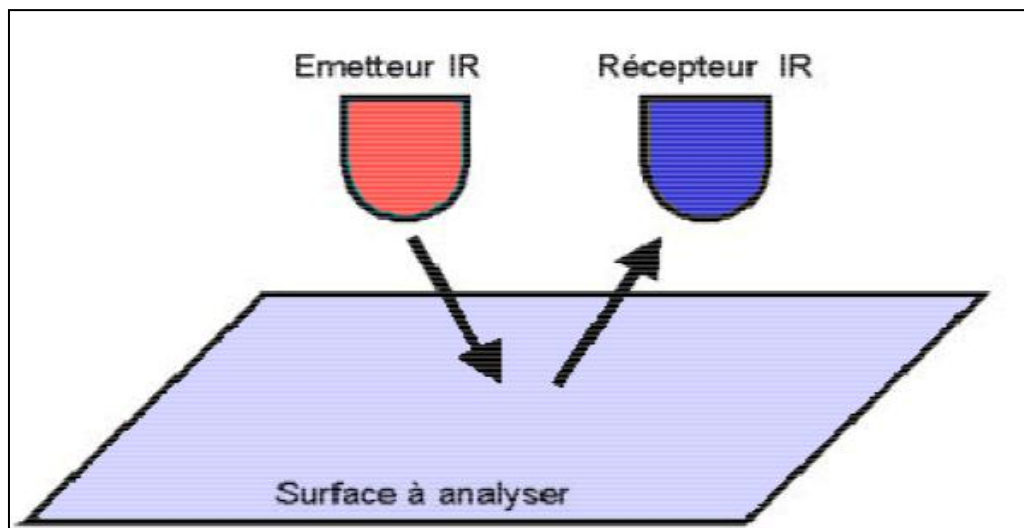


Figure.III.18. Principe de fonctionnement d'un capteur infrarouge

III.8. Moteurs à courant continu :

Les machines à courant continu sont des convertisseurs électromécaniques d'énergie : Soit ils convertissent l'énergie électrique absorbée en énergie mécanique lorsqu'ils sont capables de fournir une puissance mécanique suffisante pour démarrer puis entraîner une charge en mouvement. On dit alors qu'ils ont un fonctionnement en moteur. Soit ils convertissent l'énergie mécanique reçue en énergie électrique lorsqu'ils subissent l'action d'une charge entraînée. On dit alors qu'ils ont un fonctionnement en générateur [57].

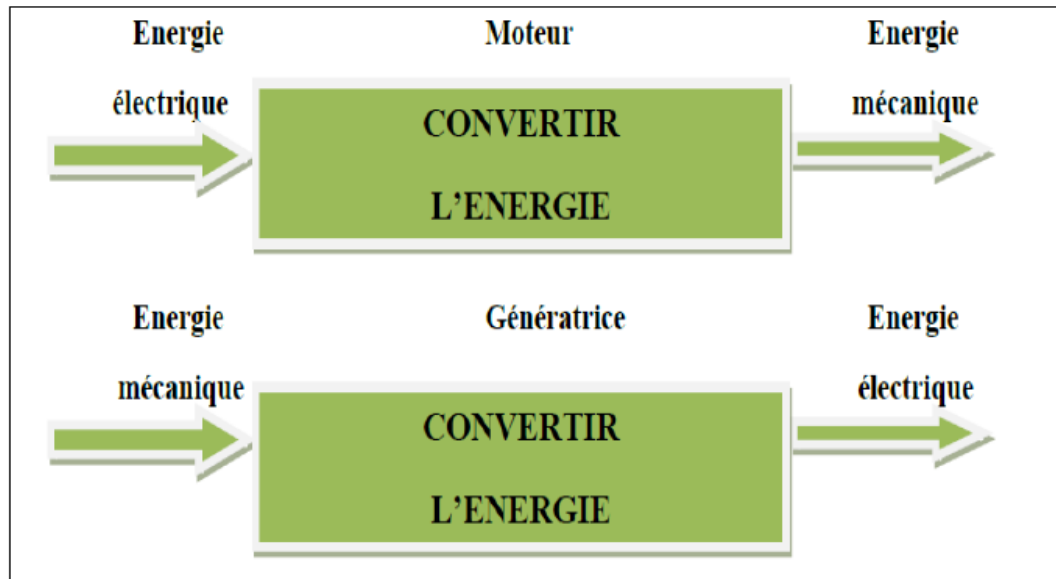


Figure.III.19. Fonctionnement de machine à courant continu.

III.8.1. Constitution :

La machine à courant continu est constituée de trois parties principales :

- L'inducteur : L'inducteur est la partie fixe du moteur. Il est constitué d'un aimant permanent ou d'un électroaimant alimenté par le courant continu d'excitation (I_e).
- Le rotor (l'induit): Le rotor est constitué d'encoches dans lesquelles est enroulé un bobinage de (N) conducteurs alimentés en courant continu (I) via le collecteur
- Le collecteur et les balais : Le collecteur est un ensemble de lames de cuivre où sont reliées les extrémités du bobinage de l'induit [58]. Les balais (ou charbons) sont situés au stator et frottent sur le collecteur en rotation.

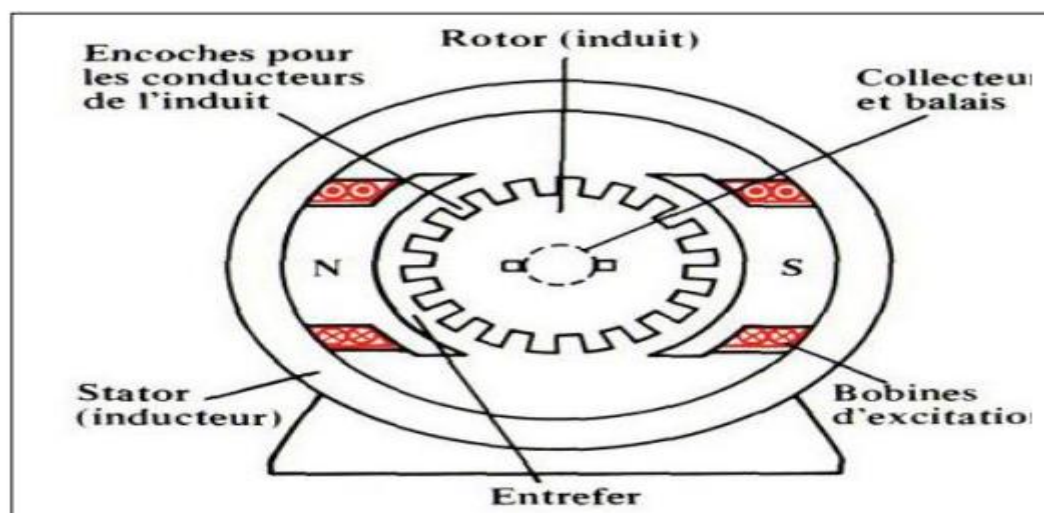


Figure III.20. Description de la machine à courant continu

III.8.2. Principe de fonctionnement :

Le fonctionnement du moteur à courant continu est basé sur le principe des forces de Laplace:

Un conducteur de longueur (L), placé dans un champ magnétique et parcouru par un courant, est soumis à une force électromagnétique.

Le champ créé par l'inducteur agit sur les conducteurs de l'induit : Chacun des (N) conducteurs de longueurs (L) placé dans le champ (B) et parcouru par un courant (I) est le siège d'une force électromagnétique perpendiculaire au conducteur :

$$F=B.I.L. \text{ (I-1)}$$

Ces forces de Laplace exercent un couple proportionnel à l'intensité (I) et au flux (Φ) sur le rotor. Le moteur met à tourner à une vitesse proportionnelle à la tension d'alimentation (V) et, inversement proportionnelle au flux (Φ).

Au passage de tout conducteur de l'induit sur la ligne neutre, le courant qui le traverse change de sens grâce au collecteur. Le moteur conserve le même sens de rotation.

Pour inverser le sens de rotation du moteur, il convient d'inverser le sens du champ produit par l'inducteur par rapport au sens du courant circulant dans l'induit :

- ✓ Soit on inverse la polarité de la tension d'alimentation de l'induit.
- ✓ Soit on inverse la polarité d'alimentation du circuit d'excitation [57].

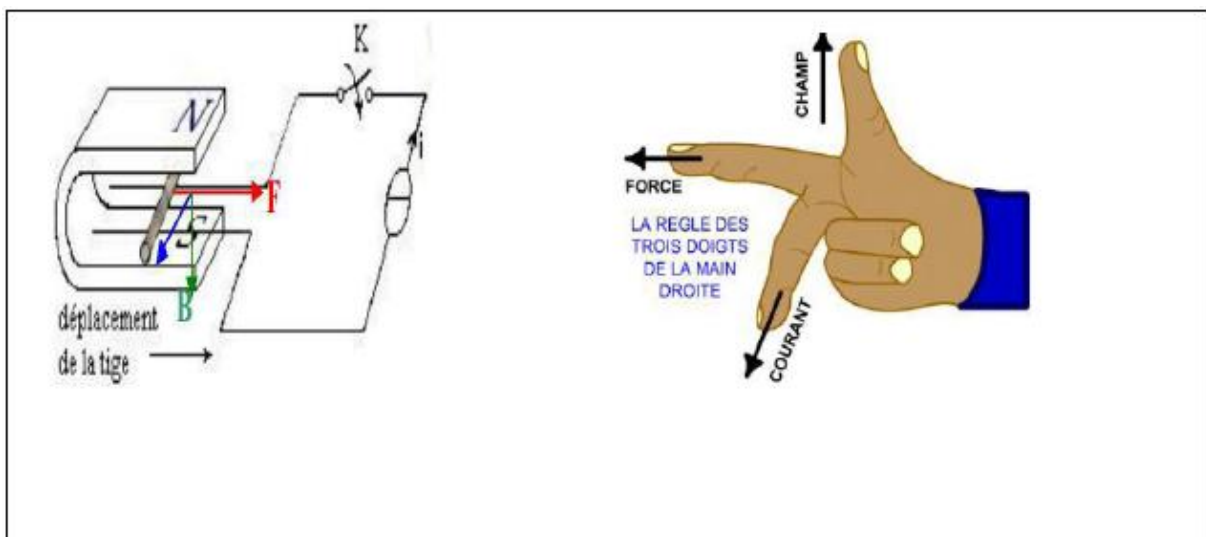


Figure.III.21. Principe de fonctionnement d'un moteur à courant continu.

III.8.3. Avantage des moteurs à courant continu :

Les moteurs à courant continu asservis apportent de grands avantages sur les moteurs pas à pas :

Meilleure précision de positionnement.

- Plus de puissance ;
- Mouvements plus rapides ;
- Meilleure rendement ;
- Moins d'échauffement [59].

III.9. Relais :

Le relais est un composant à 4 broches minimum .c'est un interrupteur que l'on peut commander en envoyant un petit courant. Au repos, il est normalement fermé, ou normalement ouvert, selon le modèle. On peut s'en servir avec l'Arduino pour commander des machines en haute tension, ou pour déclencher toute machine ou lumière [60].

III.9.1. Type des relais:

Lorsqu'on choisit ou qu'on récupère un relais, il peut être intéressant de savoir de quel type de connexion il s'agit. Il existe 4 grandes familles de relais en fonction des contacts, et nous avons choisi le relais SPDT pour notre projet.

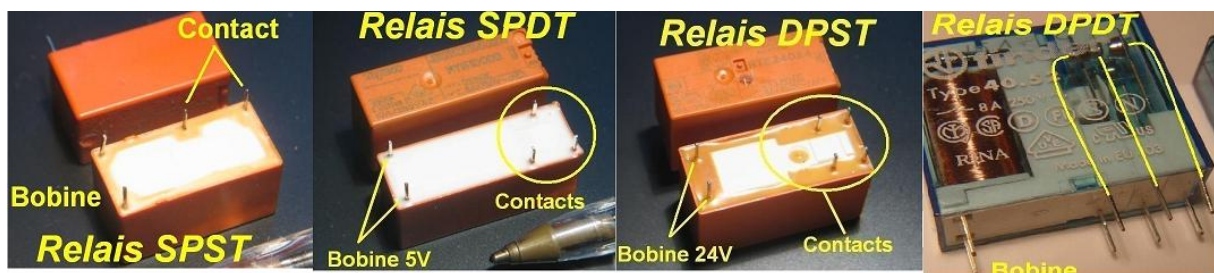


Figure III.22. Les types des relais.

III.9.2. Fonctionnement de relais :

Le relais SPDT possède un seul contact mais avec une borne commune, un contact normalement ouvert (quand il n'y a pas de tension sur la bobine) et un contact normalement fermé (quand il n'y a pas de tension sur la bobine). Quand on applique une tension sur la bobine, on entend "clic" : la borne commune va se connecter sur le contact normalement ouvert (NO) et le contact normalement fermé (NC) s'ouvre. Dès qu'on coupe la tension aux bornes de la bobine, on entend "clic" et le relais revient à son état de repos.

Le relais SPDT possède 5 broches au total : 3 pour les contacts, 2 pour la bobine.



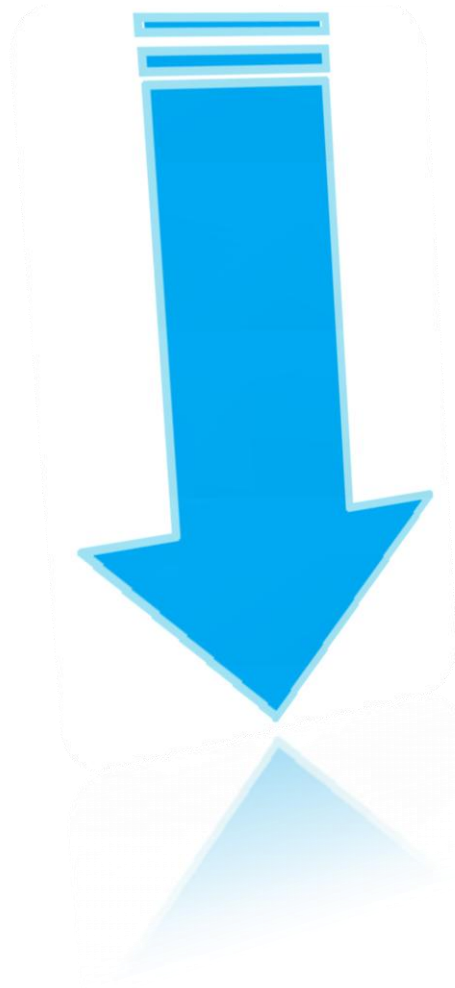
Figure III.23. Le relais SPDT

III.10. Conclusion :

Dans ce chapitre, nous avons vu la carte Arduino, puis nous avons cité les différents types de cette dernière. Ensuite, nous avons expliqué les deux parties essentielles de l'Arduino; plus précisément (la partie matérielle et la partie de programmation). Nous avons également expliqué le principe de fonctionnement de la carte Arduino sans oublier ses caractéristiques.

Ainsi quelques descriptions théoriques sur le servomoteur, PCA9685, Capteur d'obstacle infrarouge à 4 canaux et moteur DC et relais. Le chapitre suivant sera consacré à la réalisation du système distributeur des médicaments à base d'Arduino Méga2560.

Chapitre IV



Conception et réalisation

IV.1. Introduction :

Dans ce chapitre, on va procéder à l'application expérimentale, « la conception et la réalisation d'un système automatisé de distribution du médicament ». Pour cette raison, on présentera les différents éléments constitutifs afin de réaliser une telle combinaison avec le module Arduino, les Outils de programmation (PROTEUS, Arduino) ainsi que les différents tests et le montage final.

IV.2. Les étapes de la réalisation :

La réalisation de notre dispositif a connu trois étapes :

- **Première étape** : C'est l'étape la plus difficile car elle comporte la conception de ce système, le choix des composants convenables à utiliser pour aboutir à l'objectif, et le montage.
- **Deuxième étape** : Simulation du système dans l'environnement PROTEUS avec programmation d'Arduino.
- **Troisième étape** : test du montage complet de tous les composants, puis la réalisation sur la maquette.

IV.3. Description du système :

Notre système assure la distribution de 16 types de médicaments (16 casiers), de manière automatique par ordinateur, qui consiste à envoyer les données via le port série à l'Arduino sous forme d'une trame contenant les informations nécessaires, comme le nom du casier et le nombre de boîte à distribuer, A partir de ces données l'Arduino commande les servomoteurs, à travers une interface le convoyeur est aussi mis en marche afin de finaliser l'opération de la distribution.

IV.3.1. Dispositifs matériels et technologie :

Pour atteindre notre but, il est impérativement de le construire en utilisant différents équipements et matériel, Les équipements utilisés dans le montage de la maquette sont les suivants :

- Maquette (flow rack) de 16 cases et maquette de convoyeur.
- Circuit électronique (PCA968 Servo Driver + Arduino méga).
- Alimentation (5V, 12V).
- 16 servomoteurs (Sg90)
- Câbles (22 AWG – 4p).
- Convoyeur (moteur DC, relais 5V 1canal).
- Capteur d'obstacle infrarouge à 4 canaux

Technologies :

- IDE Arduino
- PROTEUS

IV.3.2. Maquette de système :**a. Description :**

La maquette est fabriquée en Forex, avec une configuration géométrique rectangulaire, d'une hauteur d'environ 77 cm, une profondeur de 50 cm et une largeur de 48 cm, elle est de type Flow rack (16 casiers inclinés), Chaque casier regroupant un ensemble d'emplacements pouvant contenir plusieurs produits identiques placés les uns à la suite des autres. Le système contient quatre casiers par lignes avec des longueurs différentes selon le type de médicament choisi :

- **Casier 1** : longueur de 10 cm
- **Casier 2** : longueur de 12 cm
- **Casier 3** : longueur de 13 cm
- **Casier 4** : longueur de 12 cm

Pour l'angle d'inclinaison des casiers on a choisi 30 degrés permettant un glissement rapide de médicament pour que le système soit plus efficace.

La figure (IV.1) représente la maquette en 3D qu'on a réalisé avec le logiciel AutoCAD.

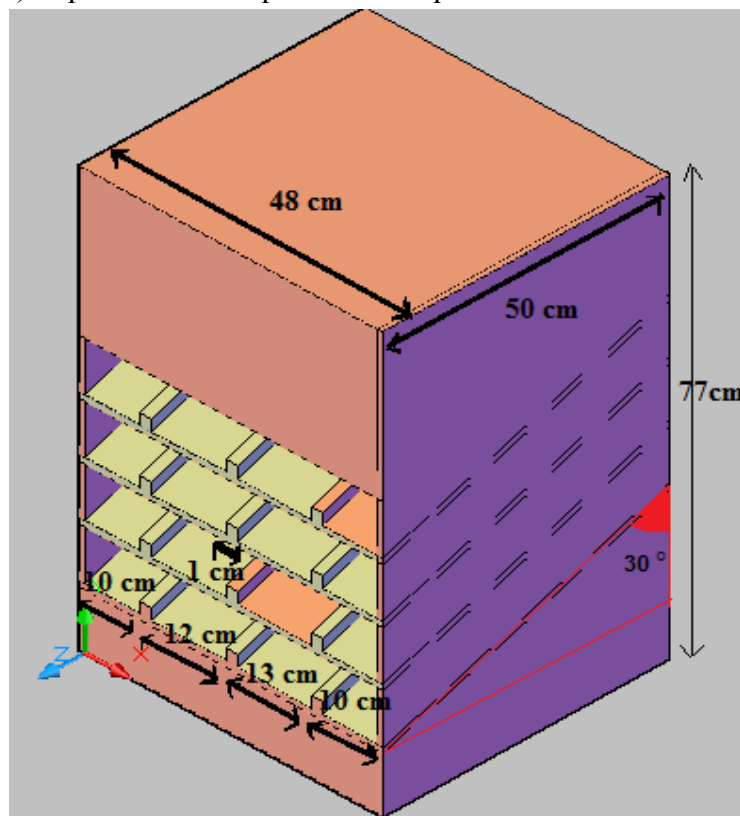


Figure IV.1.Maquette en 3D

Les figure (IV.3) et (IV.4) représente des photos réelles de la maquette fabriquer en Forex.



Figure IV.2. Vue de face de la maquette.



Figure IV.3.vue arrière de la maquette.

b. Montage :

Le système est basé sur les servomoteurs qui provoquent la descente en chute libre du produit, la phase de montage sur la maquette consiste à percer l'extrémité de chaque case (en forme d'un rectangle de 3.5 cm de longueur et 3 cm de largeur) pour placer un servomoteur comme le montre la figure (IV.5)

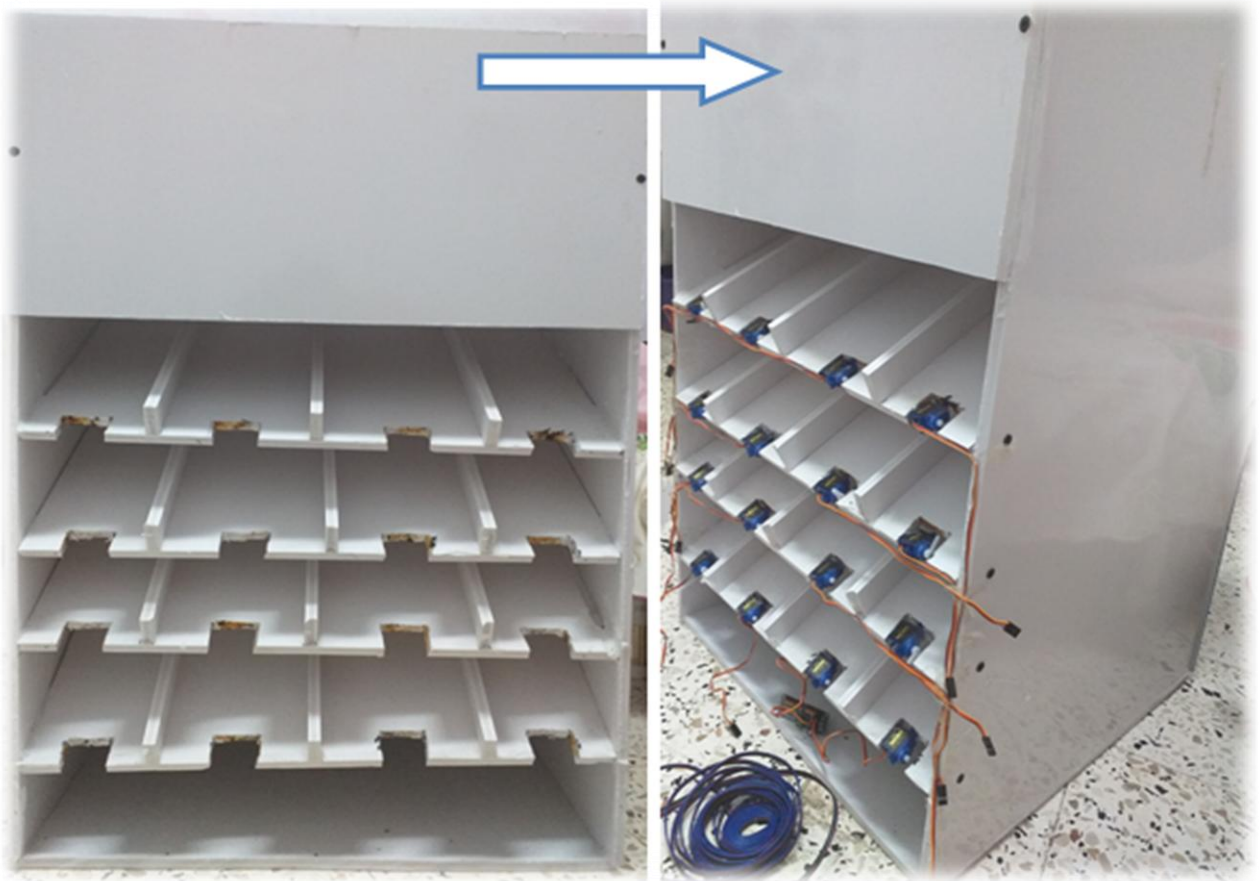


Figure IV.4. Phase du montage de maquette

IV.3.3. Convoyeur :

Un système de convoyage est quasiment tout le temps nécessaire afin d'acheminer les médicaments près des comptoirs, comme première étape de réalisation d'un convoyeur de type tapis roulant on a fait la conception avec 'SOLIDWORKS', la figure (IV.6), représente le modèle 3D de convoyeur, et la figure (IV.7) représente ses composants.

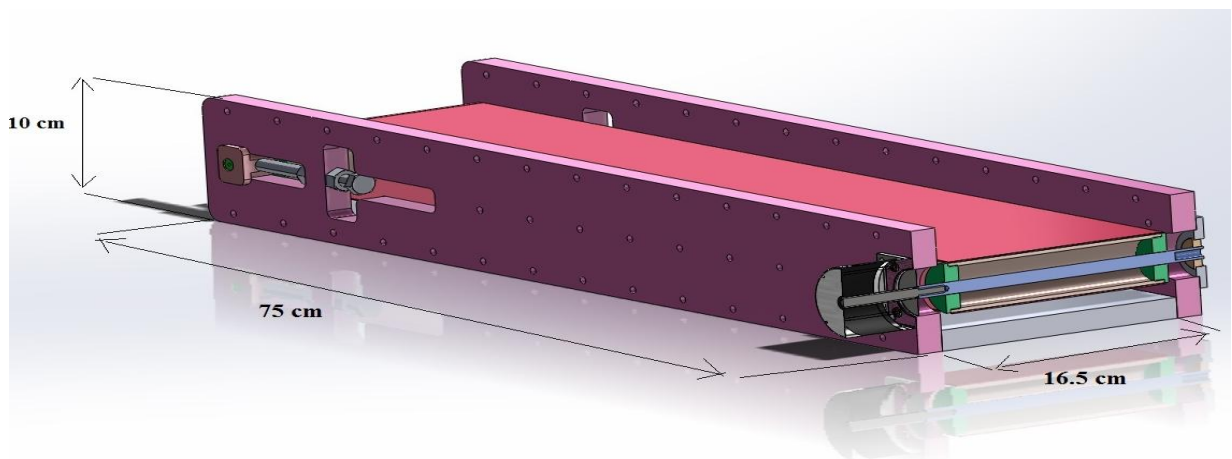


Figure IV.5. Convoyeur en 3D

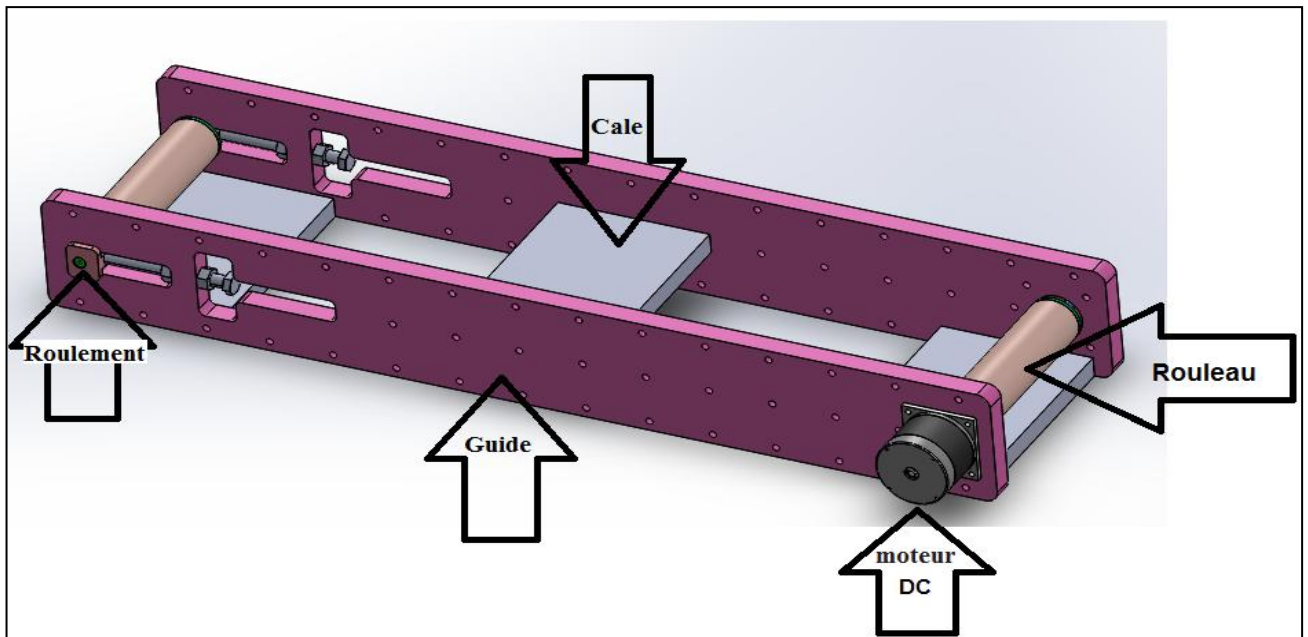


Figure IV.6.composants de convoyeur

La figure (IV.7) représente une photo réelle de convoyeur fabriqué en PVC avant et après le montage.



Figure IV.7.Photo réelle de convoyeur fabriqué en PVC avant et après le montage

IV.4. Choix des composants et justification :

a. Arduino méga

La carte Arduino Méga (2560) propose plus d'entrées /sorties. Le microcontrôleur utilisé est de la même famille que les cartes précédentes, mais il dispose de plus de ports, tiers, liens séries, mémoire. Cette carte permet de réaliser un décodeur D18 à 66 sorties.



Figure IV.8.photo réelle de la carte Arduino mega2560

▪ Caractéristiques :

Flash: 256KB

RAM: 8KB

Quartz: 16MHz

E/S : 54 digitales + 16analogiques

UART : 1 connectée à l'USB + 3

PWM : 12 (5 timers)

Contrôler les servomoteurs depuis un Arduino directement est assez simple, Cependant, il a ses limites:

- La librairie de serveurs Arduino peut entrer en conflit avec d'autres bibliothèques Arduino lorsqu'elles tentent d'utiliser le même minuteur. Cela peut parfois être résolu en recherchant des bibliothèques alternatives.
- On est limité par le nombre de broches PWM sur l' Arduino. Si le servo fait partie d'une conception nécessitant d'autres périphériques PWM, cela peut poser un problème.

b. La carte PCA9685 :

Le module PCA9685 c'est une meilleure solution qui consiste à utiliser une carte de pilote de servo distincte. Cela déchargera la tâche d'envoyer des messages PWM aux servos, ce qui permettra à notre Arduino de faire de meilleures choses d'une part .d'autre part on évite les risques électriques qui peuvent détruire la carte Arduino.

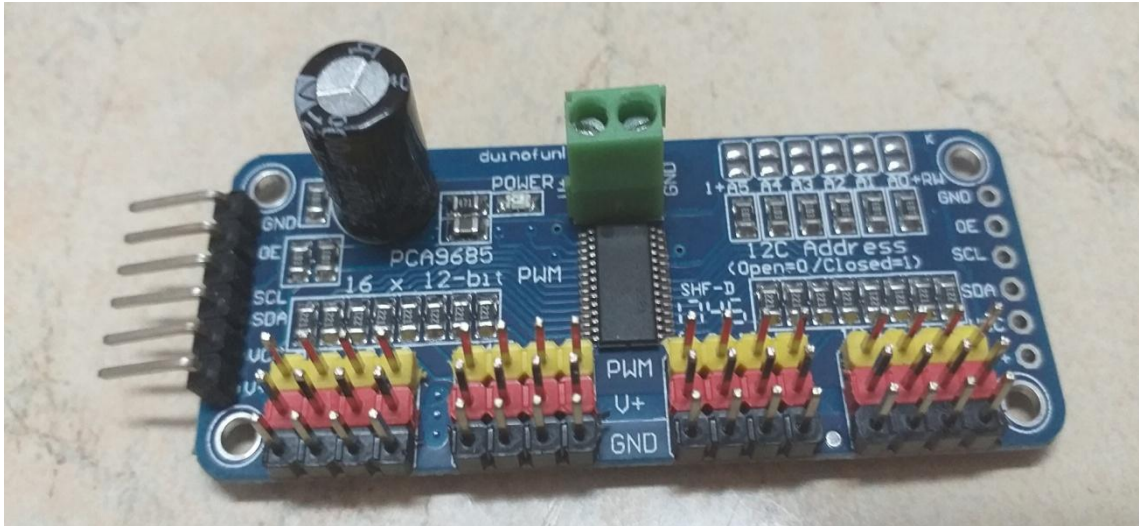


Figure IV.9,photo réelle du module PCA9685

✓ **Caractéristiques :**

- PCB Taille : 62,5 mm x 25,4 mm
- Puce : PCA9685
- Interface : I2 C
- Canaux : 16
- Tension d'entrée : 3,3-5,5 V

Le PCA9685 est alimenté par une source de tension externe, pour cela on a utilisé une alimentation 'AC 100-240v 50/60 Hz' de 5v/2 A .



Figure IV.10,photo réelle d'alimentation AC 100-240v 50/60hz

▪ Caractéristiques d'AC 100-240v 50/60hz :

- Entrée: AC 100-240V, 50-60 Hz, 2A MAX
- Sortie: 5DCV, 2000mA
- Longueur de câble: Environ 150 CM
- Connecteur: 5,5mm (à l'extérieur)* 2,1mm (intérieur), (y compris les embouts avec la dimension: 2,5*0,7mm / 4,0*1,7mm / 3,5*1,35mm)
- 2 connecteurs: Extra avoir micro connecteurs
- Également utilisé: La barre lumineuse de LED, USB-HUB, système de téléphone à la maison, routeur de téléphone de VoIP, contrôleur de Serato DJ, PC de comprimé et ainsi de suite.

c. Servomoteurs (Sg90) :

La rotation d'un servomoteur suivant un angle déterminé provoque la chute libre de médicament, on a choisi ce composant pour deux raisons : la précision de rotation, il pourra actionner des « membres » lourd car il est puissant.



Figure IV.11.photo réelle d'un servomoteur sg90.

d. Capteur d'obstacle infrarouge à 4 canaux :

La commande à distance transmet un faisceau infrarouge, invisible c'est l'œil humain, sur une distance fixe de l'appareil. Pour savoir si un casier nécessite le remplissage on a placé des capteurs d'obstacle infrarouges qui détectent la présence de la boîte (le canal à remplir s'affiche sur l'écran de l'ordinateur de contrôle).

Les caractéristiques de ce capteur sont le suivant :

- Tension de travail: DC 3.3v-5v
 - courant de travail : 1 A.
 - température de fonctionnement :-10 ° c à + 50°C
 - plage de détection : 1 mm 60 cm réglable, plus le rendement plus stable, blanc ne reflète le plus loin possible.
- interface de sortie : interface 6 fils (1234 à 4 extrémités de sortie de signal, + puissance positive, - pour la puissance négative est au sol)

e. Moteurs à courant continu :

Parce qu'on a décidé de réaliser un convoyeur, alors on a utilisé un moteur à courant Continu pour commander la marche et l'arrêt du tapis roulant.

On a utilisé un moteur d'essuie-glace, car il est entraîné par un moto-réducteur et ça c'est parfait pour un simple convoyeur, en plus il est facile à utiliser et programmable depuis une très large gamme de langages de programmation. La figure (IV.12.) Représente la photo réelle du moteur .

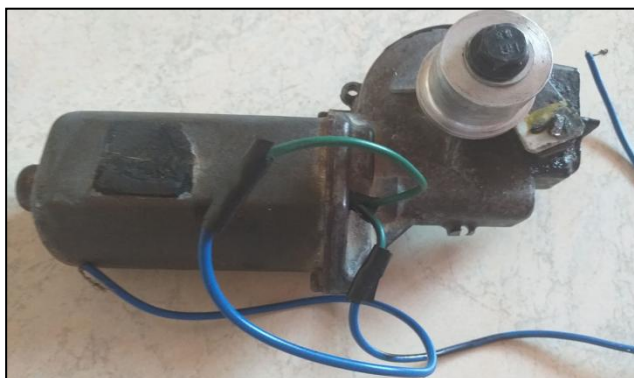


Figure IV.12. moteur DC.

Évidemment, pour pouvoir valider un moteur, il faut connaître les spécifications que nous voulons atteindre.

Caractéristiques du moteur :

- Extrêmement silencieux.
- Muni de 8 angles de balayage, réglable à 40, 50, 60, 70, 80, 90, 100 ou 110° (Réglage standard est 110°)
- Livrabale pour installation électrique de 12 volts.
- Puissance 30 watts
- Retour à l'arrêt automatique.
- 2 vitesses
- Arbre en acier inoxydable.
- Convient pour des épaisseurs de cloison entre 3 et 13 mm (Version 25 mm de longueur) ou

entre 3 et 38 mm (version 51 mm de longueur d'arbre).

-Bout de l'arbre droit et dentelé (72 dents) ou conique et dentelé selon DIN 72783, pour la réception du bras de balai.

-Tous les moteurs d'essuie-glace respectent la norme de compatibilité électro magnétique.

f. Module relais 5v 1 canal :

On a utilisé ce module pour commander le moteur qui a une tension 12 v bien supérieur à ce que l'Arduino pourrait accepter (5v).

Caractéristiques:

- Alimentation : 5V
- Pouvoir de coupure : 250 VAC 10A/ 30 VDC 10A
- Nombre de canaux : 1 NO/NC

La figure (IV.13) présente une photo réelle de Module relais 5v 1 canal :



Figure IV.13. Photo réelle de Module relais 5v 1 canal.

g. Les câbles :

On a utilisé des câbles '22 AWG – 4 p et pour établir des connexions entre les éléments du système les broches d'en-tête d'Arduino, on a relié les câbles par des 'jumpers'

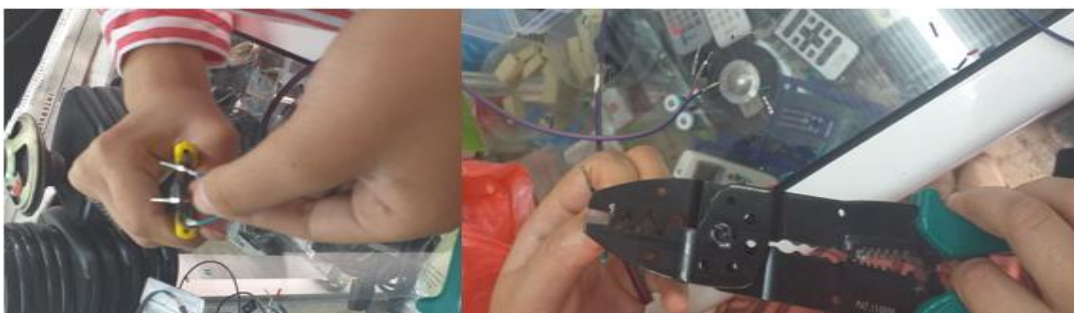


Figure IV.14. Phase de préparation des câbles.



Figure IV.15. câbles avec jumpers.

IV.5. Simulation et test avec Arduino:

IV.5.1. Branchement :

a- la carte Arduino avec PCA9685 et les servomoteurs :

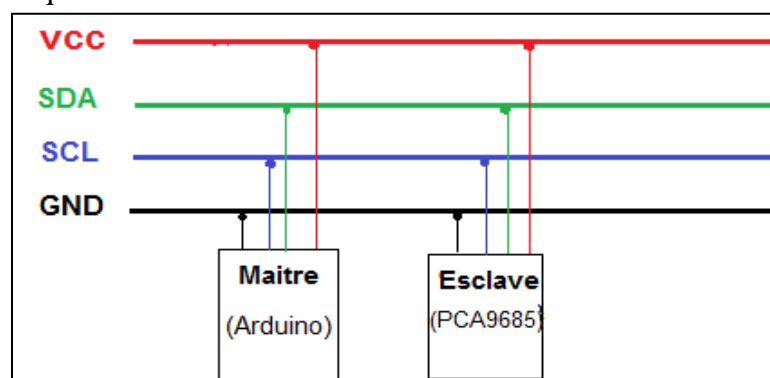
Nous avons utilisé la carte PCA9685 pour contrôler les 16 servomoteurs étant donné que la carte PCA9685 contient le bus I2C qui partage les mêmes signaux SCL/SDA. Le câblage est relativement simple, Nous avons branché les 16 servomoteurs sur les 16 ports du contrôleur de PCA9685 et cette dernière a raccordé avec Arduino sur :

- Numérique20 : SDA
- Numérique21 : SCL pour établir une communication I2C
- +5V : Vcc
- GND : GND pour avoir une masse commune entre le module PCA9685 et Arduino.

Sans oublier que La logique du PCA9685 est alimentée par l'intermédiaire de la broche VCC (+5V). Si cette dernière n'est pas raccordée, la partie logique du circuit n'est pas alimentée, il n'y a donc pas de circuit de commande.

Le BUS I2C :

Le bus I2C est une sorte de réseau présent sur la carte Arduino. Il est utilisé pour piloter des équipements tels que le PCA9685.



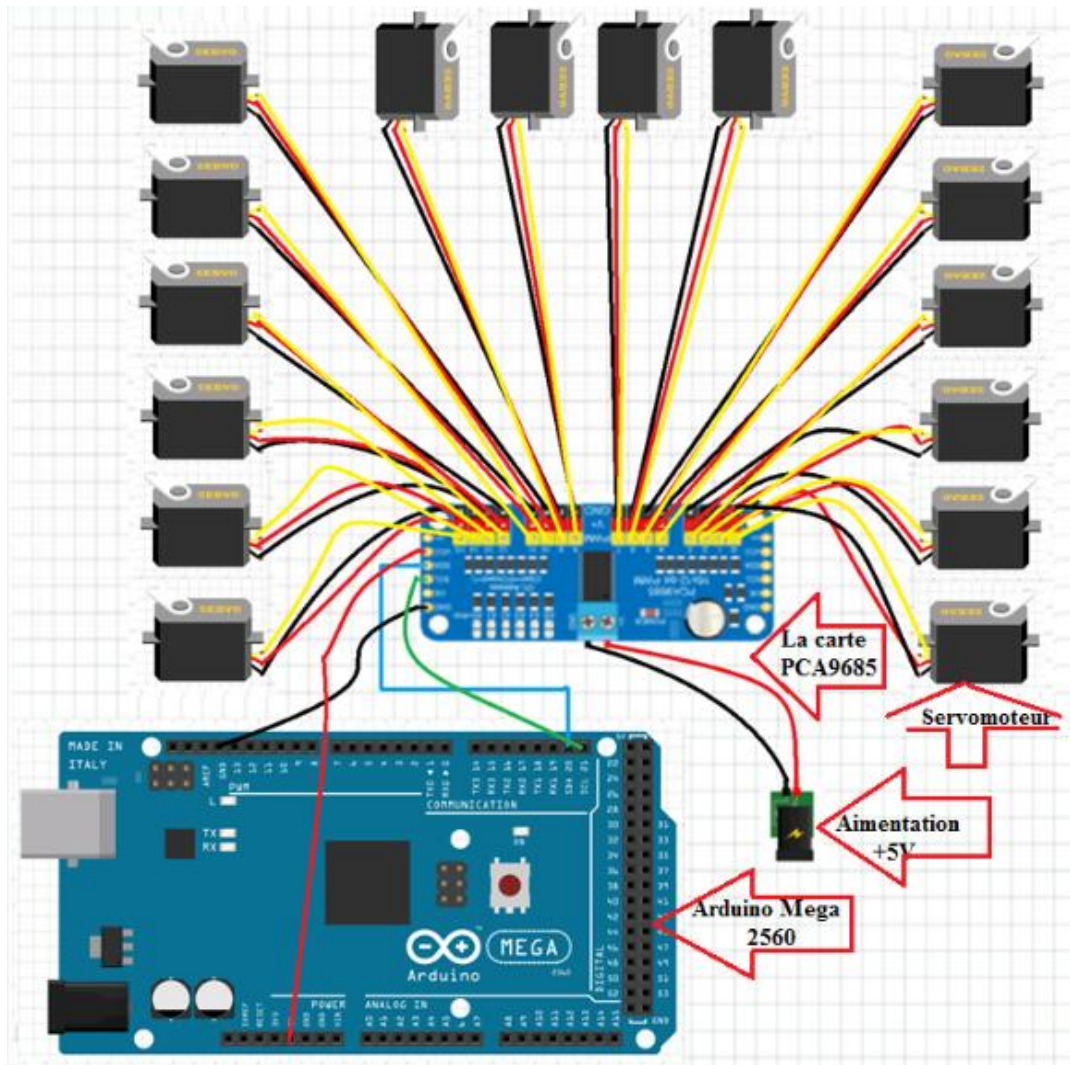


Figure IV.16. Montage d'Arduino méga et module PCA9685 avec les servomoteurs.

b- la carte Arduino avec capteur d'obstacle infrarouge à 4 canaux :

Le branchement de capteur avec Arduino est comme la figure suivante :

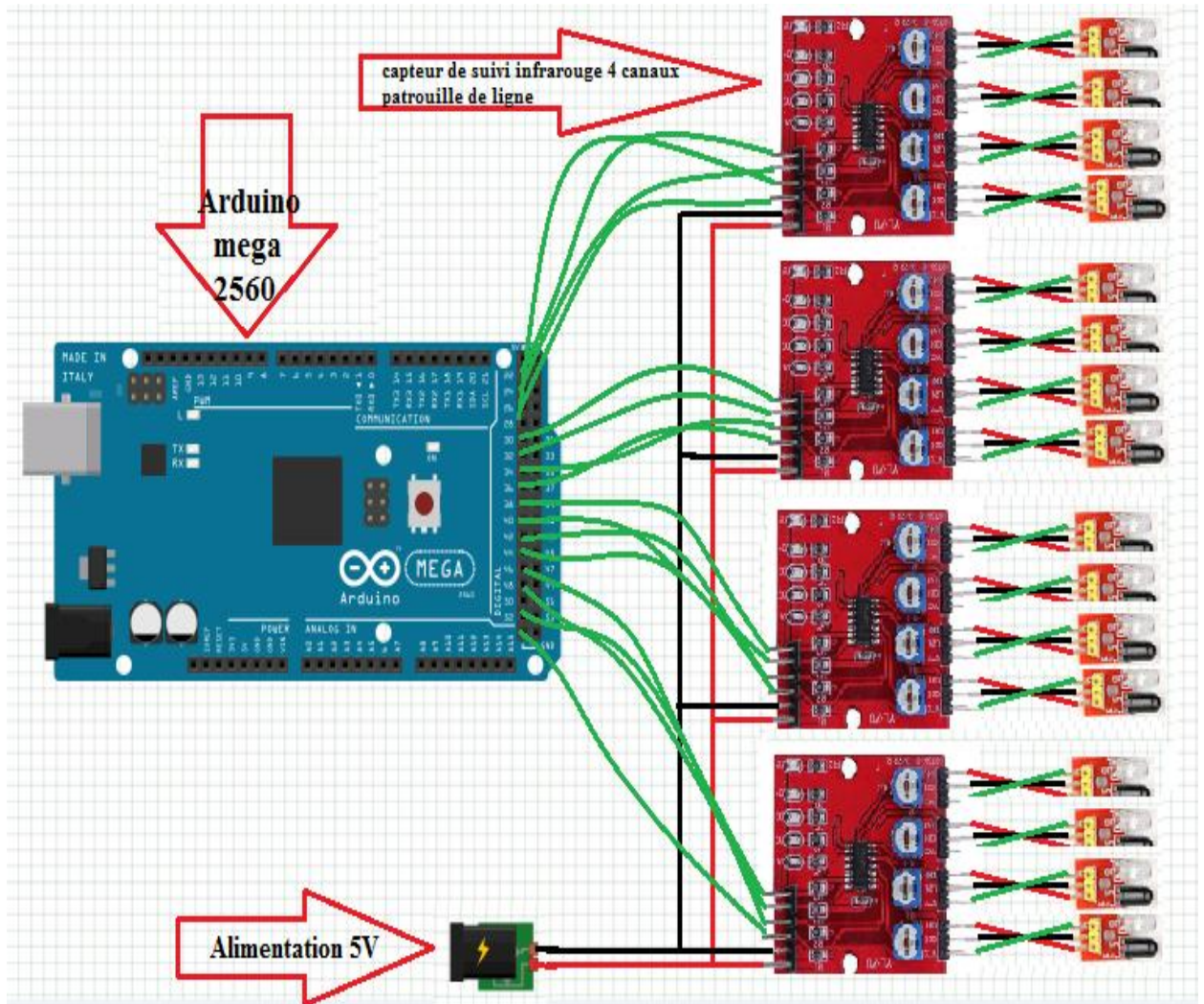


Figure IV.17. Montage d'Arduino méga avec capteur 4 canal infrarouge.

Le tableau(IV.1) représente le branchement du module émetteur avec récepteur :

L'émetteur	Le récepteur
Vcc	Vcc
GND	GND
OUT	OUT

Tableau IV.1. Branchement de module émetteur / récepteur.

Le tableau(IV.2) représente le branchement du module récepteur avec l'Arduino :

Le module récepteur	Arduino
Vcc	+5V
GND	GND
OUT1	Digitale OUT
OUT2	Digitale OUT
OUT3	Digitale OUT
OUT4	Digitale OUT

Tableau IV.2. Branchement de module récepteur/ Arduino.

c- la carte Arduino avec le DC moteur et relais :

Pour notre projet on a utilisé le moteur DC pour le convoyeur.

Le schéma électronique de l'Arduino avec moteur DC et relais sont présenté dans la figure (IV.19)

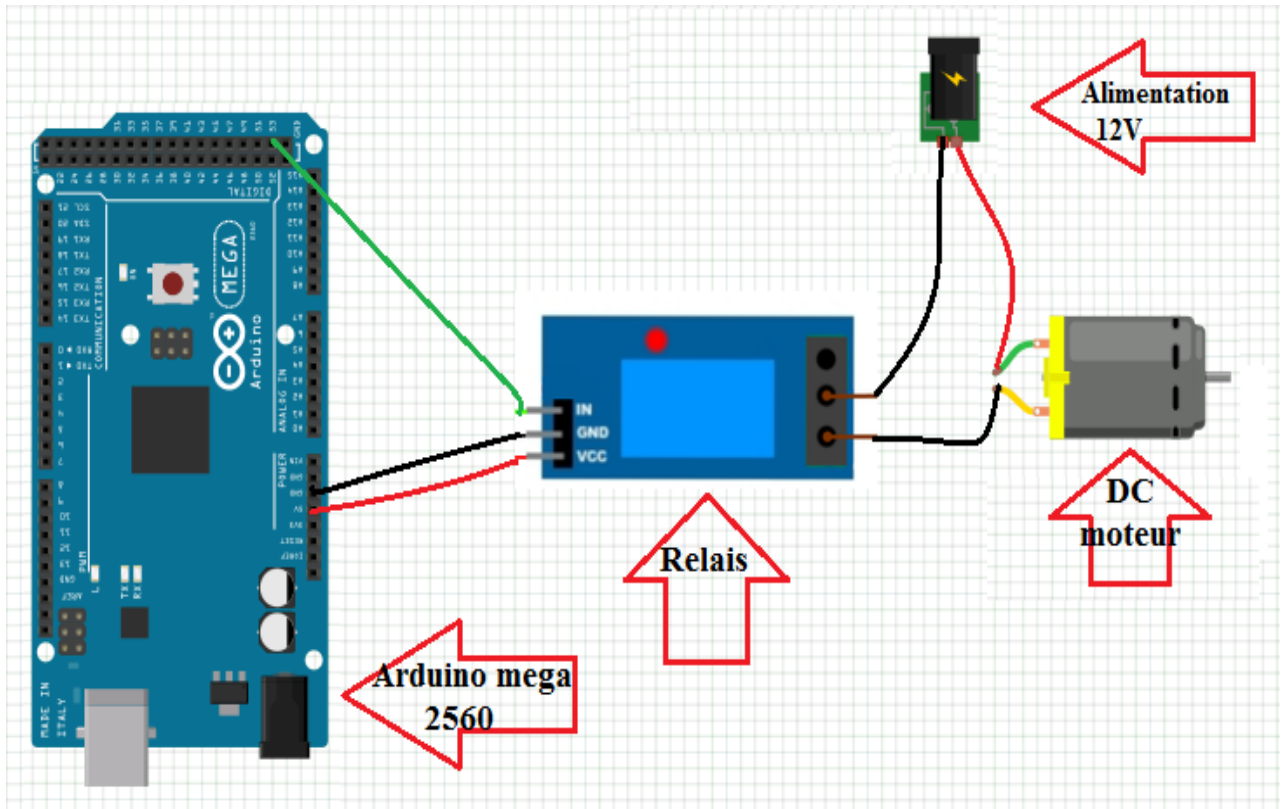


Figure IV.18. Montage d'Arduino méga avec relais et moteur DC .

Le tableau (IV.3) représente le branchement de relais avec Arduino :

Relais	Arduino
Vcc	+5V
GND	GND
OUT	Digitale OUT

Le tableau IV.3. Le branchement de relais avec Arduino.

IV.5.2. Simulation :

Nous avons faire la simulation de notre schéma électrique sur PROTEUS qui nous a permis d’observer s’il y a les défauts sur le montage et pouvoirs remédier aux éventuels problèmes aux quels nous pouvons faire face sans danger.

a- la carte Arduino avec PCA9685 et les servomoteurs :

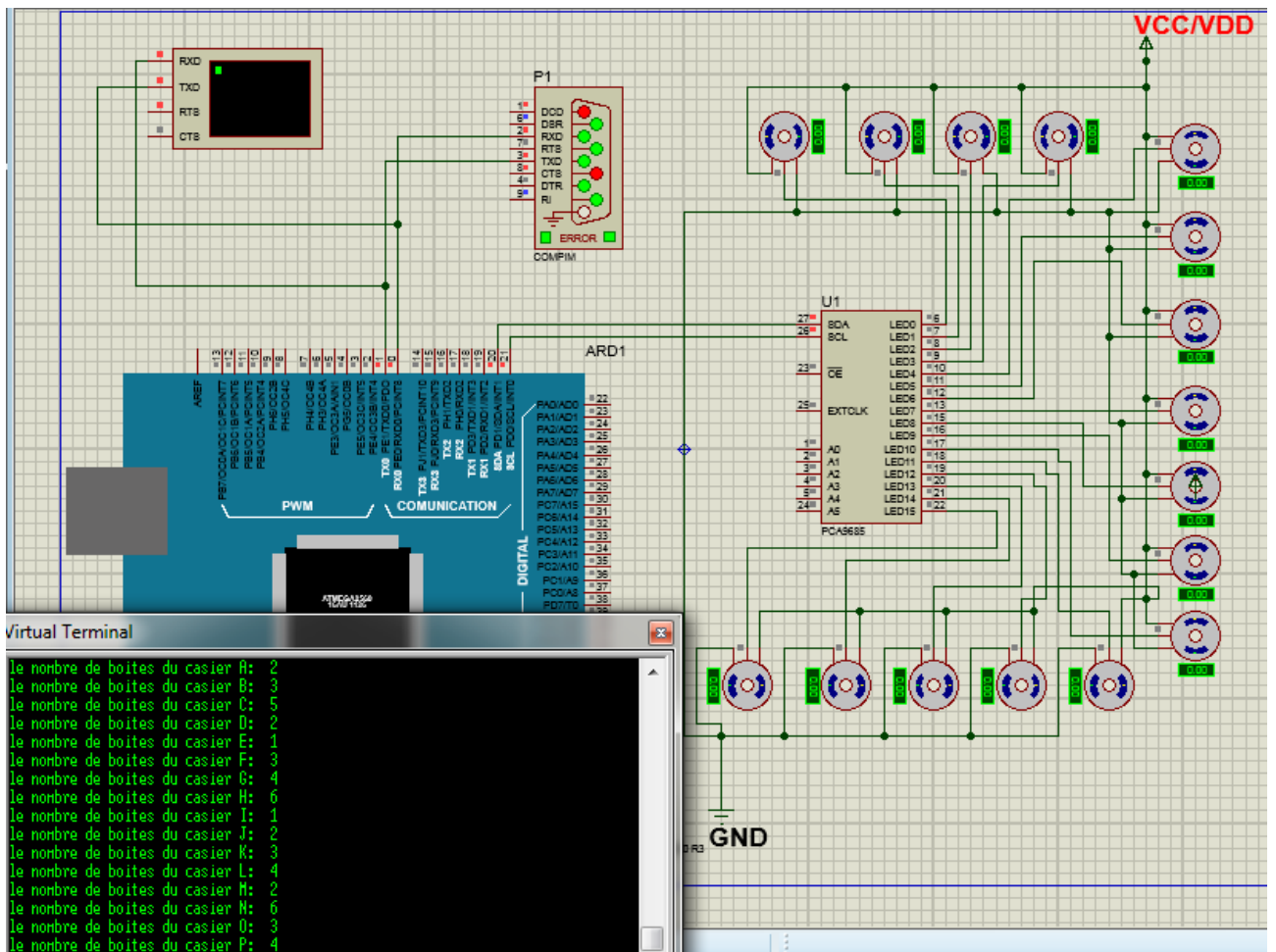


Figure IV.19. Simulation d’Arduino méga et module PCA9685 avec les servomoteurs.

b- la carte Arduino avec capteur 4 canal infrarouge :

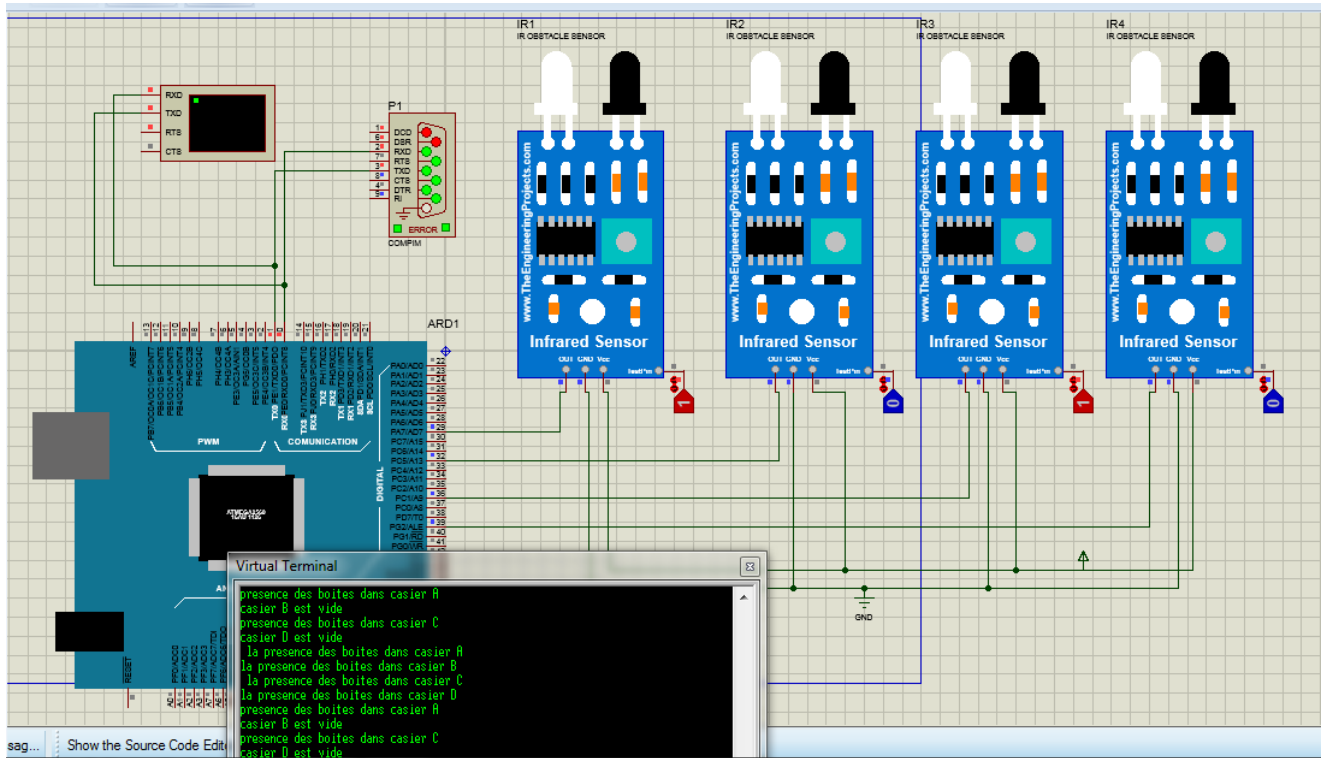


Figure IV.20. Simulation d'Arduino méga avec capteur 4 canal infrarouge.

c- la carte Arduino avec le moteur DC et relais :

Ce montage a été nécessaire pour tester et assurer du bon fonctionnement des équipements.

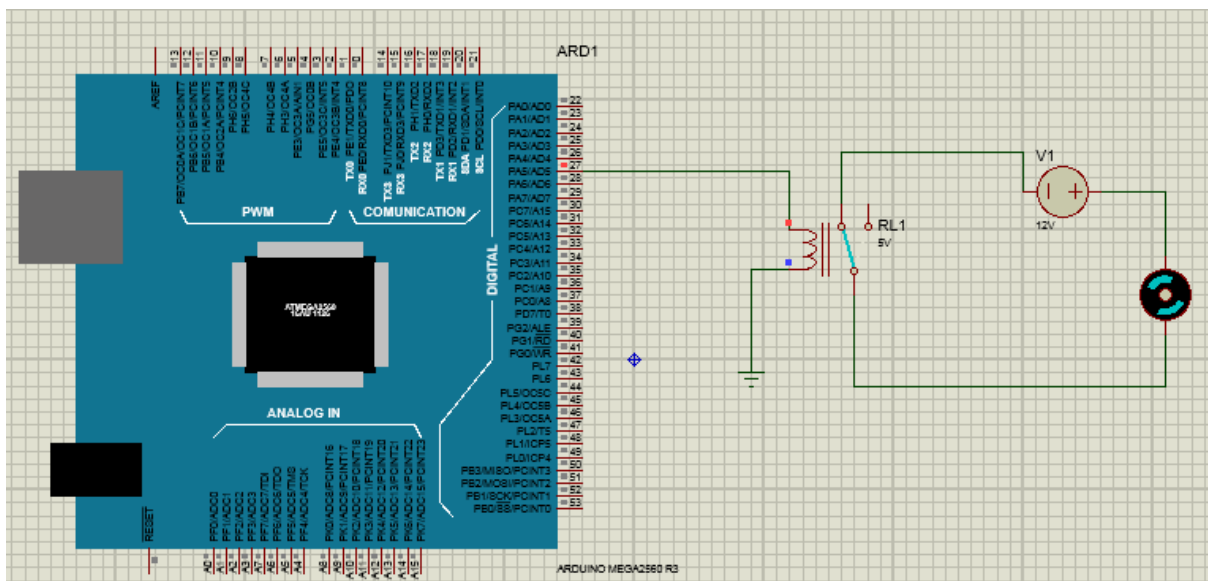


Figure IV.21. Simulation d'Arduino méga avec moteur DC et relais.

IV.5.3. Programmation avec l'Arduino :

Pour écrire le code de module PCA9685 avec l'Arduino il faut utiliser la bibliothèque **Adafruit_PWMServoDriver.h** : `#include <Adafruit_PWMServoDriver.h>`
 Pour mettre en œuvre cette bibliothèque. Il faut tout d'abord télécharger la librairie **Adafruit_PWMServoDriver.h**. Au choix, l'installer dans un répertoire **Adafruit_PWMServoDriver.h** du répertoire Library de l'environnement Arduino IDE ou alors en ajoutant la ligne suivante en haut du programme. Une fois que la bibliothèque est là, alors il nous faudra l'utiliser.

Pour utiliser la bibliothèque **Adafruit_PWMServoDriver.h** il faut d'abord déclarer :

- ✓ Largeur minimal et maximal d'impulsion
- ✓ Largeur d'impulsion par défaut et la fréquence.

```
..
#define MIN_PULSE_WIDTH      650
#define MAX_PULSE_WIDTH      2350
#define DEFAULT_PULSE_WIDTH  1500
#define FREQUENCY             50
```

Aussi la déclaration des variables :

```
String inputString = "";
boolean stringComplete = false;
int A,B,C,D,E,F,G,H,I,J,K,L,M,N,O,P ;
int cmp1, cmp2, cmp3, cmp4, cmp5, cmp6, cmp7, cmp8, cmp9, cmp10, cmp11, cmp12, cmp13, cmp14, cmp15, cmp16 ;
int TIM = 200;
int timecapt=100;
int anglemin=0;
int anglemax=180;
int cap1=22;
int cap2=24;
int cap3=26;
int cap4=28;
int cap5=30;
int cap6=32;
int cap7=34;
int cap8=36;
int cap9=38;
int cap10=40;
int cap11=42;
int cap12=44;
int cap13=46;
int cap14=48;
int cap15=50;
int cap16=52;
int moteur=14;
```

Nous avons utilisé la fonction SerialEvent pour faire la communication avec le moniteur série qui fonctionne par défaut en 9600 bauds, ainsi nous avons réservé 200 octets pour la chaîne.

Aussi nous avons déclaré les capteurs comme des entrées `INPUT` et le moteur comme une sortie `OUTPUT`

```
void setup() {
  pwm.begin();
  pwm.setPWMPfreq(FREQUENCY);
  Serial.begin(9600);
  inputString.reserve(200);
  pinMode(cap1, INPUT);
  pinMode(cap2, INPUT);
  pinMode(cap3, INPUT);
  pinMode(cap4, INPUT);
  pinMode(cap5, INPUT);
  pinMode(cap6, INPUT);
  pinMode(cap7, INPUT);
  pinMode(cap8, INPUT);
  pinMode(cap9, INPUT);
  pinMode(cap11, INPUT);
  pinMode(cap12, INPUT);
  pinMode(cap13, INPUT);
  pinMode(cap14, INPUT);
  pinMode(cap15, INPUT);
  pinMode(cap16, INPUT);
  pinMode(moteur, OUTPUT);
}
```

Le programme a établi une connexion avec le moniteur série pour lire les données reçues, il reçoit une trame de caractères 'a' jusqu'à 'q' qui indique le nom de casier (servo) et entre chaque alphabet on trouve un nombre c'est un compteur associer au chaque servo qui a indiqué le nombre de mouvement de servo.

```
if (stringComplete) {
  A=(inputString.substring(inputString.indexOf('a') + 1,inputString.indexOf('b')).toInt());
  Serial.print("nombre de boites du casier A:"); Serial.println(A);
  B=(inputString.substring(inputString.indexOf('b') + 1,inputString.indexOf('c')).toInt());
  Serial.print("nombre de boites du casier B:"); Serial.println(B);
  C=(inputString.substring(inputString.indexOf('c') + 1,inputString.indexOf('d')).toInt());
  Serial.print("nombre de boites du casier C:"); Serial.println(C);
  D=(inputString.substring(inputString.indexOf('d') + 1,inputString.indexOf('e')).toInt());
  Serial.print("nombre de boites du casier D:"); Serial.println(D);
  E=(inputString.substring(inputString.indexOf('e') + 1,inputString.indexOf('f')).toInt());
  Serial.print("nombre de boites du casier E:");Serial.println(E);
  F=(inputString.substring(inputString.indexOf('f') + 1,inputString.indexOf('g')).toInt());
  Serial.print("nombre de boites du casier F:");Serial.println(F);
  G=(inputString.substring(inputString.indexOf('g') + 1,inputString.indexOf('h')).toInt());
  Serial.print("nombre de boites du casier G:"); Serial.println(G);
  H=(inputString.substring(inputString.indexOf('h') + 1,inputString.indexOf('i')).toInt());
  Serial.print("nombre de boites du casier H:");Serial.println(H);
  I=(inputString.substring(inputString.indexOf('i') + 1,inputString.indexOf('j')).toInt());
  Serial.print("nombre de boites du casier I:"); Serial.println(I);
  J=(inputString.substring(inputString.indexOf('j') + 1,inputString.indexOf('k')).toInt());
  Serial.print("nombre de boites du casier J:"); Serial.println(J);
  K=(inputString.substring(inputString.indexOf('k') + 1,inputString.indexOf('l')).toInt());
  Serial.print("nombre de boites du casier K:");Serial.println(K);
  L=(inputString.substring(inputString.indexOf('l') + 1,inputString.indexOf('m')).toInt());
  Serial.print("nombre de boites du casier L:");Serial.println(L);
  M=(inputString.substring(inputString.indexOf('m') + 1,inputString.indexOf('n')).toInt());
  Serial.print("nombre de boites du casier M:");Serial.println(M);
  N=(inputString.substring(inputString.indexOf('n') + 1,inputString.indexOf('o')).toInt());
  Serial.print("nombre de boites du casier N:"); Serial.println(N);
  O=(inputString.substring(inputString.indexOf('o') + 1,inputString.indexOf('p')).toInt());
  Serial.print("nombre de boites du casier O:"); Serial.println(O);
  P=(inputString.substring(inputString.indexOf('p') + 1,inputString.indexOf('s')).toInt());
  Serial.print("nombre de boites du casier P:");Serial.println(P);

  cmp1 = 0;
  cmp2 = 0;
  cmp3 = 0;
  cmp4 = 0;
  cmp5 = 0;
  cmp6 = 0;
  cmp7 = 0;
  cmp8 = 0;
  cmp9 = 0;
  cmp10 = 0;
  cmp11 = 0;
  cmp12 = 0;
  cmp13 = 0;
  cmp14 = 0;
  cmp15 = 0;
  cmp16 = 0;

  inputString = "";
  stringComplete = false;
}
```


Une fois le programme reçoit le tram, chaque servo va bouger dépend de son compteur.

```

if (cmp1 < A) {
  pwm.setPWM(0, 0, pulseWidth(anglemin));
  delay(TIM);
  pwm.setPWM(0, 0, pulseWidth(anglemax));
  delay(TIM);
  cmp1++;
  if (cmp2 < B) {
    pwm.setPWM(1, 0, pulseWidth(anglemin));
    delay(TIM);
    pwm.setPWM(1, 0, pulseWidth(anglemax));
    delay(TIM);
    cmp2++;
  }
  if (cmp3 < C) {
    pwm.setPWM(2, 0, pulseWidth(anglemin));
    delay(TIM);
    pwm.setPWM(2, 0, pulseWidth(anglemax));
    delay(TIM);
    cmp3++;
  }
  if (cmp4 < D) {
    pwm.setPWM(3, 0, pulseWidth(anglemin));
    delay(TIM);
    pwm.setPWM(3, 0, pulseWidth(anglemax));
    delay(TIM);
    cmp4++;
  }
  if (cmp5 < E) {
    pwm.setPWM(4, 0, pulseWidth(anglemin));
    delay(TIM);
    pwm.setPWM(4, 0, pulseWidth(anglemax));
    delay(TIM);
    cmp5++;
  }
  if (cmp6 < F) {
    pwm.setPWM(5, 0, pulseWidth(anglemin));
    delay(TIM);
    pwm.setPWM(5, 0, pulseWidth(anglemax));
    delay(TIM);
    cmp6++;
  }
  if (cmp7 < G) {
    pwm.setPWM(6, 0, pulseWidth(anglemin));
    delay(TIM);
    pwm.setPWM(6, 0, pulseWidth(anglemax));
    delay(TIM);
    cmp7++;
  }
  if (cmp8 < H) {
    pwm.setPWM(7, 0, pulseWidth(anglemin));
    delay(TIM);
    pwm.setPWM(7, 0, pulseWidth(anglemax));
    delay(TIM);
    cmp8++;
  }
  if (cmp9 < I) {
    pwm.setPWM(8, 0, pulseWidth(anglemin));
    delay(TIM);
    pwm.setPWM(8, 0, pulseWidth(anglemax));
    delay(TIM);
    cmp9++;
  }
  if (cmp10 < J) {
    pwm.setPWM(9, 0, pulseWidth(anglemin));
    delay(TIM);
    pwm.setPWM(9, 0, pulseWidth(anglemax));
    delay(TIM);
    cmp10++;
  }
  if (cmp11 < K) {
    pwm.setPWM(10, 0, pulseWidth(anglemin));
    delay(TIM);
    pwm.setPWM(10, 0, pulseWidth(anglemax));
    delay(TIM);
    cmp11++;
  }
  if (cmp12 < L) {
    pwm.setPWM(11, 0, pulseWidth(anglemin));
    delay(TIM);
    pwm.setPWM(11, 0, pulseWidth(anglemax));
    delay(TIM);
    cmp12++;
  }
  if (cmp13 < M) {
    pwm.setPWM(12, 0, pulseWidth(anglemin));
    delay(TIM);
    pwm.setPWM(12, 0, pulseWidth(anglemax));
    delay(TIM);
    cmp13++;
  }
  if (cmp14 < N) {
    pwm.setPWM(13, 0, pulseWidth(anglemin));
    delay(TIM);
    pwm.setPWM(13, 0, pulseWidth(anglemax));
    delay(TIM);
    cmp14++;
  }
  if (cmp15 < O) {
    pwm.setPWM(14, 0, pulseWidth(anglemin));
    delay(TIM);
    pwm.setPWM(14, 0, pulseWidth(anglemax));
    delay(TIM);
    cmp15++;
  }
  if (cmp16 < P) {
    pwm.setPWM(15, 0, pulseWidth(anglemin));
    delay(TIM);
    pwm.setPWM(15, 0, pulseWidth(anglemax));
    delay(TIM);
    cmp16++;
  }
}

int pulseWidth(int angle)
{
  int pulse_wide, analog_value;
  pulse_wide = map(angle, 0, 180, MIN_PULSE_WIDTH, MAX_PULSE_WIDTH);
  analog_value = int(float(pulse_wide) / 1000000 * FREQUENCY * 4096);
  return analog_value;
}

```

Pour commander les capteurs nous avons utilisé l'instruction `digitalRead(cap1)` cette instruction permet de lire soit un niveau haut (« 1 » ou HIGH) le cas de la présence des boîtes dans le casier, soit un niveau bas (« 0 » ou LOW) le cas casier est vide.

```

int detect = digitalRead(cap1);
if(detect == LOW){ Serial.println("la présence des boîtes dans casier A"); }
else{Serial.println("casier A est vide ")};

int detectB = digitalRead(cap2);
if(detectB == LOW){ Serial.println("la présence des boîtes dans casier B"); }
else{Serial.println("casier B est vide ")};

int detectC = digitalRead(cap3);
if(detectC == LOW){ Serial.println("la présence des boîtes dans casier C"); }
else{Serial.println("casier C est vide ")};

int detectD = digitalRead(cap4);
if(detectD == LOW){ Serial.println("la présence des boîtes dans casier D"); }
else{Serial.println("casier D est vide ")};

int detectE = digitalRead(cap5);
if(detectE == LOW){ Serial.println("la présence des boîtes dans casier E"); }
else{Serial.println("casier E est vide ")};

int detectF = digitalRead(cap6);
if(detectF == LOW){ Serial.println("la présence des boîtes dans casier F"); }
else{Serial.println("casier F est vide ")};

int detectG = digitalRead(cap7);
if(detectG == LOW){ Serial.println("la présence des boîtes dans casier G"); }
else{Serial.println("casier G est vide ")};

int detectH = digitalRead(cap8);
if(detectH == LOW){ Serial.println("la présence des boîtes dans casier H"); }
else{Serial.println("casier H est vide ")};

int detectI = digitalRead(cap9);
if(detectI == LOW){ Serial.println("la présence des boîtes dans casier I"); }
else{Serial.println("casier I est vide ")};

int detectJ = digitalRead(cap10);
if(detectJ == LOW){ Serial.println("la présence des boîtes dans casier J"); }
else{Serial.println("casier J est vide ")};

int detectK = digitalRead(cap11);
if(detectK == LOW){ Serial.println("la présence des boîtes dans casier K"); }
else{Serial.println("casier K est vide ")};

int detectL = digitalRead(cap12);
if(detectL == LOW){ Serial.println("la présence des boîtes dans casier L"); }
else{Serial.println("casier L est vide ")};

int detectM = digitalRead(cap13);
if(detectM == LOW){ Serial.println("la présence des boîtes dans casier M"); }
else{Serial.println("casier M est vide ")};

int detectN = digitalRead(cap14);
if(detectN == LOW){ Serial.println("la présence des boîtes dans casier N"); }
else{Serial.println("casier N est vide ")};

int detectO = digitalRead(cap15);
if(detectO == LOW){ Serial.println("la présence des boîtes dans casier O"); }
else{Serial.println("casier O est vide ")};

int detectP = digitalRead(cap16);
if(detectP == LOW){ Serial.println("la présence des boîtes dans casier P"); }
else{Serial.println("casier P est vide ")};
delay(timecapt);

```

Pour commander le convoyeur (le moteur DC) il faut utiliser l'instruction `digitalWrite` qui permet d'envoyer soit 5V (HIGH) sur la broche, soit 0V (LOW).

IV.6. Test et réalisation finale :**IV.6.1. Test du module PCA9685 :**

On a testé la fonctionnalité du PCA968 Servo Driver via un programme de 16 servomoteurs, une fois on a téléversé le programme dans la carte Arduino on a remarqué les rotations des servomoteurs.



Figure IV.22. Montage réel d'Arduino méga et module PCA9685
Avec les servomoteurs.

IV.6.2. Test de Capteur d'obstacle infrarouge à 4 canaux :

On a testé le fonctionnement de capteur avec l'Arduino, mettant un obstacle devant le module émetteur avec un réglage de distance bien définie le moniteur série affiche 'la présence d'obstacles. Comme le montre la figure (IV.2)

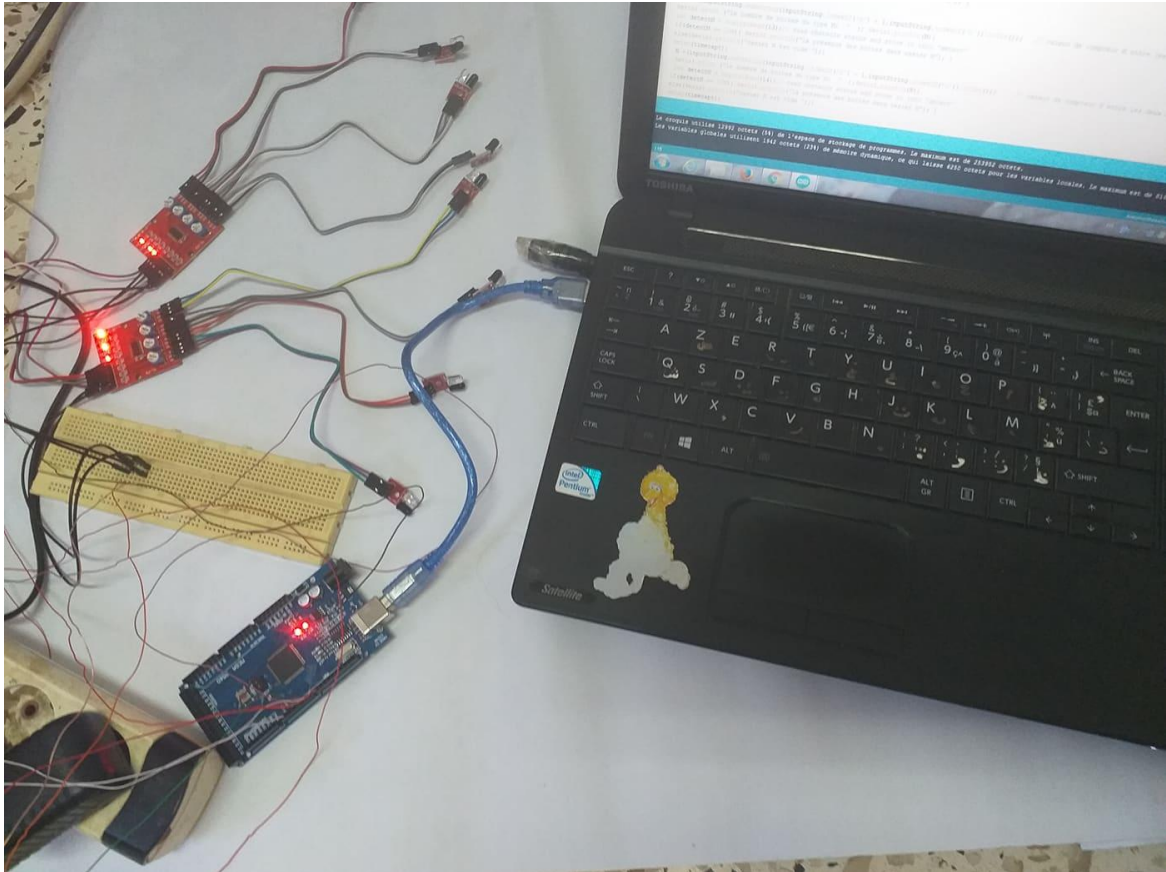


Figure IV.23. Test réel du capteur d'obstacle infrarouge a 4 canaux

IV.7. Réalisation finale du système :

IV.7.1. Configurations nécessaires :

Pour faire fonctionner le système, plusieurs réglages, installations et configurations sont nécessaires, parmi ceux-ci, nous citons:

- ✓ Installation des servomoteurs
- ✓ Installation des capteurs : Les capteurs doivent être mis en place dans les endroits désignés (à coté de servomoteur) afin de détecter la présence de la boîte dans les casiers, puis ils doivent être connectés à la carte électronique Arduino.
- ✓ L'installation de convoyeur.

Après l'installation des cartes électroniques, capteurs, modules, actionneurs, il est nécessaire de faire la liaison avec la carte Arduino.

IV.7.2. Validation et tests :

Après avoir achevé le montage sur la maquette, la phase de test est nécessaire pour s'assurer du bon fonctionnement du système.

On a envoyé les données via le port série à l'Arduino sous forme d'une trame contenant les informations nécessaires (le nom du casier et le nombre de boîtes ont distribué), les servomoteurs sont

actionnés et le nombre des boîtes dans chaque casier est affiché dans la fenêtre de moniteur série ainsi que les résultats de détection par les capteurs (présence des boîtes ou non dans chaque casier).

La figure (IV.23) une photo réel de système globale.

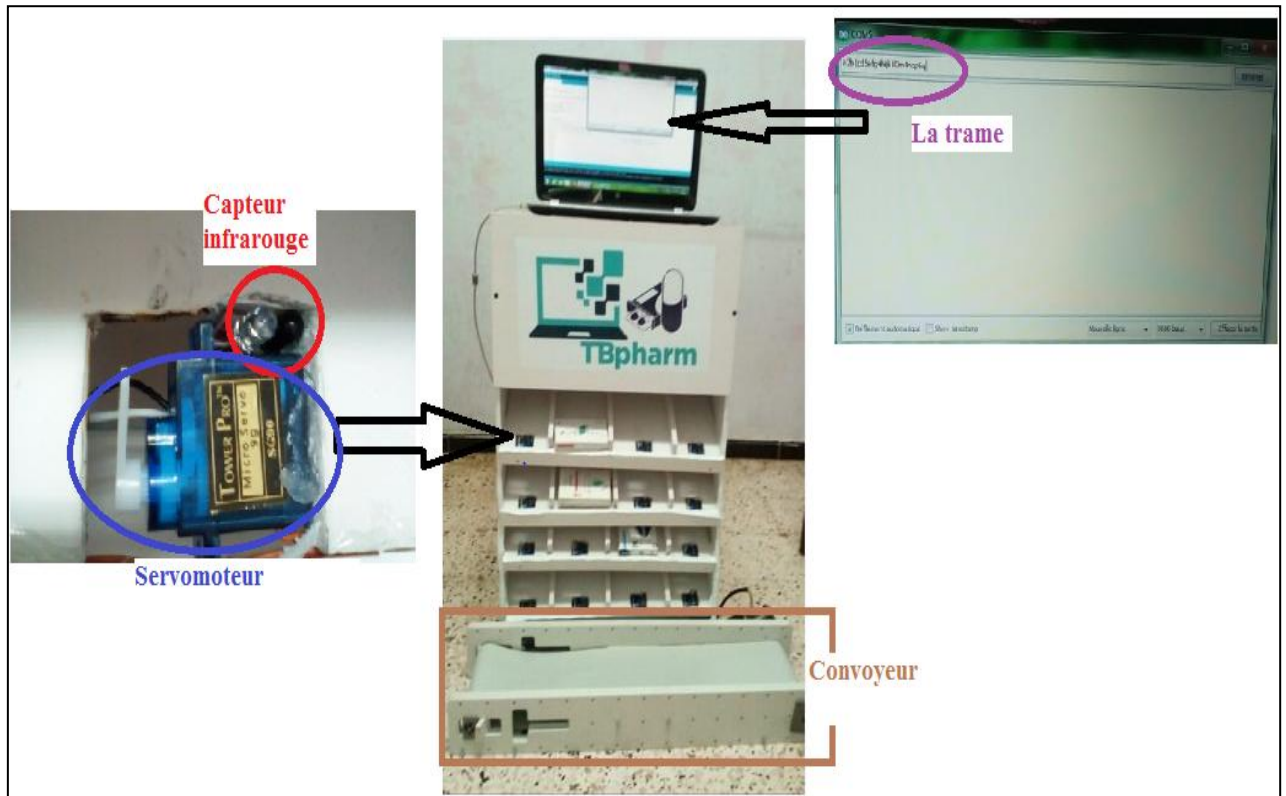


Figure IV.24. Photo réelle de système globale

IV.8. Conclusion :

Dans ce chapitre, nous avons présenté en détail les étapes de réalisation de notre système. Nous avons commencé par la description de notre maquette et la justification du choix des composants. Ensuite, Nous décrivons et réalisons la phase de simulation et de programmation avec Arduino qui est validé par l'étape suivante 'réalisation finale', c'était le fruit de notre travail.

Nous avons fabriqué un système automatisé de distribution des médicaments capable de gérer à distance.

Conclusion générale :

Ce mémoire a présenté une étude pratique qui explique la conception et la réalisation d'un système automatisé de distribution des médicaments, l'objectif de ce système est de faciliter la distribution des médicaments dans les pharmacies et d'améliorer la gestion et la qualité de leur travail.

À la lumière de notre projet nous avons fabriqué une maquette flow rack à chute libre de 16 casiers ainsi qu'un convoyeur pour transporter les médicaments.

Ce projet de réalisation nous a été une expérience très fructueuse, bénéfique et enrichissante, il nous a permis d'exploiter les différentes notions de programmation par l'Arduino. Dans une première étape nous avons développé un programme qui commande par ordinateur, 16 servomoteurs pilotés à l'aide d'un module PCA9685, pour provoquer la chute libre des médicaments vers le convoyeur. Ensuite nous avons installé des capteurs d'obstacle infrarouges afin de détecter la présence ou non des boîtes dans les casiers. Également, nous avons appris à concevoir, contrôler et commander un convoyeur.

Nous avons cherché à minimiser les coûts de réalisation d'un système automatisé de distribution de médicaments par un système de commande Arduino au lieu d'un automate programmable qui coute plus cher. Le coût de ce projet est estimé à 29400 DA. L'intérêt de ce projet a été double pour nous. D'une part il constituera un prototype réaliste permettant de faire des démonstrations pour les professionnels qui voudraient acquérir ce système, et d'autre part il servira comme une maquette didactique pour les étudiants en travaux pratiques d'automatisation et de conception d'Interfaces Homme-Machine.

Ce travail reste, comme toute œuvre humaine, incomplète et perfectible, nous recommandons d'en améliorer la conception et la réalisation et pour cela nous proposons ci-dessous des améliorations pour les futurs développements :

- Augmenter la capacité de stockage (nombre des casiers).
- Ajouter un module Wi-Fi qui permette la connexion sans fil entre le système et le PC.

- [1]. Alpha A-A. La guerre des robots. Pharmacien Manager [Internet]. 7 mai 2013 [cité 4 janv2016];(127). Disponible sur: <http://www.lemoniteurdespharmacies.fr/revues/pharmacien-manager/article/n-127/la-guerre-des-robots.html>
- [2]. Huret N. Automatisation et robotisation à l'officine: avantages et inconvénients [Thèse D'exercice]. [Lille, France]: Université du droit et de la santé; 2009. Disponible sur: <http://thesesante.ups-tlse.fr/1876/1/2017TOU32045.pdf>
- [3]. Pouzaud F. Automates et robots des solutions pour petits et grands. Le Moniteur des Pharmacies [Internet]. 29 juin 2013 [cité 25 sept 2016];(2990). Disponible sur: <https://www.lemoniteurdespharmacies.fr/revues/le-moniteur-des-pharmacies/article/n-2990/automates-et-robots-des-solutions-pour-petits-et-grands.html>
- [4]. Airtec-Tecnilab [Internet]. Tecnilab.com. [cité 9 oct 2016]. Disponible sur: <http://www.tecnilab.com/fra/convoyage-et-transitique/airtec-systeme-pneumatique.html>
- [5]. Le Moniteur des Pharmacies. Transitique : Roulez business ! Le Moniteur des Pharmacies [Internet]. 27 nov 2004 [cité 25 sept 2016];(2559).
- [6]. Convoyage - Mékapharm [Internet]. Mekapharm.com. [cité 9 oct 2016]. Disponible sur: <http://mekapharm.com/le-convoyage/>
- [7]. Documentation publicitaire PlusInfo-PEEM, 2000-2001
- [8]. Site Internet ApotéKa.
- [9]. Site Internet TECNILAB [26])
- [10]. SAUREL V. Voici venu le temps des automates Le Moniteur des Pharmacies et des Laboratoires, 2001,1, n02395, p 72-74
- [11]. COMTEP. Stock automatisé pour officine humanisée Th : Pharmacie: Montpellier I : 2000, 94 f.
- [12]. Documentation publicitaire FAHRENBERGER TECHNOLOGIE – Automate Ariana la liberté, 200
- [13]. Site Internet FAHRENBERGER 1231.

- [14]. Site PEEM disponible sur Internet : <http://www.peem.com/>. juillet 2001
- [15]. Documentation publicitaire TECNY-FARMA- Pharmatrack S I036,2001
- [16]. Documentation publicitaire TECNY· FARMA
- [17]. Site Internet FAHRENBERGER [23] et document publicitaire FAHRENBERGER TECHNOLOGIE [14].
- [18]. PHARMAGORA (2001 ; Paris)
- [19]. Documentation publicitaire ARX - ROWA, 200 1
- [20]. Documentation publicitaire ARX - Unguator, 2001
- [21]. documentation 171129_PRO_fr_komplett.pdf.
- [22]. Documentation publicitaire ARX /161.
- [23]. Documentation technique PlusInfo-PEEM, Pourquoi choisir le Pharmamat, 2001
- [24]. Documentation publicitaire TECNILAB LOGISTIC DIVISION- Movetec, 2001
- [25]. Documentation publicitaire FAHRENBERGER TECHNOLOGIE – Automate Ariana la liberté, 2001
- [26]. Cassette VHS Publicitaire L'automate ApotéKa, mars 2001.
- [27]. 25, 26 mars 2001.Paris-Expo, Parc des Expositions, Porte de Versailles, 75015 Paris, Hall 7 niveau 3
- [28]. BREUILLARD N. L'automatisation de la dispensation en milieu hospitalier: le modèle Américain et Son application en FranceTh : Pharmacie: Paris V : 1999, pagination multiple .LEROYG.
- [29]. VINCENTN. Regards sur l'agencement officinalTh : Pharmacie: Clermont I : 1999, 130 f.
- [30]. De l'agencement d'espaces séducteurs dans l'officine à un service de santé personnaliséTh : Pharmacie: Nancy I : 2000, 105
- [31]. Linn, R.J., and Wysk, R.A. 1990 «An expert system based controller for an automated storage/retrieval system» International Journal of Production Research 28(4), 735-756.

- [32]. Lee, H.F., and Schaefer, S.K. 1996 «Retrieval sequencing for unit-load automated storage and retrieval systems with multiple openings» *International Journal of Production Research* 34(10), 2943-2962.
- [33]. Manzini, R. 2006 «Correlated storage assignment in an order picking system» *International Journal of Industrial Engineering* 13(4), 384-394.
- [34]. Material Handling Institute, 1977. Consideration for planning and installing an automated storage/retrieval systems, Inc., AS/RS Document-100 7M.
- [35]. H.F., and Schaefer, S.K. 1996 «Retrieval sequencing for unit-load automated Storage and retrieval systems with multiple openings» *International Journal of Production Research* 34(10), 2943-2962.
- [36]. Van den Berg, J.P., and Gademann, A.J.R.M. 2000 «Simulation study of an automated storage/retrieval system» *International Journal of Production Research* 38(6), 1339-1356.
- [37]. Roodbergen K. J. and I. F. A. Vis, 2009 «A survey of literature on automated storage and retrieval systems» *European Journal of Operational Research*, 194, p. 343-362
- [38]. J.A. White Y.A. Bozer. Optimum design of automated storage/retrival systems.*TIMS/ORSA Joint National Meeting, Washington, DC, 1980. 16*
- [39]. H.D. Kinney J.A. White. Storage and warehousing in handbook of industrial engineering. Salvendy. *John Wiley and sons, 1982. 16*
- [40]. Documentation technique Phi-CONCEPT - Notice d'implantation ApotéKa, 2001
- [41]. Frédéric MERY. Thèse doctorat en pharmacie .2001.p 41-43
- [42]. M. METAHRI Dhiyaeddine. Thèse doctorat en électromécanique.2019.p.4
- [43]. M. METAHRI Dhiyaeddine. Thèse doctorat en électromécanique.2019.p.48
- [44]. M. METAHRI Dhiyaeddine. Thèse doctorat en électromécanique.2019.p.47

- [45]. Metahri D, Hachemi K (2017) Automated storage and retrieval systems: A performances comparison between Free-fall-flow-rack and classic flow-rack. In: Systems and control (ICSC), 2017 6th international conference on IEEE, pp 589–594 Google Scholar
- [46]. Rosenblatt MJ, Roll Y, Vered Zyser D (1993) A combined optimization and simulation approach for designing automated storage/retrieval systems. IIE Trans 25(1):40–50C Google Scholar
- [47]. <http://mekapharm.com/apoteka/>. Accédé le 22 Oct 2016
- [48]. site consulté on ligne : <https://wiki.mdl29.net/.../fetch.php...Arduino-pour-bien-commencer-en-électronique.pdf>
- [49]. Jean-Noël Montagné, « Livret Arduino en français », Centre de Ressources Art Sensitif sous licence CC, novembre 2006.
- [50]. site consulté on ligne : https://arduino.technologiescollege.fr/IMG/pdf/cahier_0_initialisation.
- [51]. Logiciel Arduino 1.8.1, outil capteur. PC, 2019.
- [52]. site consulté en ligne : http://www.planete-sciences.org/iledefrance/images/psidf/pdf/Fiche_F4_Commander_des_servomoteurs_classiques_et_a_rotation_continue.pdf
- [53]. site consulté en ligne : <http://users.polytech.unice.fr/~ferrero/TPelec2/arduino4.pdf> kshop
- [54]. Hamid HAMOUCHI. *Conception et réalisation d'une centrale embarquée de la domotique « Smart Home »*. 06/07/2015, mémoire de master en génie électrique, Université Mohammed V École Normale Supérieure d'Enseignement Technique – Rabat.
- [55]. site consulté en ligne : http://jp79dsfr.free.fr/_Docs%20et%20infos/Elec%20_%20Arduino%20-%20Bus%20I2c.pdf
- [56]. site consulté en ligne : http://robotix.ah-oui.org/user_docs/dos10/AdaFruit%20PWM%20Driver.pdf

[57]. www.elec.jbd.free.fr, « Machine électromagnétique », Chapitre 13.

[58]. Luc Lasne, Notions de base et machine électrique, Dunod, Paris, 2005.

[59]. A.ABDERRAHMAN , M.MOHAMMED, « Automatisation et Réalisation a petite échelle (maquette) d'une chaine transporteuse de briques ».université chlef. , 2016

[60]. site consulté en ligne : https://wiki.mdl29.net/lib/exe/fetch.php?media=elec:arduino_dossier_ressource.pdf

Câblages :

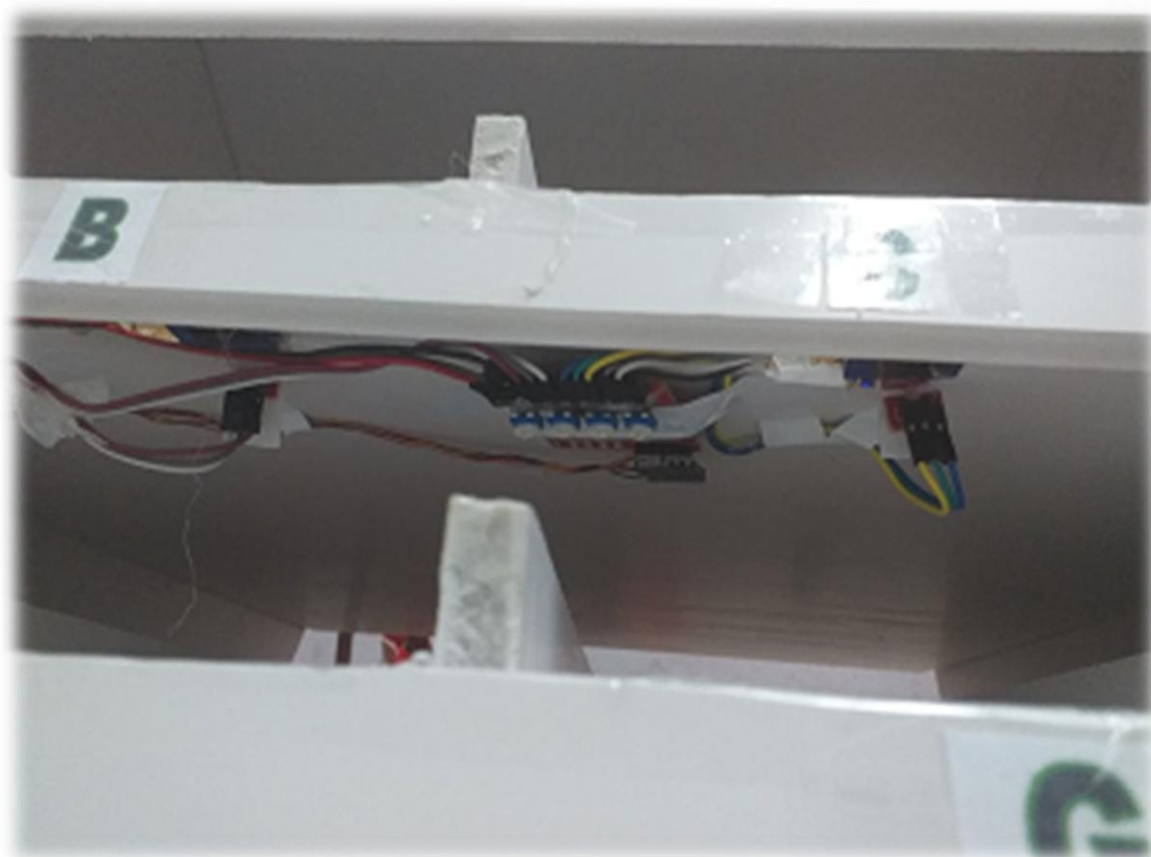
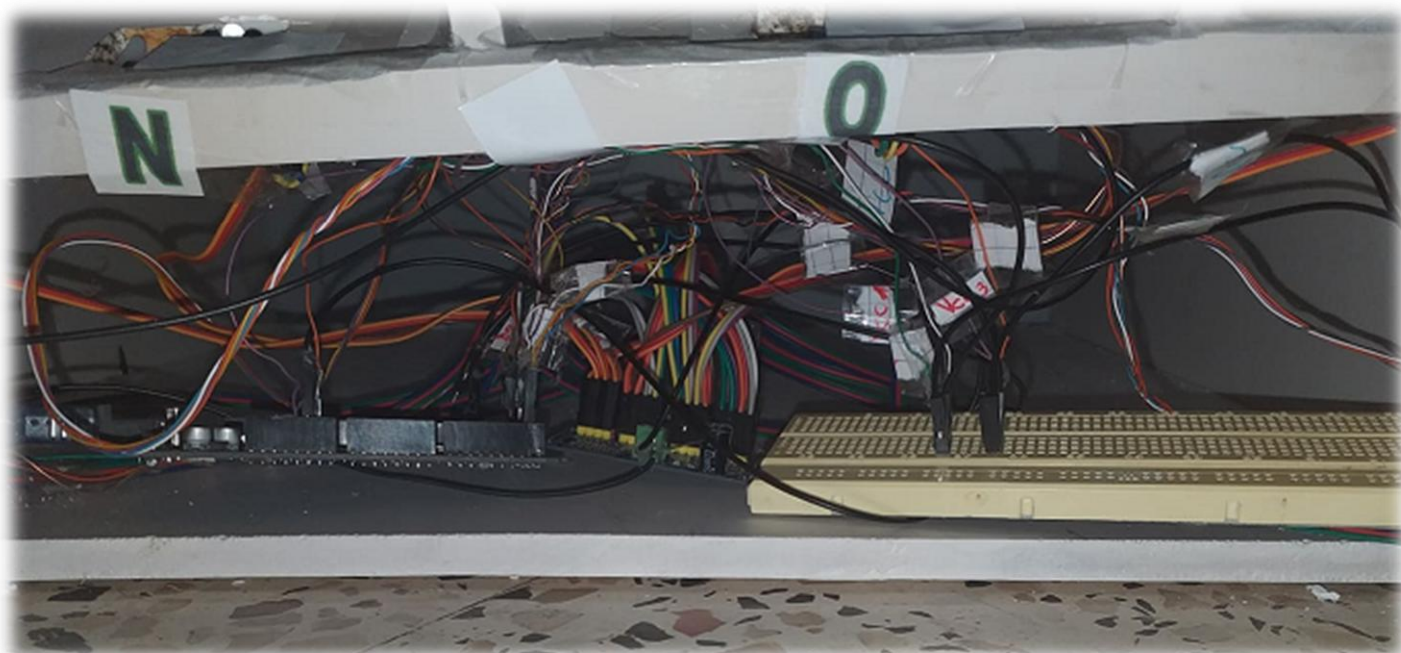


Tableau des pins utilisé de la carte ARDUINO :

Matérielles	PIN
Capteur Infrarouge pour casier A	D22
Capteur Infrarouge pour casier B	D24
Capteur Infrarouge pour casier C	D26
Capteur Infrarouge pour casier D	D28
Capteur Infrarouge pour casier E	D30
Capteur Infrarouge pour casier F	D32
Capteur Infrarouge pour casier G	D34
Capteur Infrarouge pour casier H	D36
Capteur Infrarouge pour casier I	D38
Capteur Infrarouge pour casier J	D40
Capteur Infrarouge pour casier K	D42
Capteur Infrarouge pour casier L	D44
Capteur Infrarouge pour casier M	D46
Capteur Infrarouge pour casier N	D48
Capteur Infrarouge pour casier O	D50
Capteur Infrarouge pour casier P	D52
Bus I2C	SDL : 20 SCL : 21
DC moteur	D53

Le code Arduino de projet :

```
#include <Wire.h>
```

```
#include <Adafruit_PWMServoDriver.h>
```

```
Adafruit_PWMServoDriver pwm = Adafruit_PWMServoDriver();
```

```
// on définit la valeur standard du servo
```

```
#define MIN_PULSE_WIDTH    650
```

```
#define MAX_PULSE_WIDTH    2350
```

```
#define DEFAULT_PULSE_WIDTH 1500
```

```
#define FREQUENCY          50
```

```
String inputString = "";
```

```
boolean stringComplete = false;
```

```
int A,B,C,D,E,F,G,H,I,J,K,L,M,N,O,P ;
int cmp1,cmp2,cmp3,cmp4,cmp5,cmp6,cmp7,cmp8,cmp9,cmp10,cmp11,cmp12,cmp13,cmp14,cmp15,cmp16 ;
int TIM = 200;
int timecapt=100;
int anglemin=0;
int anglemax=180;
int cap1=22;
int cap2=24;
int cap3=26;
int cap4=28;
int cap5=30;
int cap6=32;
int cap7=34;
int cap8=36;
int cap9=38;
int cap10=40;
int cap11=42;
int cap12=44;
int cap13=46;
int cap14=48;
int cap15=50;
int cap16=52;
int moteur=53;

void setup() {
  pwm.begin();
  pwm.setPWMFreq(FREQUENCY);
  Serial.begin(9600);
  inputString.reserve(200);
  pinMode(cap1, INPUT);
  pinMode(cap2, INPUT);
  pinMode(cap3, INPUT);
  pinMode(cap4, INPUT);
  pinMode(cap5, INPUT);
  pinMode(cap6, INPUT);
```

```
pinMode(cap7, INPUT);
pinMode(cap8, INPUT);
pinMode(cap9, INPUT);
pinMode(cap11, INPUT);
pinMode(cap12, INPUT);
pinMode(cap13, INPUT);
pinMode(cap14, INPUT);
pinMode(cap15, INPUT);
pinMode(cap16, INPUT);
pinMode(moteur,OUTPUT);

  cmp1 = 0;
  cmp2 = 0;
  cmp3 = 0;
  cmp4 = 0;
  cmp5 = 0;
  cmp6 = 0;
  cmp7 = 0;
  cmp8 = 0;
  cmp9 = 0;
  cmp10 = 0;
  cmp11 = 0;
  cmp12 = 0;
  cmp13 = 0;
  cmp14 = 0;
  cmp15 = 0;
  cmp16 = 0;

}

void loop() {

  if (stringComplete) {

  capteur();
```

```
A =(inputString.substring(inputString.indexOf('a') + 1,inputString.indexOf('b')).toInt());  
Serial.print ("le nombre de boites du casier A: " ); Serial.println(A);
```

```
B =(inputString.substring(inputString.indexOf('b') + 1,inputString.indexOf('c')).toInt());  
Serial.print ("le nombre de boites du type B: " ); Serial.println(B);
```

```
C =(inputString.substring(inputString.indexOf('c') + 1,inputString.indexOf('d')).toInt());  
Serial.print ("le nombre de boites du type C: " );Serial.println(C);
```

```
D =(inputString.substring(inputString.indexOf('d') + 1,inputString.indexOf('e')).toInt());  
Serial.print ("le nombre de boites du type D: " );Serial.println(D);
```

```
E =(inputString.substring(inputString.indexOf('e') + 1,inputString.indexOf('f')).toInt());  
Serial.print ("le nombre de boites du type E: " );Serial.println(E);
```

```
F =(inputString.substring(inputString.indexOf('f') + 1,inputString.indexOf('g')).toInt());  
Serial.print ("le nombre de boites du type F: " );Serial.println(F);
```

```
G =(inputString.substring(inputString.indexOf('g') + 1,inputString.indexOf('h')).toInt());  
Serial.print ("le nombre de boites du type G: " );Serial.println(G);
```

```
H =(inputString.substring(inputString.indexOf('h') + 1,inputString.indexOf('i')).toInt());  
Serial.print ("le nombre de boites du type H: " ); Serial.println(H);
```

```
I =(inputString.substring(inputString.indexOf('i') + 1,inputString.indexOf('j')).toInt());  
Serial.print ("le nombre de boites du casier I: " );Serial.println(I);
```

```
J =(inputString.substring(inputString.indexOf('j') + 1,inputString.indexOf('k')).toInt());  
Serial.print ("le nombre de boites du type J: " ); Serial.println(J);
```

```
K =(inputString.substring(inputString.indexOf('k') + 1,inputString.indexOf('l')).toInt());  
Serial.print ("le nombre de boites du type K: " ); Serial.println(K);
```

```
L =(inputString.substring(inputString.indexOf('l') + 1,inputString.indexOf('m')).toInt());  
Serial.print ("le nombre de boites du type L: " );Serial.println(L);
```

```
M=(inputString.substring(inputString.indexOf('m') + 1,inputString.indexOf('n')).toInt());
Serial.print ("le nombre de boites du type M: " ); Serial.println(M);
```

```
N=(inputString.substring(inputString.indexOf('n') + 1,inputString.indexOf('o')).toInt());
Serial.print ("le nombre de boites du type N: " );Serial.println(N);
```

```
O=(inputString.substring(inputString.indexOf('o') + 1,inputString.indexOf('p')).toInt());
Serial.print ("le nombre de boites du type O: " ); Serial.println(O);
```

```
P=(inputString.substring(inputString.indexOf('p') + 1,inputString.indexOf('q')).toInt());
Serial.print ("le nombre de boites du type P: " ); Serial.println(P);
delay(TIM);
```

```
    inputString = "";                stringComplete = false;
}
if (cmp1 < A) {
    pwm.setPWM(0, 0, pulseWidth(anglemin));
    delay(TIM);
    pwm.setPWM(0, 0, pulseWidth(anglemax));
    delay(TIM);
    cmp1++;}
if (cmp2 < B) {
    pwm.setPWM(1, 0, pulseWidth(anglemin));
    delay(TIM);
    pwm.setPWM(1, 0, pulseWidth(anglemax));
    delay(TIM);
    cmp2++;}
if (cmp3< C) {
    pwm.setPWM(2, 0, pulseWidth(anglemin));
    delay(TIM);
    pwm.setPWM(2, 0, pulseWidth(anglemax));
    delay(TIM);
    cmp3++;}
if (cmp4 < D) {
    pwm.setPWM(3, 0, pulseWidth(anglemin));
    delay(TIM);
```



```
pwm.setPWM(3, 0, pulseWidth(anglemax));
  delay(TIM);
  cmp4++;}
  if (cmp5 < E) {
pwm.setPWM(4, 0, pulseWidth(anglemin));
  delay(TIM);
  pwm.setPWM(4, 0, pulseWidth(anglemax));
  delay(TIM);
  cmp5++;}
  if (cmp6 < F) {
pwm.setPWM(5, 0, pulseWidth(anglemin));
  delay(TIM);
  pwm.setPWM(5, 0, pulseWidth(anglemax));
  delay(TIM);
  cmp6++;}
  if (cmp7 < G) {
pwm.setPWM(6, 0, pulseWidth(anglemin));
  delay(TIM);
  pwm.setPWM(6, 0, pulseWidth(anglemax));
  delay(TIM);
  cmp7++;}
  if (cmp8 < H) {
pwm.setPWM(7, 0, pulseWidth(anglemin));
  delay(TIM);
  pwm.setPWM(7, 0, pulseWidth(anglemax));
  delay(TIM);
  cmp8++;}
if (cmp9 < I) {
  pwm.setPWM(8, 0, pulseWidth(anglemin));
  delay(TIM);
  pwm.setPWM(8, 0, pulseWidth(anglemax));
  delay(TIM);
  cmp9++;}
  if (cmp10 < J) {
pwm.setPWM(9, 0, pulseWidth(anglemin));
  delay(TIM);
```

```
pwm.setPWM(9, 0, pulseWidth(anglemax));
  delay(TIM);
  cmp10++;}
  if (cmp11 < K)
pwm.setPWM(10, 0, pulseWidth(anglemin));
  delay(TIM);
  pwm.setPWM(10, 0, pulseWidth(anglemax));
  delay(TIM);
  cmp11++;}
  if (cmp12 < L) {
pwm.setPWM(11, 0, pulseWidth(anglemin));
  delay(TIM);
  pwm.setPWM(11, 0, pulseWidth(anglemax));
  delay(TIM);
  cmp12++;}
  if (cmp13 < M) {
pwm.setPWM(12, 0, pulseWidth(anglemin));
  delay(TIM);
  pwm.setPWM(12, 0, pulseWidth(anglemax));
  delay(TIM);
  cmp13++;}
  if (cmp14 < N) {
  pwm.setPWM(13, 0, pulseWidth(anglemin));
  delay(TIM);
  pwm.setPWM(13, 0, pulseWidth(anglemax));
  delay(TIM);
  cmp14++;}
  if (cmp15 < O) {
pwm.setPWM(14, 0, pulseWidth(anglemin));
  delay(TIM);
  pwm.setPWM(14, 0, pulseWidth(anglemax));
  delay(TIM);
  cmp15++;}
  if (cmp16 < P) {
pwm.setPWM(15, 0, pulseWidth(anglemin));
  delay(TIM);
```

```
pwm.setPWM(15, 0, pulseWidth(anglemax));
  delay(TIM);
  cmp16++;}
motor();
```

```
}
```

```
int pulseWidth(int angle)
{
  int pulse_wide, analog_value;
  pulse_wide = map(angle, 0, 180, MIN_PULSE_WIDTH, MAX_PULSE_WIDTH);
  analog_value = int(float(pulse_wide) / 1000000 * FREQUENCY * 4096);
  return analog_value;
}
```

```
void serialEvent() {
  while (Serial.available()) {
    char inChar = (char)Serial.read();
    if (inChar == '\n') {
      stringComplete = true;
    }
    else {
      inputString += inChar;
    }
  }
}
```

```
void capteur(){
  int detect = digitalRead(cap1);
  if(detect == LOW){ Serial.println("la présence des boîtes dans casier A"); }
  else{Serial.println("casier A est vide ");}

  int detectB = digitalRead(cap2);
  if(detectB == LOW){ Serial.println("la présence des boîtes dans casier B"); }
  else{Serial.println("casier B est vide ");}
```

```
int detectC = digitalRead(cap3);
if(detectC == LOW){ Serial.println("la présence des boîtes dans casier C"); }
else{ Serial.println("casier C est vide ");}
```

```
int detectD = digitalRead(cap4);
if(detectD == LOW){ Serial.println("la présence des boîtes dans casier D"); }
else{ Serial.println("casier D est vide ");}
```

```
int detectE = digitalRead(cap5);
if(detectE == LOW){ Serial.println("la présence des boîtes dans casier E"); }
else{ Serial.println("casier E est vide ");}
```

```
int detectF = digitalRead(cap6);// read obstacle status and store it into "detect"
if(detectF == LOW){ Serial.println("la présence des boîtes dans casier F"); }
else{ Serial.println("casier F est vide ");}
```

```
int detectG = digitalRead(cap7);// read obstacle status and store it into "detect"
if(detectG == LOW){ Serial.println("la présence des boîtes dans casier G"); }
else{ Serial.println("casier G est vide ");}
```

```
int detectH = digitalRead(cap8);// read obstacle status and store it into "detect"
if(detectH == LOW){ Serial.println("la présence des boîtes dans casier H"); }
else{ Serial.println("casier H est vide ");}
```

```
int detectI = digitalRead(cap9);// read obstacle status and store it into "detect"
if(detectI == LOW){ Serial.println("la présence des boîtes dans casier I"); }
else{ Serial.println("casier I est vide ");}
```

```
int detectJ = digitalRead(cap10);// read obstacle status and store it into "detect"
if(detectJ == LOW){ Serial.println("la présence des boîtes dans casier J"); }
else{ Serial.println("casier J est vide ");}
```

```
int detectK = digitalRead(cap11);// read obstacle status and store it into "detect"
if(detectK == LOW){ Serial.println("la présence des boîtes dans casier K"); }
else{ Serial.println("casier K est vide ");}
```

```
int detectL = digitalRead(cap12);// read obstacle status and store it into "detect"
if(detectL == LOW){ Serial.println("la présence des boîtes dans casier L"); }
else{ Serial.println("casier L est vide ");}
```

```
int detectM = digitalRead(cap13);
if(detectM == LOW){ Serial.println("la présence des boîtes dans casier M"); }
else{ Serial.println("casier M est vide ");}
```

```
int detectN = digitalRead(cap14);
if(detectN == LOW){ Serial.println("la présence des boîtes dans casier N"); }
else{ Serial.println("casier N est vide ");}
```

```
int detectO = digitalRead(cap15);
if(detectO == LOW){ Serial.println("la présence des boîtes dans casier O"); }
else{ Serial.println("casier O est vide ");}
```

```
int detectP = digitalRead(cap16);
if(detectP == LOW){ Serial.println("la présence des boîtes dans casier P"); }
else{ Serial.println("casier P est vide ");}
digitalWrite(moteur, HIGH);
delay(timecapt);
}
void motor(){
```

```
digitalWrite(moteur, HIGH);
delay(10000);
digitalWrite(moteur, LOW);
delay(100);
}
```

Organigramme :

