REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE

MINISTERE DE L'ENSEIGNEMENT SUPERIEUR ET DE LA RECHERCHE SCIENTIFIQUE

UNIVERSITE YAHIA FARES DE MEDEA



Faculté de **T**echnologie

Mémoire de fin d'études de Master

Filière : Electronique
Spécialité : Electronique des systèmes embarqués

Etude et réalisation d'une maison intelligente

Présenté par :

ZEMIR Zineddine

KHELIL CHERFI Tarek

Proposé et Dirigé par :

Mr. RAGGAI Djahid

Promotion 2019

REMERCIEMENT

Au début nous tenons à remercier le bon dieu pour le courage et la volonté et la force qu'il nous a donnés,
L'espoir et la santé qu'il nous a octroyé afin que soit réalisé ce
Modeste travail.
On remercie notre encadreur ***Mr. raggai***
Pour son soutien moral et pédagogique pour ses
Orientations bienveillantes.
Nous tenons à remercier
tous les enseignants qui nous ont suivi durant notre formation
et pour avoir accepté d'examiner notre travail
nos remerciement vont également à tous les
Membres de jury, qui font l'honneur de juger notre travail.
Rien n'aurait été possible sans le soutien de nos proches, merci encore à vous, rien n'a su nous motiver davantage que votre appui.

Grand Merci À TOUS...

Zineddine et Tarek

Dédicace

Je dédie ce modeste travail particulièrement :

*A **ma mère**, A celle qui a tant souffert, sans me faire souffrir, Qu'elle trouve dans ce mémoire le témoignage de ma reconnaissance et de mon affection pour tous les sacrifices, l'extrême amour et la bonté, Qu'elle M'a offert pour me voir Réussir ;*

*A **mon père**, A l'homme que je dois ma réussite, mon bonheur, et tout le respect ; Qu'il trouve ici l'expression de mon affection et une récompense des sacrifices*

Consentis pour moi.

A MES frère sans oublier ma sœur

*A toute la famille « **ZEMIR** »*

*A mon promoteur ***Mr. Raggai DJahid*** et mon binôme **Tarek***

qui a contribué à la réalisation de ce travail.

A Tous mes collègues sans exception …

Z. Zineddine

Dédicace

Je dédie ce travail

A ma maman

Qui m'a soutenu et encouragé durant toutes ces années d'études.

Qu'elle trouve le témoignage de ma profonde reconnaissance

A toi mon père

Tu as toujours été a mes cotés pour me soutenir et m'encourager.

Que ce travail traduit ma gratitude et mon affection

A mes braves frères et ma sœur qui ont été toujours présent pour moi

A mon grand père

A ma grand-mère

A mes oncles et mes tantes

A tous ceux qui m'ont aidé à faire ce travail

A tous mes chers amis : mon binôme ***Zinou****, mes amis et mes collègues d'etude*

A tous ceux que j'aime.

KH. Tarek

ملخص:

تعتبر المنازل الذكية قفزة نوعية في عصر التكنولوجيا الذي نعيشه حاليا لما تحتويه من اجهزة الكترونية متطورة.حيث تقوم فكرة المنزل الذكي ببساطة على إمكانية أداء أي مهمة دون تدخل مباشرمن صاحب المنزل ، وطالما هومتصل بشبكات داخلية مغلقة ومحمية جيدًا فان لديه اليد العليا للتحكم في كافة التفاصيل بداخل منزل،عن طريق الهاتف المحمول ومن مسافة تتجاوز الكيلومترات أو حتى في بلد آخر، فكرة المنزل الذكي متواجد منذ فترة طويلة، وكل ما نحتاج له هو توافر التكنولوجيا المطلوبة لتنفيذها فقط.

الكلمات المفتاحية: Domotique Arduino Uno, Arduino Mega, ESP8266, RFID

Résumé :

Les maisons intelligentes représentent un saut qualitatif dans la technologie de nos jours. L'idée est basée sur l'automatisation des fonctionnalités de la maison notamment à travers une application sur smartphone ou tablette rendant la commande possible même à des kilomètres. L'idée de la maison intelligente est ancienne, elle attendait juste l'avènement de la technologie permettant sa mise au point.

Mots clés : Domotique Arduino Uno, Arduino Mega, ESP8266, RFID

Abstract :

Smart homes are a qualitative leap in technology today. The idea is based on automating the features of the house through an application on smartphone or tablet making the order possible even for miles. The idea of the smart home is old, it was just waiting for the advent of technology for its development.

Keyword: Domotique Arduino Uno, Arduino Mega, ESP8266, RFID

Table des matières

INTRODUCTION GENERALE 1

CHAPITRE I : Les systèmes domotiques

I.1 Introduction.....2
I.2 Définition de la domotique 2
I.3 Le principe de la domotique 2
I.4 Domain de la domotique.....3
I.4.1 Le confort.....3
I.4.2 L'économie d'énergie 3
I.4.3 La sécurité.....4
I.5 La communication dans la domotique 4
I.5.1 Communication filaire 5
I.5.2 La communication sans fils 9
I.6 Conclusion 25

CHAPITRE II : Matériel et logiciel utilisé dans le projet

II.1 Introduction.....26
II.2 Matériel.....26
II.2.1 Carte Arduino Méga 2560 26
II.2.2 La carte Arduino Uno 28
II.2.3 Le module Wi-Fi (ESP8266-01).....31
II.2.4 Le module RFID 36
II.2.5 Afficheur LCD 16x2.....40
II.2.6 Le servo-moteur (Sg90) 42
II.2.7 Capteur de lumière (HM-Fliying fish).....44
II.2.8 Capteur de gaz (MQ-5) 45
II.2.9 Capteur de flamme (flying Fish).....47
II.2.10 Le capteur de température et humidité DHT-22 :.....49
II.2.11 Le capteur ultrasonic HC-SR04.....50
II.2.12 Le clavier 4x4 (keypad) 53

II.2.13 Capteur de mouvement (PIR) 55
II.2.14 Buzzer et Ventilateur 57
II.3 Logiciel 59
II.3.1 Arduino IDE.....59
II.3.2 Proteus professionnel 8 61
II.3.3 MIT App Inventor.....62
II.4 Conclusion 63

CHAPITRE III : La réalisation final du projet

III.1 Introduction.....64
III.1.1 Description de périphérique et leur position 64
III.1.2 La programmation.....66
III.1.3 Application sur android.....76
III.2 Aspect hardware de la réalisation. 78
III.2.1 Simulation par le logiciel Proteus (ISIS) 78
III.2.2 Assemblage du matériel sur ARES.....84
III.2.3 Assemblage du matériel sur la plaquette d'essaie.....87
III.2.4 Structure de la maison.....90
III.3 Le produit fini 91
III.3.1 La plaque électronique de le circuit imprimé 91
III.3.2 La maquette de la maison 93
III.4 Conclusion 94
CONCLUSION GENERALE…….…….……..…….....95

Liste des figures

Chapitre I

Figure I.1: les différents types de la paire torsadée 6
Figure I.2 : Le câble Ethernet (RJ45) 8
Figure I.3 : La prise RJ45 9
Figure I.4 : Logo de Bluetooth 9
Figure I.5 : logo de zigbee 10
Figure I.6 : logo de wifi 11
Figure I.7: la structure de Mode ad-hoc 13
Figure I.8 : le Mode Infrastructure 14
Figure I.9 : Logo de RFID 17
Figure I.10 : composition du système RFID.....18
Figure I.11: Interrogateurs RFID (Lecteur).....18
Figure I.12: Représentation de l'étiquette d'une RFID 19
Figure I.13 : Les caractéristiques des tags RFID 19
Figure I.14 : Fonctionnement du système RFID 21
Figure I.15 : Le couplage tag RFID / lecteur RFID.....22

Chapitre II

Figure II. 1: La carte Arduino Méga. 26
Figure II. 2: Explication de la carte Arduino Méga. 27
Figure II. 3: La carte Arduino Uno.....29
Figure II. 4: Explication de la carte Arduino Méga. 30
Figure II. 5: L'ESP8266-01.....31
Figure II. 6: La gamme ESP8266XY. 32
Figure II. 7: Diagramme de bloque de L'ESP8266.....33
Figure II. 8: Le module RC522 37
Figure II. 9: Le brochage de RFID-RC522 38
Figure II. 10 : Porte-clés de RFID.....39
Figure II. 11: badge de RFID 39
Figure II. 12: Afficheur LCD 16x2 40
Figure II. 13: L'Afficheur LCD avec l'adaptateur I2C 42
Figure II. 14: Adaptation I2C.....42
Figure II. 15: Le servo-moteur (Sg90) 43
Figure II. 16: Capteur de lumière 44
Figure II. 17 : Explication du module 45
Figure II. 18: Capteur de gaz (MQ-5) 46
Figure II. 19: Explication du module MQ5. 47
Figure II. 20: Capteur de flamme 47

Figure II. 21: Explication du module de flamme 48
Figure II. 22: Le capteur DHT-22 49
Figure II. 23: Le chronogramme de HC-SR04 52
Figure II. 24: Le clavier 4x4.....53
Figure II. 25: Fonctionnement du clavier 4x4 55
Figure II. 26: Capteur de mouvement (PIR).....55
Figure II. 27: Fonctionnement de l'infrarouge. 56
Figure II. 28: Réglage de sensibilité.....57
Figure II. 29: Réglage du temp.....57
Figure II. 30: le buzzer 58
Figure II. 31: La ventilateur 12v. 59
Figure II. 32: L'interface du logiciel IDE. 60
Figure II. 33: La barre des tâches. 60
Figure II. 34: L'interface du ISIS. 61
Figure II. 35: L'interface du ARES. 62
Figure II. 36: L'interface du MIT. 63

Chapitre III

Figure III. 1: L'organigramme de capteur de mouvement (PIR) 66
Figure III. 2: L'organigramme de système d'ouverture et fermeture de la porte. 67
Figure III. 3: L'organigramme du capteur ultrasonique HC-SR04. 68
Figure III. 4: L'organigramme de capteur DHT. 69
Figure III. 5: L'organigramme de capteur de gaz MQ5 70
Figure III. 6: L'organigramme de capteur de flamme.....71
Figure III. 7: L'organigramme de module RFID. 72
Figure III. 8: L'organigramme du capteur de la lumière. 73
Figure III. 9: L'organigramme de module ESP8266-01. 74
Figure III. 10: L'organigramme general. 75
Figure III. 11: L'interface de l'application Android. 76
Figure III. 12: les bouton ON/OFF de la chambre 1. 76
Figure III. 13: les bouton ON/OFF de la chambre 2. 77
Figure III. 14: Les bouton ON/OFF du salon 3.....77
Figure III. 15: les boutons ON/OFF de la cuisine. 77
Figure III. 16: Le schéma général de la simulation. 78
Figure III. 17: système d'ouverture et fermeture de la porte. 79
Figure III. 18: système de capteur ultrasonique HC-SR04. 79
Figure III. 19: système de capteur de gaz MQ5. 80
Figure III. 20: système de capteur de flamme. 80
Figure III. 21: Système de capteur de mouvement (PIR). 81
Figure III. 22: Systeme RFID. 81
Figure III. 23: Système de capteur DHT22 avec un afficheur LCD. 82
Figure III. 24: Système de capteur de Lumière. 82
Figure III. 25: Le système de module ESP8266-01 avec Arduino Uno. 83

Figure III. 26: schéma de routage ARES 84
Figure III. 27: schéma de routage PDF. 85
Figure III. 28: Le typon. 86
Figure III. 29: Adaptateur I2c avec LCD 16x2. 87
Figure III. 30: Assemblage sur la plaquette d'essaie 88
Figure III. 31: Assemblage finale sur la plaquette. 89
Figure III. 32: le plan de la maison 90
Figure III. 33: Block A de la carte. 91
Figure III. 34: Block B de la carte.....91
Figure III. 35: l'arrière-plan de la carte.....92
Figure III. 36 : circuit final. 93
Figure III. 37: la maquette de la realisation 93
Figure III. 38: la façade de la maquette.....94

Liste des tableaux

Tableau II.1 : Caractéristiques de l'Arduino.....27
Tableau II. 2 : Caractéristiques de l'Arduino Uno.....30
Tableau II. 3 : Spécifications de L'ESP8266-01.....34
Tableau II. 4 : explication des pins de LCD 16x2.....41

Liste des abbreviation

AES	**A**dvanced **E**ncryption **S**tandard.
AP	**A**ccess **P**oint
AT	**A**ttention.
ASCII	**A**merican **S**tandard **C**ode for **I**nformation **I**nterchange.
CPL	**C**ourants **P**orteurs en **L**igne.
CIE	**C**leveland **I**nstitute of **E**lectronics
COM	**C**ommunications.
DIP	**D**ual **I**n-line **P**ackage.
DEL	**D**elete.
DHT	**D**i**H**ydro**T**estosterone
DOUT	**D**igital **O**utput.
AOUT	**A**nalog **O**utput
EIA	**E**lectronic **I**ndustries **A**lliance
ESP	**E**spressif **S**ystems.
FTP	**F**oiled **T**wisted **P**air.
FTDI	**F**uture **T**echnology **D**evices **I**nternational.
GPIO	**G**eneral **P**urpose **I**nput **O**utput
GND	**G**round
GPL	**G**az de **P**étrole **L**iquéfié.
HF	**H**aute **F**réquence.
ISO	**I**nternational **S**tandards **O**rganization.
IEEE	**I**nstitute of **E**lectrical and **E**lectronics **E**ngineer.
ISO	**I**nternational **O**rganization for **S**tandardization.
I2S	**I**ntelligent **S**ystems **S**olutions.
I2C	**I**nter **I**ntegrated **C**ircuit.
ICSP	**I**n **C**ircuit **S**erial **P**rogramming.
IP	**I**nternet **P**rotocol.

IRQ	**I**nterrupt **R**equest.
IDE	**I**ntegrated **D**evelopment **E**nvironment
IRDA	**I**nsurance **R**egulatory and **D**evelopment **A**uthority.
LCD	**L**iquid **C**rystal **D**isplay.
LAN	**L**ocal **A**rea **N**etwork.
LF	**B**asse **F**réquence.
LNA	**L**ow **N**oise **A**mplifier.
LDR	**L**ight **D**etect **R**esistor.
MIT	**M**assachusetts **I**nstitute of **T**echnology
MAC	**Media A**ccess **C**ontrol.
MAN	**M**etropolitan **A**rea **N**etwork.
MOSI	**M**aster **O**ut **S**lave **I**n.
MISO	**M**aster **I**n **S**lave **O**ut.
NXP	**N**ext **E**xperience.
NPN	**N**egative **P**ositive **N**egative **T**ransistor.
OSI	**O**pen **S**ystems **I**nterconnection.
PCB	**P**rinted **C**ircuit **B**oards.
PIR	**P**assive **I**nfrared **S**ensor.
PHY	**P**hysical **L**ayer **D**evice.
PWM	**P**ulse **W**idth **M**odulation.
P2P	**P**eer to **P**eer.
RST	**R**eset
RJ45	**R**egistered **J**ack – 45.
RFID	**R**adio **F**requency **I**dentification.
RX	**R**eception.
SFTP	**S**hielded and **F**oiled **T**wisted **P**air, nouvelle denomination SF/UTP.
SSTP	**S**hielded and **S**hielded **T**wisted **P**air, nouvelle denomination S/FTP.
STP	**S**hielded **T**wisted **P**air.
SHF	**S**uper **H**autes **F**réquences.
SPI	**S**erial **P**eripheral **I**nterface.

SOC	**S**ystem **O**n **C**hip..
SCL	**S**erial **C**lock **L**ine.
SCK	**S**erial **C**lock.
SS	**S**lave **S**elect.
STA	**S**elect **T**ype of **A**ddress
TKIP	**T**emporary **K**ey Integrity **P**rotocol.
TIA	**T**elecommunications **I**ndustry **A**ssociation.
TTL	**A**ransistor **T**ransistor **L**ogic
TX	**T**ransaction.
TWI	**T**wo **W**ire **I**nterface.
TCP	**T**ransmission **C**ontrol **P**rotocol
USB	**U**niversal **S**erial **B**us.
UTP	**U**nshielded **T**wisted **P**air.
UHF	**U**ltra **H**autes **F**réquences
UID	**U**ser **I**dentification.
UART	**U**niversal **A**synchronous **R**eceiver-**T**ransmitter.
VCC	**V**oltage at the **C**ommon **C**ollector
Wi-Fi	**W**ireless **F**idelity.
WPANs	**W**ireless **P**ersonal **A**rea **N**etworks.
WiGig	**W**ireless **G**igabit **A**lliance.
WEP	**W**ired **E**quivalent **P**rivacy.
WPA	**W**i-Fi **P**rotected **A**ccess.

Introduction générale

Introduction Générale

Durant ces dernières années, la technologie a connu une grande évolution dans tous les domaines, un des buts recherchés étant de faciliter l'usage et la communication entre l'être humain et les objets dont il se sert au quotidien.

Dans la vie quotidienne, notamment dans le travail, l'être humain fait de grands efforts. Pour que ces efforts suivent une cadence régulière sans baisse de rythme, il doit disposer d'un cadre de vie extraprofessionnelle bénéficiant d'un maximum de confort et de sécurité. Ce cadre de vie recherché devient une nécessité vitale pour certaines catégories vulnérables telles que les handicapés et les personnes âgées. Pour cela, la domotique offre des solutions qui répondent à ce besoin. Les possibilités sont énormes : de l'ouverture automatique du garage à la commande automatique des lumières en passant par la commande des rideaux des fenêtres, le déclenchement de la climatisation et du chauffage ...etc.

Notre projet consiste à réaliser une maison intelligente, comportant un certain nombre de fonctionnalités visant l'optimisation du confort dans l'habitat et la gestion de l'énergie. Ce mémoire sera organisé en trois chapitres :

Le premier chapitre aborde d'une manière générale les systèmes domotiques, leur principe, leurs domaines d'utilisation ainsi que les différentes technologies de communication et leur application dans la domotique.

Dans le deuxième chapitre, l'accent sera mis sur tous les éléments matériels et logiciels ayant relation avec notre réalisation. L'aspect fonctionnel de chaque élément sera donc détaillé pour rendre le chapitre réalisation le plus claire possible.

Le dernier chapitre consiste à exposer les détails de réalisation d'une maison intelligente à base d'Arduino Méga et Uno, et d'une application Android. Tous les aspects logiciels et matériels de la réalisation seront détaillés.


Chapitre I
les systémes domotiques

I.1 Introduction

Dans nos jours, le temps passe très vite à cause des nombreuses charges quotidiennes, qui ont donné de temps à autre l'occasion de gaspiller l'énergie et le temps, et parfois même de tomber dans des dangers qui sont mortels.

La maison intelligente va permettre de régler plusieurs problèmes en offrant la sécurité et le confort pour l'être humaine, qui va avoir l'accès de contrôler sa maison à l'aide d'une application d'où il veut. Dans ce chapitre, on va concentrer sur les domaines et les options de la domotique ainsi que les différentes technologies de communication.

I.2 Définition de la domotique

La domotique est une spécialité du bâtiment regroupant les techniques permettant de contrôler, d'automatiser et de programmer l'habitat. C'est une discipline récente issue de l'application à l'habitat de la programmation informatique.

Le mot "domotique" est utilisé depuis peu dans le langage et est une contraction des mots domus (domicile en latin) et du suffixe "tique" associé à la technique. Les tâches associées à la domotique dans la maison concernent des appareils techniques alimentés électriquement :

- La programmation des appareils électriques (éclairage, chauffage, audio-visuel, électro-ménager...)
- Le contrôle (alarme, contrôle d'accès, température, éclairage, incendie...). [1]

I.3 Le principe de la domotique

Le principe de la domotique est de programmer et contrôler à distance ou localement le comportement d'appareils que l'on aura intégrés dans un réseau. Le réseau qui peut être câblé ou sans fil est destiné à recevoir et émettre des informations entre les unités de commande et les appareils commandés.L'information circule dans les deux sens sur le réseau :

- Une unité de commande envoie des informations aux récepteurs chargés de faire effectuer une tâche précise à des appareils qui eux-mêmes envoient vers la ou les unités de commande des informations concernant leur état.

- Les informations échangées entre les différents appareils sur un réseau domotique communiquent en respectant un langage appelé protocole.

Certains réseaux domotiques nécessitent des interfaces appelées passerelles chargées de faire communiquer des appareils qui communiquent via des protocoles différents (ce sont en quelque sorte des traducteurs). [2]

I.4 Domain de la domotique

I.4.1 Le confort

Bien sûr, le fait d'automatiser sa maison a un véritable apport sur le confort qu'on y trouve. Plus besoin de se prendre une averse pour ouvrir le portail en rentrant à la maison, plus besoin de prendre froid en ouvrant les volets le matin, et fini les retours de weekend dans une maison toute froide.[3]

Aujourd'hui, une maison intelligence est capable de savoir quand vous rentrez à la maison (grâce à votre smartphone par exemple), et donc d'ouvrir le portail avant même que vous n'arriviez. Les volets peuvent s'ouvrir et se fermer au rythme du soleil, et peuvent même aller jusqu'à s'adapter à la saison et la température pour laisser entrer la lumière et la chaleur du soleil l'hiver, ou au contraire conserver le frais l'été en fermant les volets des fenêtres exposées au soleil. De la même façon, votre maison sait quand vous êtes présent, et peut ainsi adapter elle-même votre chauffage pour que la maison soit toujours à la température idéale pour vous. Il est même possible de diffuser automatiquement votre playlist musicale préférée à votre réveil, ou quand vous rentrez à la maison. Pendant ce temps**,** un robot peut passer l'aspirateur dans la maison à votre place, et le système d'arrosage automatique arrosera votre jardin, tout en tenant compte des prévisions météo des prochains jours, histoire de ne pas arroser inutilement.[3]

I.4.2 L'économie d'énergie

Dans l'air du temps, les économies d'énergies sont aussi considérées en domotique, la solution pour les économies d'énergie est de minimiser le gaspillage. Il s'agit en fait de supprimer une énergie dépensée alors qu'elle n'est pas utilisée. Pour y arriver, disposer de l'information sur la consommation : on peut se trouver dans une pièce et voir les lampes allumées dans les autres sur une simple LED. De ce fait, on pourra piloter l'énergie pour qu'elle soit attribuée au besoin (à la demande). Mais certaines actions peuvent être gérées automatiquement par le système en place, par exemple, on peut :

- Ne chauffer que les pièces où les personnes se trouvent.

- Gérer efficacement l'éclairages en mettant éteignant les lumières non utilisées.
- Utiliser des lampes économiques.[4]

I.4.3 La sécurité

La sécurité est aussi un point recherché. Quand on quitte son domicile on souhaite qu'elle reste surveillée. On cherche à éviter les intrusions, les tentatives de violations, intempéries (même légères). On a également d'autres types problèmes : incendie, inondation, fuite de gaz…

La domotique peut proposer un panel très large de détecteurs et de capteurs

- Des mouvements.
- Des inondations.
- Le bris de verre.
- Les vibrations.
- L'ouverture de porte et fenêtre.
- La fumée / feu.
- L'humidité / inondation.
- La pluie, le vent, la tempête.

Une autre forme de sécurité de la maison est le fait d'être capable d'agir sur sa maison à distance. Il va ainsi être possible de bloquer ou libérer les serrures à distance. Ici encore, on peut être alerté en cas de détection de problèmes.[4]

I.5 La communication dans la domotique

La communication tient une place de plus en plus importante dans le logement. Une installation domotique adaptée, avec les appareils de la maison montés en réseau, satisfait aux besoins et aux loisirs de chaque personne du foyer. Les échanges de données d'une installation domotique peuvent se faire via un réseau filaire ou sans fil. Le choix dépend avant tout des fonctions que l'on veut installer sur sa box domotique et des facilités ou contraintes de mise en œuvre par rapport au chantier : construction neuve, rénovation lourde ou légère.

Rendre son habitat communicant, donc domotique, se fait de 3 manières :

- Avec un câblage domotique bien pensé (communication filaire).
- Par domotique CPL ou courant porteur (appelé X10).

- En domotique sans fil (Wi-Fi, ondes radio, Bluetooth…).[5]

I.5.1 Communication filaire

I.5.1.1 Pair torsadé

Un câble paires torsadées décrit un modèle de câblage où une ligne de transmission est formée de deux conducteurs enroulés en hélice l'un autour de l'autre, cette configuration a pour but de maintenir précisément la distance entre les fils et de diminuer la diaphonie.
Le maintien de la distance entre fils de pair permet de définir une impédance caractéristique de la paire, afin de supprimer les réflexions de signaux aux raccords et en bout de ligne. [6]

Les contraintes géométriques (épaisseur de l'isolant/diamètre du fil) maintiennent cette impédance autour de 100 Ohm :

- 100 ohm pour les réseaux Ethernet en étoile.
- 150 ou bien 105 Ohm pour les réseaux Token ring.
- 100 ou bien 120 Ohm pour les réseaux de téléphonie.
- 90 Ohm pour les câbles USB.

Plus le nombre de torsades est important, plus la diaphonie est réduite. Le nombre de torsades moyen par mètre fait partie de la spécification du câble, mais chaque paire d'un câble est torsadée de manière légèrement différente pour éviter la diaphonie.[6]

L'utilisation de la signalisation différentielle symétrique permet de réduire davantage les interférences.

- Les types de blindage :

Les paires torsadées sont souvent blindées afin de limiter les interférences. Comme le blindage est fait de métal, celui-ci constitue également un référentiel de masse. Le blindage peut être appliqué individuellement aux paires ou à l'ensemble formé par celles-ci. Lorsque le blindage est appliqué à l'ensemble des paires, on parle d'écrantage.[6]

Il existe cinq types de paires torsadées :

- **Paire torsadée non blindée :** (Unshielded Twisted pair (UTP))

La paire torsadée non blindée n'est pas entourée d'un blindage protecteur. C'est le type de câble souvent utilisé pour le téléphone et certains réseaux informatiques.

- **Paire torsadée écrantée :** (Foiled twisted pair (FTP))

Les paires torsadées ont un blindage général assuré par une feuille d'aluminium. L'écran est disposé entre la gaine extérieure et les 4 paires torsadées. Elle est utilisée pour le téléphone et les réseaux informatiques.

- **Paire torsadée écrantée et blindée :** (Shielded and Foiled Twisted pair (SFTP) – nouvelle denomination SF/UTP.)

Cable dote d'un double écran commun à l'ensemble des paire (feuille métallisé et tresse)

- **Paire torsadée blindée et générale blindée :** (Shielded and Shielded Twisted Pair (SSTP) – nouvelle denomination S/FTP.) [6]

Cable STP dote en plus d'u écran commun entre la gaine extérieure et les 4 paire

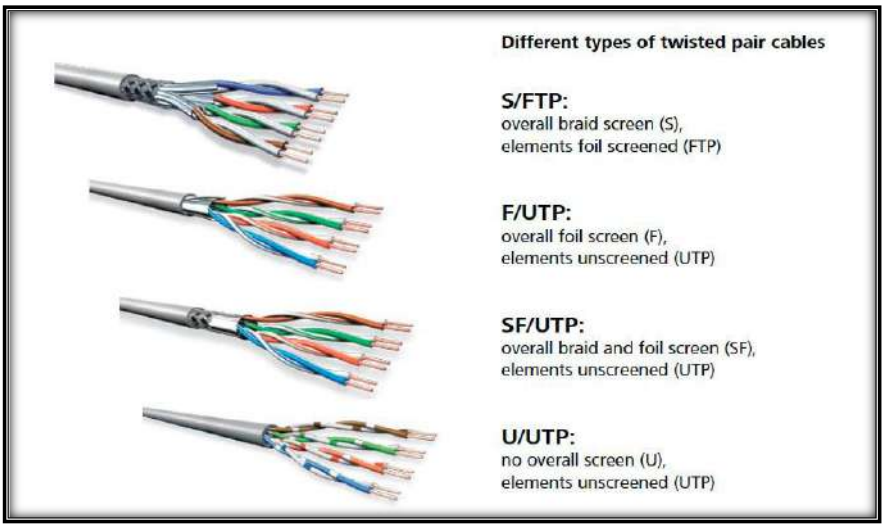


Figure I.1: *les différents types de la paire torsadée*

I.5.1.2 Cable Ethernet

Ethernet est un protocole de réseau informatique à commutation de paquets implémentant la couche physique et la sous-couche Medium Access Control du modèle OSI mais le protocole Ethernet est classé dans la couche de liaison, car les formats de trames que le standard supporte est normalisé et peut être encapsulé aussi dans d'autres protocoles que les couches physiques MAC et PHY de l'Ethernet, ces couches physiques faisant l'objet de normes séparées en fonction des débits, du support de transmission, longueur et conditions environnementales.[7]

C'est au départ une technologie de réseau local permettant que toutes les machines d'un réseau soient connectées à une même ligne de communication, formée de câbles cylindriques (câble coaxial, paires torsadées). Le standard qui a été le plus utilisé dans les années 1990 et qui l'est toujours est le 802.3 de l'IEEE (maintenant aussi adopté comme norme internationale ISO/CIE 8802-3). Ce dernier a largement remplacé d'autres standards comme le Token Ring et l'ARCNET.

Il existe aujourd'hui une grande variété d'Ethernet allant de l'Ethernet 10 mégabits/s à l'Ethernet 10 gigabits/s. À l'échelle internationale.[7]

Tableau I. 1 : *les variétés d'Ethernet. [8]*

Nom usuel	Dénomination	Standard	Support	Vitesse / Débit	Longueur
Ethernet	10Base-T	IEEE 802.3	Cuivre	10 Mbps	100 m
Fast Ethernet	100Base-TX	IEEE 802.3u	Cuivre	100 Mbps	100 m
Gigabit Ethernet	1000Base-T	IEEE 802.3ab	Cuivre	1000 Mbps	100 m
10 Gigabit Ethernet	10GBase-T	IEEE 802.3an	Cuivre	10 Gbps	100 m

- Câble et prise RJ45

Le câblage d'un réseau RJ 45 est réalisé à partir de câbles nommés paires torsadées. Le câblage consiste en quatre paires de fils : marron, orange, vert et bleu. Pour chaque paire, un fil est de couleur unie et l'autre rayé de cette même couleur. [9]

Il existe deux standards de câblage définis par le Télécommunication International Association / Electronic Industry Association : TIA/EIA 568 A (obsolète) et TIA/EIA 568 B, qui diffèrent par la position des paires vertes et oranges. La même norme doit être utilisée pour toute

la maison. Pour un câble droit, l'ordre des brins doit être le même des deux côtés du câble. C'est le cas des deux standards TIA/EIA 568 A et TIA/EIA 568 B. Pour un câble croisé, certains brins seront inversés : les paires qui assurent l'émission d'un côté seront en réception de l'autre côté. [9]

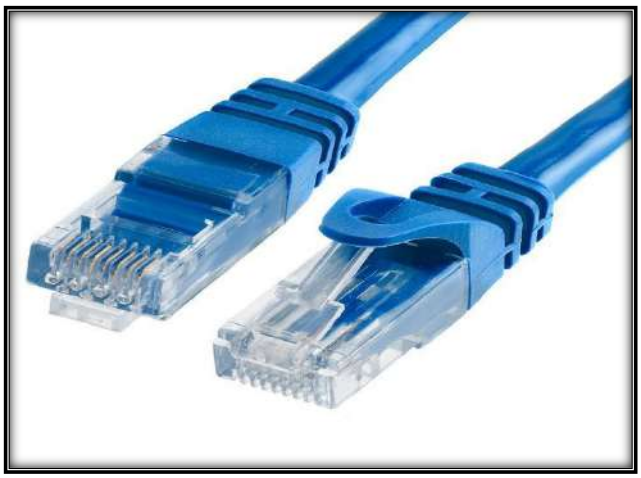


Figure I.2 : *Le câble Ethernet (RJ45)*

La plupart du temps, les câbles sont utilisés comme ceci : deux paires pour internet, une paire pour la télévision et une paire pour le téléphone, deux types de câbles sont disponibles sur le marché. [9]

- **Le câble monobrin :** est constitué d'un seul fil de cuivre épais. Plus rigide que le multibrin, il transporte le signal sur une plus longue distance.
- **Le câble multibrin :** est moins coûteux, plus facile d'emploi mais n'est pas adapté pour relier deux appareils espacés par une grande distance.

Le câble catégorie 5 est le plus utilisé pour le câblage de réseaux Ethernet (RJ 45) domestiques. Il correspond à des fréquences jusqu'à 100 MHz et des débits jusqu'à 100 Mbits/s, la catégorie 6 véhicule un plus grand débit et coûte par conséquent plus cher.Une prise RJ45, également appelée port Ethernet ou connecteur RJ45 se présente sous la forme d'un connecteur avec huit fils sertis dans une broche plastique à l'extrémité d'un câble ou d'une prise murale. Ce type de prises est utilisé pour

les communications et transporte des courants faibles. Il remplace progressivement les anciennes prises en T qui étaient utilisées pour le téléphone.[9]



Figure I.3 : *La prise RJ45*

I.5.2 La communication sans fils

I.5.2.1 Bluetooth

C'est en 1994 qu'Ericsson, fabricant suédois de téléphone mobile, a mis au point la technologie Bluetooth. Cette dernière permet tout simplement de relier des appareils électroniques entre eux sans aucune liaison par fils. La puissance d'émission du Bluetooth est relativement faible et son débit limité, car cette technologie est utilisée sur de petites distances, dans une même maison par exemple. En effet, le Bluetooth est fait pour connecter des appareils assez proches les uns des autres (clavier et ordinateur, enceinte et smartphone, etc), même si les progrès vont de plus en plus vite, permettant de relier des appareils de plus en plus éloignés. Il existe trois classes de Bluetooth, permettant de transmettre des données jusqu'à 10 mètres de portée (classe III), 20 mètres (classe II) ou 100 mètres (classe I).Les périphériques grands publics font partie de la classe III pour la plupart. [10]

Figure I.4 : *Logo de Bluetooth*

❖ Un peu plus de détails sur le fonctionnement du Bluetooth

Le Bluetooth fonctionne grâce à des ondes radio, sur une bande de fréquence de 2,402 GHz à 2,480 GHz, découpée en 79 canaux d'une largeur totale de 1 MHz. Le Bluetooth ne brouille pas les signaux des autres appareils car il utilise un signal très faible et change de canal jusqu'à 1600 fois par seconde. Le réseau formé par la connexion de plusieurs périphériques Bluetooth est appelé picoréseau (ou piconet en anglais). Pour que deux appareils puissent communiquer, il faut passer par ce que l'on appelle l'appairage, qui permet par exemple à un smartphone et à une enceinte de se « rencontrer ». Une fois cette connexion établie, l'un des appareils prend le contrôle sur les autres périphériques du picoréseau. Au maximum, huit appareils peuvent communiquer simultanément entre eux. Le Bluetooth n'a cessé d'évoluer (version 1,2,3,4 !) et de s'améliorer. Le Bluetooth 5.0 arrive en 2017, bien plus rapide, avec plus de portées, plus de possibilités, ouvrant de nouvelles portes à la connexion sans fil ! [10]

I.5.2.2 Zigbee

ZigBee est un protocole de haut niveau permettant la communication de petites radios, à consommation réduite, basée sur la norme IEEE 802.15.4 pour les réseaux à dimension personnelle (Wireless Personal Area Networks : WPANs). L'essence même d'un protocole est de donner un cadre de fonctionnement à une communication. Ainsi, l'IEEE définit dans le cadre de sa norme, le cadre de ce protocole pour les couches basses (physique et mac).[11]

Cette technologie a pour but la communication de courte distance telle que le propose déjà la technologie Bluetooth, tout en étant moins chère et plus simple. À titre d'exemple, les nœuds ZigBee classiques nécessitent environ 10 % du code nécessaire à la mise en œuvre de nœuds Bluetooth ou de réseaux sans fil, et les nœuds ZigBee les plus élémentaires peuvent ainsi descendre jusqu'à 2 % ! .[11]



Figure I.5 : *logo de zigbee*

❖ Un peu plus de détails sur le fonctionnement du zigbee

La spécification initiale de ZigBee propose un protocole lent dont le rayon d'action est relativement faible, mais dont la fiabilité est assez élevée, le prix de revient faible et la consommation considérablement réduite. On retrouve donc ce protocole dans des «environnements embarqués » où la consommation est un critère de sélection. Ainsi, la domotique et les nombreux capteurs et télécommandes qu'elle implémente apprécie particulièrement ce protocole en plein essor et dont la configuration du réseau maillée se fait automatiquement en fonction de l'ajout ou de la suppression de nœuds. On retrouve aussi ZigBee dans les contrôles industriels et les applications médicales. Les nœuds sont conçus pour fonctionner plusieurs mois (jusqu'à dix ans pour les moins consommant) en autonomie complète grâce à une simple pile alcaline de 1,5 V. [11]

Il est donc nécessaire d'implémenter les couches de plus haut niveau, afin que ce modèle soit parfaitement fonctionnel. Les protocoles ZigBee peuvent donc fonctionner, en théorie, sur plusieurs supports mac mais sont en général présentes pour le média 802.15.4. C'est la ZigBee Alliance qui s'occupe de cette partie du protocole en fournissant un stack (ou pile) de référence. Celle-ci est réservée aux membres de l'alliance qui doivent l'implémenter dans leurs solutions.[11]

I.5.2.3 La technologie Wi-Fi

a) Définition

Le Wi-Fi, aussi orthographié wifi est un ensemble de protocoles de communication sans fil régis par les normes du groupe IEEE 802.11 (ISO/CEI 8802-11). Un réseau Wi-Fi permet de relier par ondes radio plusieurs appareil informatiques (ordinateur, routeur, smartphone, décodeur, etc.) au sein d'un réseau informatique afin de permettre la transmission de données entre eux.[12]



Figure I.6 : *logo de wifi*

b) IEEE 802.11

Est un ensemble de normes concernant les réseaux sans fil locaux (le Wi-Fi) qui ont été mises au point par le groupe de travail 11 du comité de normalisation LAN/MAN de l'IEEE (IEEE 802).[13]

c) Les normes de IEEE 802.11

Avec un débit maximal de 54 Mbit/s les normes Wifi 802.11a/b et g sont un peu dépassées aujourd'hui. Ce sont pourtant des normes qui ont été très utiles durant de longues années lors des balbutiements du Wifi.Voyons comment s'en sortent ces normes Wifi aujourd'hui :

- **La norme 802.11a :** Avec une vitesse maximale de 54 Mbit/s cette norme fait partie des plus rapides à l'époque. Cette vitesse est due à l'utilisation de la bande 5 Ghz qui permet

un bon transfert des données. Cette fréquence limite aussi la portée du signal qui sera de 35m. Cette fréquence est moins congestionnée et permet d'avoir une connexion plus stable et avec moins d'interférences.[14]

- **La norme 802.11b** : c'est un débit inférieur qu'on retrouve avec cette norme, en effet le 802.11b est limité à 11 Mbit/s. Cette bande passante limitée est due à l'utilisation de la fréquence 2.4 Ghz. La portée est encore une fois limitée à 35 m.[14]
- **La norme 802.11g** : Datant de 2003 c'est la plus récente des 3, elle permet d'atteindre un débit de 54 Mbit/s en utilisant la fréquence 2.4 Ghz. Les concepteurs de cette nouvelle norme ont réussi à étendre la portée du signal jusqu'à 40 m environ. En 2003 c'est la meilleure norme Wifi existante à la fois stable, rapide et avec la meilleure portée du signal jamais observée.[14]
- **La norme 802.11n :** Avec l'apparition de la fibre optique à la fin des années 2000, les constructeurs ont dû trouver une nouvelle norme pour garantir le meilleur débit possible aux particuliers. C'est à ce moment (en 2009) que la norme wifi 802.11n apparaît. Cette norme a révolutionné le wifi en apportant un débit incroyable allant jusqu'à 288 Mbit/s sur la fréquence 2.4 Ghz et jusqu'à 600 Mbit/s sur la fréquence 5 GHz, ailleurs la portée du signal a aussi été améliorée, en effet sur la fréquence 2.4 Ghz la portée est maintenant de 70 m soit le double de ce qu'on faisait avant.[14]
- **IEEE 802.11ac** : fonctionne dans la bande de fréquence de 5 GHz offrant des débits de données de 450 Mbit/s à 1,3 Gbit/s (1 300 Mbit/s). Cette norme est rétro compatible avec les périphériques 802.11a/n. [15]

- **IEEE 802.11ad** : également appelée « WiGig ». Cette norme utilise une solution Wi-Fi tri-bande sur les bandes de fréquences de 2,4 GHz, 5 GHz et 60 GHz et permet des débits théoriques jusqu'à 7 Gbit/s. [15]

Norme	Débit maximal	Fréquence	Rétrocompatibilité
802.11a	54 Mbit/s	5 GHz	Non
802.11b	11 Mbit/s	2,4 GHz	Non
802.11g	54 Mbit/s	2,4 GHz	802.11b
802.11n	600 Mbit/s	2,4 GHz ou 5 GHz	802.11a/b/g
802.11ac	1,3 Gbit/s (1 300 Mbit/s)	2,4 GHz et 5 GHz	802.11a/n
802.11ad	7 Gbit/s (7 000 Mbit/s)	2,4 Ghz, 5 Ghz et 60 Ghz	802.11a/b/g/n/ac

Tableau I.2: *Classification des normes IEEE 802.11 [15]*

d) Modes de fonctionnement

Un réseau Wifi peut fonctionner de deux manières différentes

- **Le mode Ad-Hoc**

C'est un mode de fonctionnement qui permet de connecter directement les ordinateurs équipés d'une carte réseau Wi-Fi, sans utiliser un matériel tiers. Ce mode est idéal pour interconnecter rapidement des machines entre elles sans matériel supplémentaire. Avec ce mode, vous n'avez pas besoin de point d'accès pour gérer le réseau. Le problème dans ce type de réseau, c'est que d'une part, la bande passante du réseau est basée sur la vitesse de l'hôte le plus lent et que d'autre part, la bande passante du réseau est divisée par le nombre d'hôtes sur ce réseau, ce mode peut être utilisé dans une maison pour un réseau simple.[16]

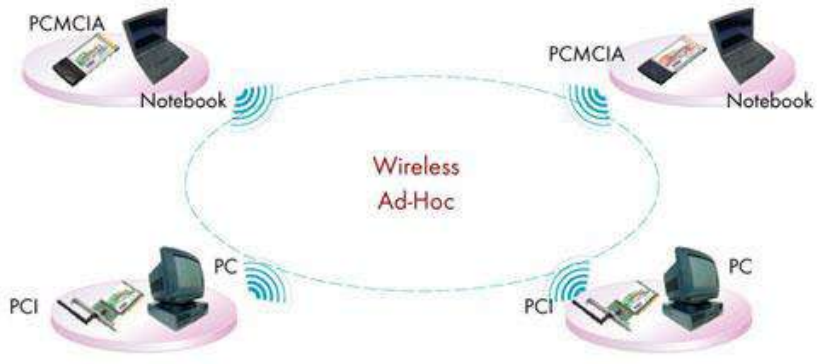


Figure I.7: *la structure de Mode ad-hoc*

❖ **Le mode Infrastructure**

C'est un mode de fonctionnement qui permet de connecter les ordinateurs équipés d'une carte réseau WiFi entre eux via un ou plusieurs points d'accès qui agissent comme des concentrateurs. Il est essentiellement utilisé en entreprise. Avec ce mode, tout est géré par un point d'accès, les données qu'un hôte émet lui sont transmises et lui seul les renvoie aux autres membres du réseau. Ainsi, la bande passante est économisée. De plus, plusieurs points d'accès peuvent être reliés ensemble pour augmenter la portée du réseau Wifi. Ce mode est le plus utilisé car il est bien plus fiable.[16]



Figure I.8 : *le Mode Infrastructure*

e) Sécurité de wifi

• **Le WEP :** est un protocole de sécurité, spécifié dans la norme IEEE 802.11b, conçu pour fournir un réseau local sans fil (WLAN) avec un niveau de sécurité et de confidentialité. Utilisant l'algorithme symétrique RC4 avec des clés d'une longueur de 64 ou 128 bits. Le principe du WEP consiste à définir dans un premier temps une clé secrète de 40 ou 128 bits. Cette clé secrète doit être déclarée au niveau du point d'accès et des clients. La clé sert à créer un nombre pseudo-aléatoire d'une longueur égale à la longueur de la trame. Ce mode de sécurité est compatible avec la norme wifi a/b/g.[17]

• **Le WPA** est une amélioration de l'algorithme WEP et de l'authentification des réseaux 802.11. Développé par l'IEEE pour combler les faiblesses du WEP, le WPA offre une sécurité nettement supérieure par rapport au WEP. Le WPA reposant sur des protocoles d'authentification

et un algorithme de cryptage robuste : TKIP (Temporary Key Integrity Protocol). Le protocole TKIP permet la génération aléatoire de clés et offre la possibilité de modifier la clé de chiffrement plusieurs fois par secondes, pour plus de sécurité.[17]

Le fonctionnement de WPA repose sur la mise en œuvre d'un serveur d'authentification (la plupart du temps un serveur RADIUS), permettant d'identifier les utilisateurs sur le réseau et de définir leurs droits d'accès. Néanmoins, il est possible pour les petits réseaux de mettre en œuvre une version restreinte du WPA, appelée WPA-PSK, en déployant une même clé de chiffrement dans l'ensemble des équipements, ce qui évite la mise en place d'un serveur RADIUS. Ce mode de sécurité est compatible avec la norme wifi a/b/g.[17]

• **Le WPA2 :** L'amélioration la plus significative de WPA2 par rapport a WPA-est l'utilisation de l'Advanced Encryption Standard (AES) pour le chiffrement. C'est un algorithme extrêmement puissant. Il est considéré comme étant le plus sûr et le plus utilisé des algorithmes de chiffrement. La garantie fournie par AES est suffisante (et approuvé) pour une utilisation par le gouvernement américain pour crypter des informations classées top secret. Ce mode de sécurité est compatible avec la norme wifi a/b/g/n.[17]

• **Le filtrage d'adresse MAC :** sur chaque carte réseau possède un identifiant unique,
Le routeur wifi permet généralement de créer une liste des adresses MAC des cartes réseau qu'on autorise à se connecter à notre réseau.[17]

f) Les avantages et les inconvénients réseau WiFi

➢ **Les avantages**

- ❖ Mobilité : les utilisateurs sont généralement satisfaits des libertés offertes par un réseau sans fil et de fait sont plus enclins à utiliser le matériel informatique.
- ❖ Facilité et souplesse : un réseau sans fil peut être utilisé dans des endroits temporaires, couvrir des zones difficiles d'accès aux câbles, et relier des bâtiments distants.
- ❖ Coût : si leur installation est parfois un peu plus coûteuse qu'un réseau filaire, les réseaux sans fil ont des coûts de maintenance très réduits ; sur le moyen terme, l'investissement est facilement rentabilisé.
- ❖ Évolutivité : les réseaux sans fil peuvent être dimensionnés au plus juste et suivre simplement l'évolution des besoins. [18]

➢ **Les Inconvénients**

❖ Qualité et continuité du signal : Un réseau WiFi bien installé et bien configuré est généralement fiable et d'une qualité constante. Cependant, il suffit parfois de peu pour perturber le signal : un radar de gendarmerie ou un émetteur Bluetooth, par exemple.

❖ Sécurité : Le WiFi étant un réseau sans fil, il est possible de s'y connecter sans intervention matérielle. Cela veut dire qu'il faut particulièrement étudier la sécurisation du réseau si l'on veut éviter la présence d'indésirables ou la fuite d'informations.

❖ Vulnérable vis-à-vis de l'environnement extérieur, peut perturber le passage des ondes électromagnétiques (émetteurs d'ondes électromagnétiques, obstacles) , il suffit parfois de peu pour perturber le signal : un radar de gendarmerie ou un émetteur Bluetooth, par exemple.[18]

Protocole	Zigbee	Bluetooth	Wi-Fi
IEEE	802.15.4	802.15.1	802.11a/b/g/n/n-draft
Besoins mémoire	4-32 Kb	250 Kb +	1 Mb +
Autonomie avec pile	Années	Jours	Heures
Nombre de nœuds	65000+	7	32
Vitesse de transfert	250 Kb/s	1 Mb/s	11-54-108-320 Mb/s
Portée	100 m	10-100 m	300 m

Tableau I.3: *la différence entre les protocoles*

I.5.2.4 La technologie RFID

I.5.2.4.1 La définition de RFID

Le terme RFID (de l'anglais radio frequency identification) englobe toutes les technologies qui utilisent les ondes radio pour identifier automatiquement des objets ou des personnes. Le système RFID, autrement dit l'identification par radio-fréquence, est une technologie qui permet de mémoriser et de récupérer des informations à distance grâce à une étiquette qui met des ondes radio. [19]



Figure I.9 : *Logo de RFID*

Cette technologie d'identification peut être utilisée pour identifier :

- Les objets, comme avec un code-barres (on parle alors d'étiquette électronique)
- Les personnes, en étant intégrée dans les passeports, carte de transport, carte de paiement (on parle alors de carte sans contact) .
- Les carnivores domestiques (chats, chiens et furets) dont l'identification RFID est obligatoire dans de nombreux pays, en étant implantée sous la peau. C'est également le cas de manière non obligatoire pour d'autres animaux de travail, de compagnie ou d'élevage de rente (on parle alors de puce sous-cutanée).

I.5.2.4.2 Les composants du système RFID

Un système RFID est généralement composé de trois types de dispositifs : tags, lecteurs et un serveur central d'arrière-plan communément appelé back-end. Ces entités interagissent ensemble via des protocoles de communication où des messages sont échangés dans le but d'atteindre un objectif donné (identifier ou authentifier les tags du système).

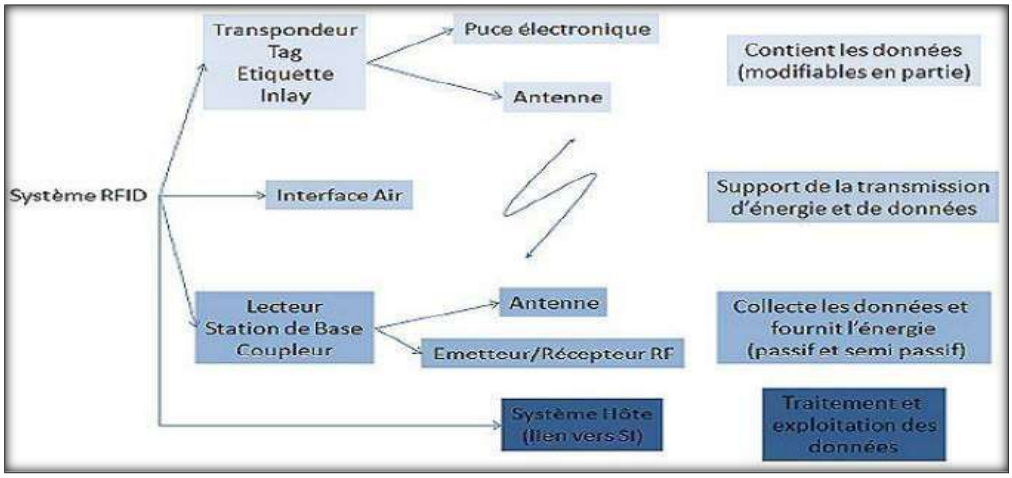


Figure I.10 : *composition du système RFID*

- **Le lecteur (Interrogateurs)**

Ce sont des dispositifs actifs, émetteurs de radiofréquences Interrogateurs RFID (Lecteur). Le lecteur/enregistreur est constitué d'un circuit qui émet une énergie électromagnétique à travers une antenne, et d'une puce électronique qui reçoit et décode les informations envoyées par l'étiquette et les envoie au dispositif de collecte des données. Le lecteur RFID est l'élément responsable de la lecture des étiquettes radiofréquence et de la transmission des informations qu'elles contiennent.[20]

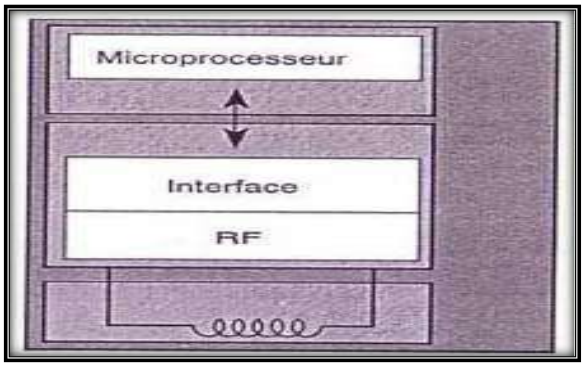


Figure I.11: Interrogateurs RFID (Lecteur)

- **Le back-end**

Il contient une base de données qui stocke les informations relatives à chaque tag et lecteur du système. Le back-end peut aussi être une sorte de Switch qui ne fait que transférer les communications entre les lecteurs. Ce composant n'est pas toujours nécessaire dans un système RFID.[21]

- **L'étiquette (tag)**

L'étiquette RFID également nommée étiquette intelligente, étiquette à puce ou tag est un support d'identification électronique qui n'a pas besoin d'être vu pour être lu. C'est un dispositif récepteur, que l'on place sur des éléments (objet, animal…). Ils sont munis d'une puce contenant les informations et d'une antenne pour permettre les échanges d'informations. Il existe trois types d'étiquette : des étiquettes passives, des étiquettes actives et des étiquettes semi actives.[20]

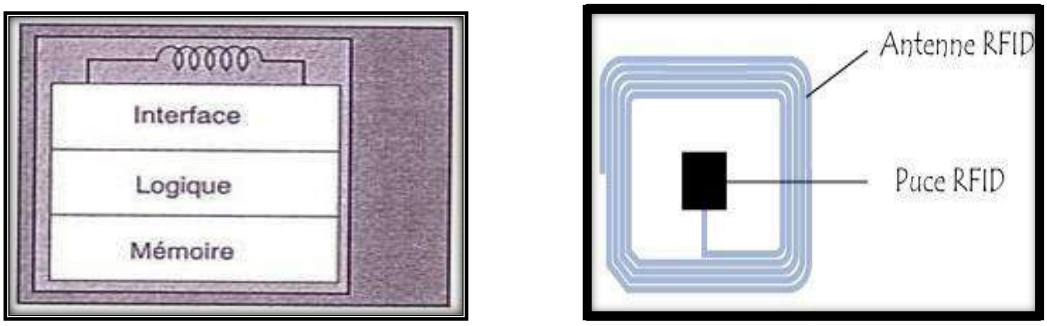


Figure I.12: *Représentation de l'étiquette d'une RFID*

- **Les caractéristiques des tags RFID**

Les principales caractéristiques des tags RFID sont représentées sur la figure et décrites ci-dessous :

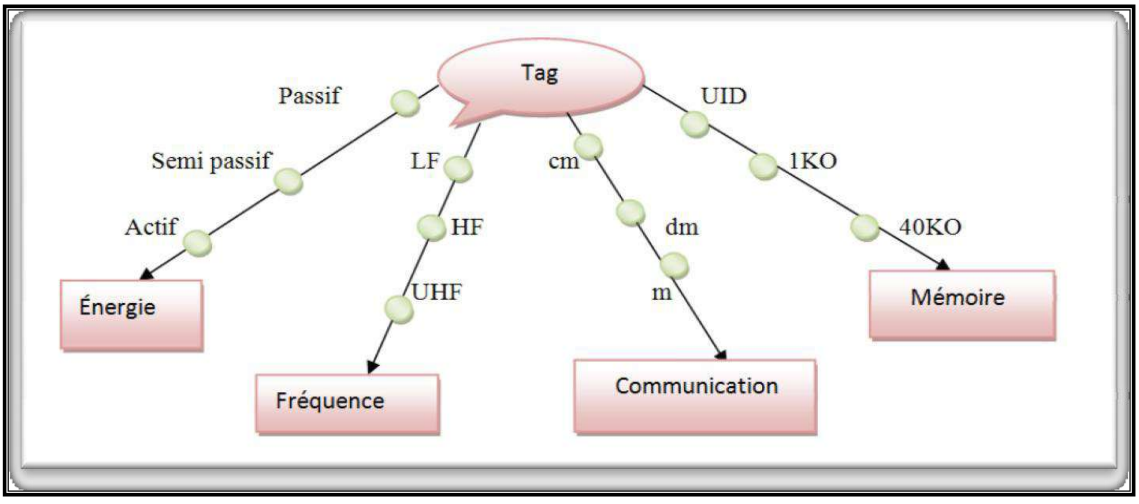


Figure I.13 *: Les caractéristiques des tags RFID*

- **Énergie**

Selon la source d'énergie considérée, on distingue trois classes de tags :

- Tags actifs : ils sont alimentés par leur propre batterie pour leurs calculs internes et leurs communications avec les lecteurs.
- Tags passifs : les tags alimentés par le champ électromagnétique des lecteurs.
- Tags semi passif : ils utilisent leur propre batterie pour les calculs 1Aet l'énergie émise par le lecteur pour les communications.[22]

- **La Fréquence**

Les systèmes RFID génèrent et réfléchissent des ondes électromagnétiques, ce sont donc des systèmes radio qui ne doivent pas perturber le fonctionnement des autres systèmes radio (Télévision, téléphones mobiles, etc.).

Trois types de fréquences sont utilisés pour les puces RFID :

- Basse fréquence (125Khz) : identification animale.
- Haute fréquence (13,56 Mhz) : paiement, contrôle d'accès, billettique.
- Très haute fréquence (UHF) (433 Mhz) : contrôle d'accès pour les parkings.

- **Distance de communication**

Plusieurs paramètres influent sur la distance de communication entre un tag et un lecteur notamment la fréquence, la puissance d'émission, l'environnement, l'antenne, etc. Pour les tags passifs, il est possible d'exhiber les trois distances de communication suivantes en fonction des bandes de fréquence.

- Basse Fréquence (LF) : quelques centimètres.
- Haute Fréquence (HF) : quelques centimètres à quelques décimètres.
- Ultra-hautes Fréquence (UHF) : quelques mètres.

Ces distances sont données selon les standards et spécifications des fabricants. Cependant, plusieurs études ont démontré que ces distances peuvent être considérablement augmentées avec un matériel de lecture adéquat.[22]

❖ **Mémoire**

La quantité de mémoire disponible sur un tag dépend des besoins de l'application RFID. Il est nécessaire d'avoir au minimum quelques dizaines de bits pour stocker l'UID (Unique Identifier) du tag.

I.5.2.4.3 Fonctionnement du système RFID

L'étiquette RFID (ou transpondeur ou tag) est elle-même équipée d'une puce reliée à une antenne. L'antenne permet à la puce de transmettre les informations (numéro de série, poids...) qui peuvent être lues grâce à un lecteur émetteur-récepteur. Une fois les informations transmises au lecteur RFID équipé d'une antenne intégrée ou externe, celui-ci n'a plus qu'à convertir les ondes-radios en données et celles-ci pourront être lues par un logiciel RFID. [23]

La figure ci-dessous représente le fonctionnement d'un système RFID :

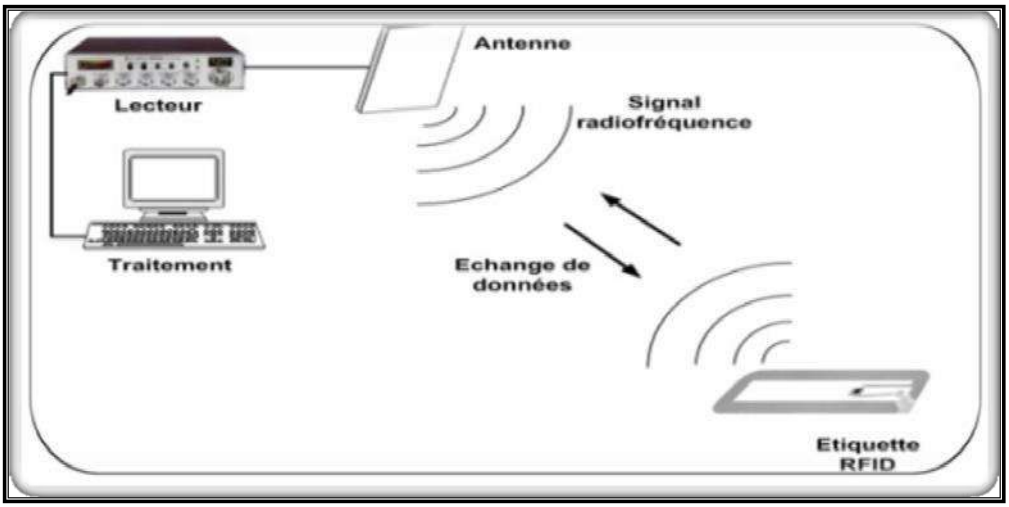


Figure I.14 : *Fonctionnement du système RFID*

❖ Le couplage tag RFID / lecteur RFID

La liaison entre tag et interrogateur se réalise par :

- Couplage magnétique : dans le cas d'un champ proche (quelques cm à 1,5 m). L'interrogateur utilise alors des LF (Basses Fréquences) ou des HF (Hautes Fréquences). Les antennes sont alors constituées de boucles inductives.

- Couplage électrique dans le cas d'un champ lointain (jusqu'à 15 m). L'interrogateur utilise alors des UHF (Ultra Hautes Fréquences) ou des SHF (Super Hautes Fréquences). Les antennes de base sont alors des dipôles ou des patchs.[20]

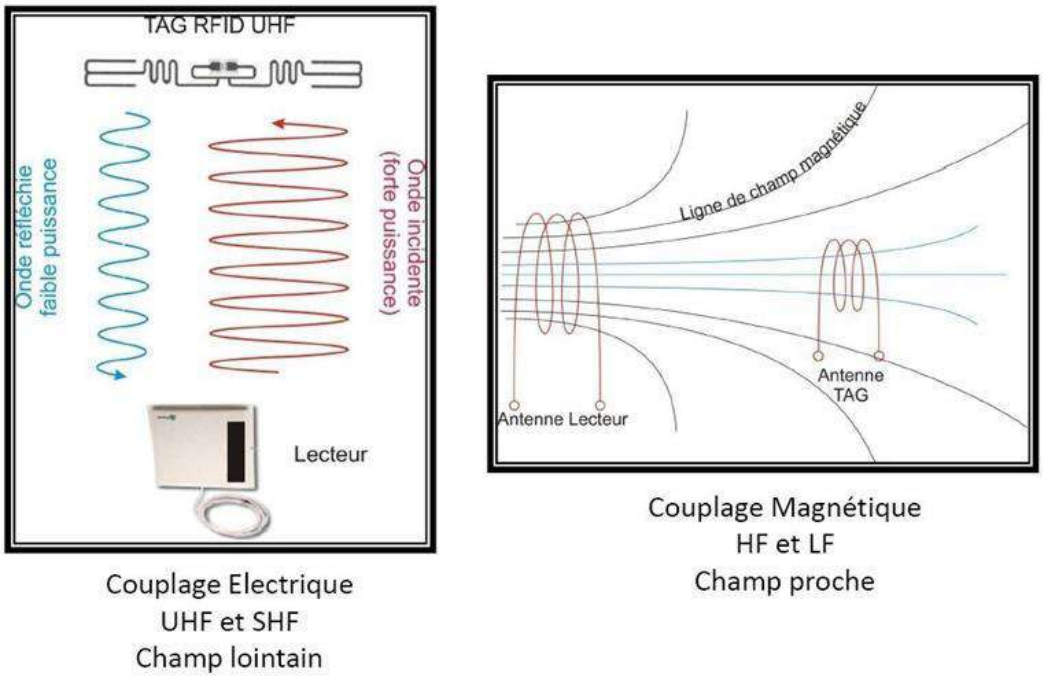


Figure I.15 : *Le couplage tag RFID / lecteur RFID*

❖ Autres classifications

On pourrait aussi classer les systèmes RFID selon, la capacité de la mémoire, le format des tags, la capacité de traitement de l'information, la procédure utilisée par le transpondeur pour renvoyer des données au lecteur, le type de codage binaire et d'autres critères encore.
On retiendra qu'il existe une multitude de systèmes, qu'il faut choisir selon l'usage que l'on veut en faire, en étudiant les aspects suivants : portée, environnement électromagnétique, réglementation, prix, interopérabilité, capacité de traitement, capacité mémoire, sécurité.

I.5.2.4.4 La sécurité

Les enjeux relatifs à la sécurité des tags RFID sont nombreux et importants. Le concepteur de systèmes RFID rencontre les mêmes problèmes que celui qui conçoit des cartes à puces : comment empêcher qu'un attaquant puisse lire, modifier ou fabriquer, voire dupliquer les données d'un tag, de manière à perturber le système et éventuellement obtenir frauduleusement un accès à un bâtiment ou un service ? Il s'agit d'un problème complexe qui ne connait pas de réponses parfaitement satisfaisantes.

Dans le cas des systèmes RFID, le fait que les échanges de données se font par ondes radio, ou champs électromagnétiques, ajoute une dimension supplémentaire au problème. D'une part, les

menaces qui pèsent sur les smart cartes avec contact physique sont renforcées par le fait qu'un attaquant peut interagir avec un tag RFID à distance, sans même que son propriétaire légitime ne s'en rende compte. D'autre part, il existe une nouvelle menace, qui pose un défi de taille aux concepteurs : comment empêcher un attaquant d'intercepter la communication radio, pour ensuite « rejouer » la transaction en imitant le tag original.[20]

I.5.2.4.5 Les problèmes techniques

La technologie RFID, si elle a un petit côté « magique », ne fonctionne pas sans problèmes en voici quelques-uns, strictement limités au domaine technique.[20]

- **L'orientation des antennes**

Les tags RFID ne nécessitent pas de liaison optique pour fonctionner, mais les lecteurs ne peuvent pas communiquer normalement avec un tag dont l'antenne est orientée perpendiculairement à l'antenne du lecteur. On constate le même effet quand on essaie de recevoir une station radio très faible avec une petite radio portable. Si plusieurs produits sont disposés au hasard dans un chariot à commissions, certains seront orientés de telle sorte qu'ils seront invisibles par le lecteur.

Si les produits marqués ne peuvent pas être réorientés, il faut alors modifier l'orientation du lecteur ou construire des antennes moins sensibles à l'orientation. Une autre approche est de mélanger les données de plusieurs lecteurs orientés de manières différentes.

On peut encore installer plusieurs antennes et les connecter au lecteur l'une après l'autre, ce qui est certainement la solution la plus économique.

- **Les collisions**

Les premiers tags RFID, fabriqués dans les années 1960, fonctionnaient sur de basses fréquences. De nos jours, des fréquences élevées, typiquement sur la bande UHF, améliorent la capacité d'un lecteur à lire plusieurs tags se trouvant dans son champ de lecture, car le taux de transfert est plus grand et les données de chaque tag sont transmises plus rapidement, réduisant la probabilité de collision entre les données de chacun des tags typiquement lors de la lecture des produits dans un chariot de supermarché.

Pour réduire les chances que deux tags émettent en même temps (collision), les tags utilisent un protocole anticollision pour contrôler la fenêtre de temps durant laquelle chacun répond ce temps dépend du numéro unique des tags.

- **Les lecteurs multiples**

La plupart des lecteurs RFID ne sont pas conçus pour fonctionner en présence d'un autre lecteur qui scannerait les tags en même temps. L'ISO est en train d'établir des normes à ce sujet.

- **Les matériaux d'emballage**

Les ondes radio peuvent être perturbées par certains emballages, notamment les métaux ferreux

- **Les coûts**

Le problème principal des systèmes RFID reste leur coût de fabrication. En 2005, un tag passif pour étiqueter les produits coûte environ 0.25 $US. Or, cela n'a pas de sens de placer un tel tag sur un produit qui coûte à peine plus chère. On estime que les tags RFID pourront se répandre dans la grande distribution quand on pourra les produire pour 0.05 $US.

I.5.2.4.6 Les Avantages de RFID

La technologie RFID possède plusieurs avantages : [24]

- Contrairement aux codes-barres, l'interaction entre l'étiquette et le lecteur ne requiert pas une proximité optique.
- L'étiquette peut contenir plus d'informations que le code-barres.
- L'étiquette peut être reprogrammée et réutilisée.
- L'étiquette est résistante à la chaleur, à l'eau et à l'usure.

I.5.2.4.7 Les applications pour la technologie RFID

La RFID est une technologie répandue dans le monde entier. Un des usages les plus évidents se situe dans les univers de la logistique et de la vente. La RFID est également beaucoup utilisée dans les bibliothèques pour la traçabilité et la protection des livres. Mais ce ne sont pas les seuls usages qui sont faits de la technologie RFID. Parmi les plus connus, on peut citer :[25]

- Le contrôle d'accès
- Les cartes de paiement sans contact
- Les passports

- Les puces sous-cutanées des animaux domestiques
- L'accès aux transports publics.
- Les télépéages d'autoroutes et la gestion des parcs.

I.6 Conclusion

On a conclu de ce chapitre que la maison intelligente peut rendre la vie de m'être humain moins difficile, en lui protégeant et économisant son temps, son effort et son argent. On a basé aussi sur quelques technologies de connexion possibles qu'on a utilisé dans la domotique.


Chapitre II
matériel et logiciel utilisé dans le projet

II.1 Introduction

Le composant principale de notre réalisation est l'Arduino qui nous permet de contrôler et gérer tous les autres composant qu'on va utiliser dans la domotique.

Dans ce chapitre on a expliqué d'une façon générale le matériel et les logiciels utilisés dans la domotique mais concernant notre travail, nous allons détailler chaque composant, ses caractéristiques et son mode de fonctionnement, ensuite parler aussi sur plusieurs logiciels utilisé dans ce projet, il y a des logiciels de programmation et des logiciels de simulation.

II.2 Matériel

II.2.1 Carte Arduino Méga 2560

II.2.1.1 Définition

La carte Arduino Méga 2560 est basée sur un microcontrôleur ATMega2560 cadencé à 16 MHz. Elle dispose de 54 E/S dont 14 PWM, 16 analogiques et 4 UARTs. Elle est idéale pour des applications exigeant des caractéristiques plus complètes que la Uno. Des connecteurs situés sur les bords extérieurs du circuit imprimé permettent d'enficher une série de modules complémentaires. Elle peut se programmer avec le logiciel Arduino. Le contrôleur ATMega2560 contient un bootloader qui permet de modifier le programme sans passer par un programmateur. Le logiciel est téléchargeable gratuitement.[26]

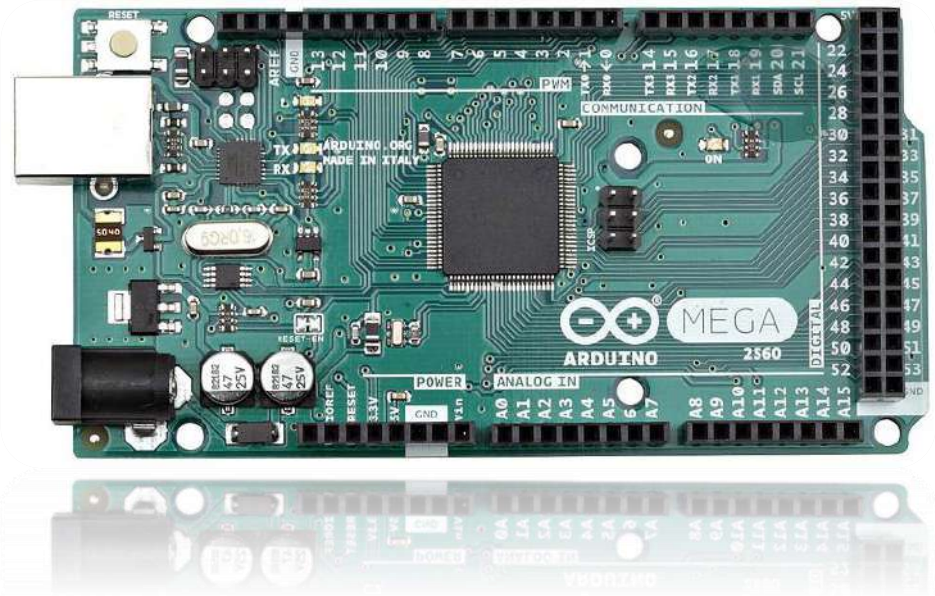


Figure II. 1: *La carte Arduino Méga.*

II.2.1.2 Les caractéristiques principales

Tableau II.1 : *Caractéristiques de l'Arduino*

Catégorie	Valeur
Microcontroleur	**ATmega 2560**
Fréquence D'horloge	**16 MHz**
Tension de service	**5 V**
Tension d'entrée	**7-12 V**
Tension d'entrée (limites)	**6-20 V**
Ports numériques	**54 E/S (15 Sortie PWM)**
Ports analogiques	**16 entrées analogiques**
Courant maxi.par broche d'E/S (c.c)	**40 mA**
Courant maxi.par broche 3.3 V	**50 mA**
Mémoire	**256 Ko Flash, 8 Ko SRAM, 4 Ko EEPROM**
Chargeur D'amourçage	**8 Ko (en mémoire Flash)**
Interface	**USB**
Dimensions	**10.6 cm X 5.3 cm**

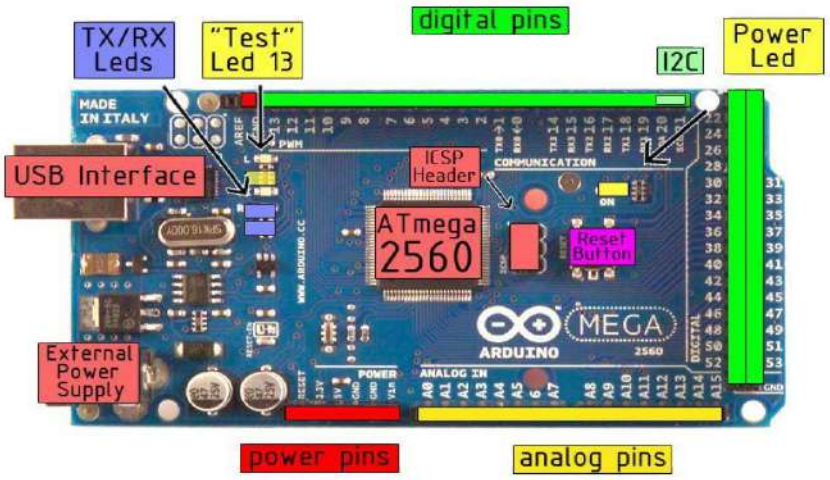


Figure II. 2: Explication de la carte Arduino Méga.

II.2.1.3 Les avantages et les inconvénients

- **Les avantages**

- Nombreuses entrées et sorties pour raccorder des capteurs ou des actionneurs.
- Capacité de mémoire suffisante pour les gros projets,
- Plus de broches UART (4 ports de communication série).
- Plus de broches ML (15 sorties numériques peuvent être utilisées comme MLI).
- Compatible avec la plupart des Shields conçus pour l'Arduino Uno, par exemple.
- Il existe des Shields spéciaux pour le prototypage qui, en raison de leur surface supérieure, peuvent recevoir plus de composants.
- De nombreux schémas et exemples sont disponible sur internet.[27]

- **Les inconvenients:**

- Facteur de forme plus élevé que l'Arduino Uno, par exemple.
- Deux fois plus chère que l'Arduino Uno.[27]

II.2.2 La carte Arduino Uno

II.2.2.1 Définition

Arduino Uno est une carte à microcontrôleur basée sur l'ATmega328. Il a 14 broches numérique d'entrée / sortie (dont 6 peuvent être utilisées en tant que sorties PWM), 6 entrées analogiques, une céramique de 16 MHz résonateur, une connexion USB, une prise d'alimentation, un en-tête ICSP et un bouton de réinitialisation. Il contient tout nécessaire pour supporter le microcontrôleur ; connectez-le simplement à un ordinateur avec un câble USB ou alimentez-le avec un adaptateur AC / DC ou une batterie pour commencer.[28]

L'Uno diffère des cartes précédentes en ce qu'il n'utilise pas la puce de pilote FTDI USB-to-serial. Au lieu de cela, il dispose de l'Atmega16U2 (Atmega8U2 jusqu'à la version R2) programmé en tant que port USB à série convertisseur.[29]



Figure II. 3: *La carte Arduino Uno*

II.2.2.2 Communication d'Arduino Uno

L'Arduino Uno dispose de nombreuses fonctionnalités pour communiquer avec un ordinateur, un autre Arduino ou autres microcontrôleurs. L'ATmega328 fournit une communication série UART TTL (5V), qui est disponible sur les broches numériques 0 (RX) et 1 (TX). Un ATmega16U2 sur la carte canalise cette série communication via USB et apparaît comme un port de communication virtuel pour le logiciel sur l'ordinateur. Le '16U2 micrologiciel utilise les pilotes COM USB standard et aucun pilote externe n'est nécessaire.[30]

Cependant, sous Windows, un fichier .inf est requis. Le logiciel Arduino comprend un moniteur série qui permet aux données textuelles simples d'être envoyé de et vers le conseil Arduino. Les voyants RX et TX de la carte clignotent lorsque des données sont en cours de traitement transmis via la puce USB vers série et la connexion USB à l'ordinateur (mais pas pour les communications série). Une bibliothèque Software-Serial permet la communication en série sur n'importe laquelle des broches numériques du Uno.L'ATmega328 prend également en charge les communications I2C (TWI) et SPI. Le logiciel Arduino comprend un Bibliothèque de fils pour simplifier l'utilisation du bus I2C; . Pour la communication SPI nous utilisons la bibliothèque SPI.[30]

II.2.2.3 Les Caractéristiques principales

.

Tableau II. 2 : *Caractéristiques de l'Arduino Uno*

Catégorie	Valeur
Microcontroleur	**ATmega 328**
Fréquence D'horloge	**16 MHz**
Tension de service	**5 V**
Tension d'entrée	**7-12 V**
Tension d'entrée (limites)	**6-20 V**
Ports numériques	**14 E/S (6 Sortie PWM)**
Ports analogiques	**6 entrées analogiques**
Courant maxi.par broche d'E/S (c.c)	**40 mA**
Courant maxi.par broche 3.3 V	**50 mA**
Mémoire	**32 Ko Flash, 2 Ko SRAM, 1 Ko EEPROM**
Chargeur D'amourçage	**0.5 Ko (en mémoire Flash)**
Interface	**USB**
Dimensions	**6.86 cm X 5.3 cm**

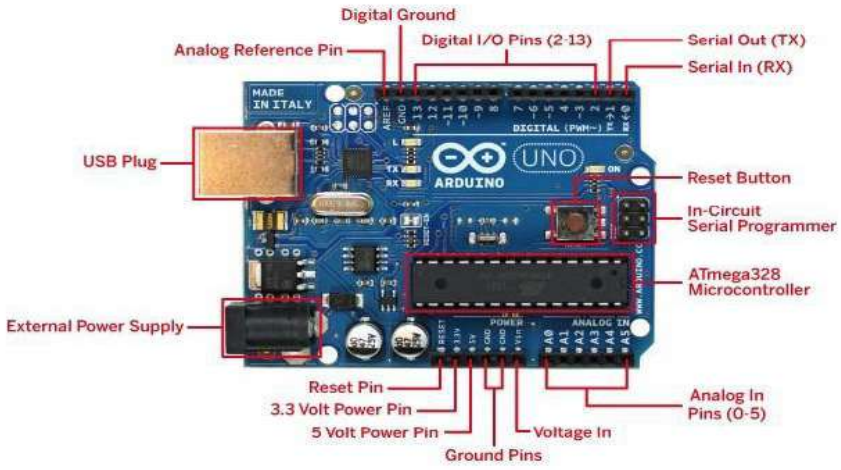


Figure II. 4: *Explication de la carte Arduino Méga.*

II.2.2.4 Les avantages et les inconvénients

- **Les Avantages**

- Carte Arduino par excellence pour laquelle de nombreux exemples de montage sont disponible sur internet.
- Nombre suffisant de broches d'entrée-sorties pour les projets élémentaires.
- Vaste choix de shield.
- Bon marché.[31]

- **Les inconvénients**

- Nombre insuffisant de broches d'entrée-sorties pour les projets ambitieux.
- La mémoire disponible risque d'être un peu juste pour les gros projets.
- Ne peut pas être utilisée comme hôte USB pour simuler un clavier ou une souris.[32]

II.2.3 Le module Wi-Fi (ESP8266-01)

II.2.3.1 Définition

L'ESP8266 est un circuit intégré à microcontrôleur avec connexion Wi-Fi développé par le fabricant chinois Espressif. Ce module est un SOC autonome (Système sur puce) qui n'a pas nécessairement besoin d'un microcontrôleur pour manipuler les entrées et les sorties comme vous le feriez normalement avec un Arduino, par exemple, parce que l'ESP-01 agit comme un petit ordinateur.[33]

En fonction de la version de l'ESP8266, il est possible d'avoir jusqu'à 9 GPIO (General Purpose Input Output). Ainsi, nous pouvons donner un accès Internet à un microcontrôleur, comme le fait le bouclier Wi-Fi à l'Arduino, ou simplement programmer l'ESP8266 pour avoir non seulement accès à un réseau Wi-Fi, mais également agir comme un microcontrôleur.[34]



Figure II. 5: *L'ESP8266-01*

Il permet de connecter n'importe quel microcontrôleur (du moment qu'il possède un UART Rx/Tx) à un réseau WiFi. Le module possède 8 broches et il doit être alimenté en 3,3v. Mais attention il en existe deux sortes, un couleur bleu avec 512Mo de mémoire et le noir (comme ci-dessous) avec 1024Mo. Dans le cadre cet exemple, il s'agira de la version 1024, aussi appelée ESP-01S. Mais avant d'apprendre à Flasher un ESP-01 avec le Logiciel Flash_download_tools. [35]

L'ESP8266 est disponible en plusieurs versions, il faut donc veiller à prendre celle qui convient le mieux.



Figure II. 6: *La gamme ESP8266XY.*

II.2.3.2 Les caractéristiques du module

- Le plus petit module Wi-Fi SOC 802.11b/g /n.
- Wi-Fi Direct (P2P), soft-AP.
- CPU 32 bits basse consommation, peut également servir de processeur d'application.
- 10-bit ADC intégré.
- Vitesse d'horloge jusqu'à 160 MHz.
- SDIO 2.0, (H) SPI, UART, I2C, I2S, IRDA, PWM, GPIO.
- Emballage DIP-8 pour un soudage facile.

- Wi-Fi intégré MAC / BB / RF / PA / LNA.
- Sommeil profond <10uA, courant de fuite <5uA.
- Pile de protocoles TCP / IP intégrée.
- Commutateur TR intégré, balun, LNA, amplificateur de puissance et réseau d'adaptation.
- Prise en charge de la diversité d'antenne.
- Wi-Fi 2,4 GHz, prend en charge WPA / WPA2.
- Prise en charge des modes de fonctionnement STA / AP / STA + AP.
- Prise en charge de la fonction Smart Link pour les appareils Android et iOS.
- Prise en charge de la fonction Smart Link pour les appareils Android et iOS.
- PLL, régulateurs et unités de gestion de l'alimentation intégrés.
- Prise en charge du mode de fonctionnement STA / AP / STA + AP.
- Plage de température de fonctionnement -40 ° C ~ 125 ° C.
- Débit UART jusqu'à 4 Mbps.
- Réveillez-vous et transmettez les paquets en moins de 2ms.
- Les commandes AT générales peuvent être utilisées rapidement.
- Puissance de sortie +20 dBm en mode 802.11b.
- Agrégation A-MPDU et A-MSDU et intervalle de garde de 0,4 s.[36]

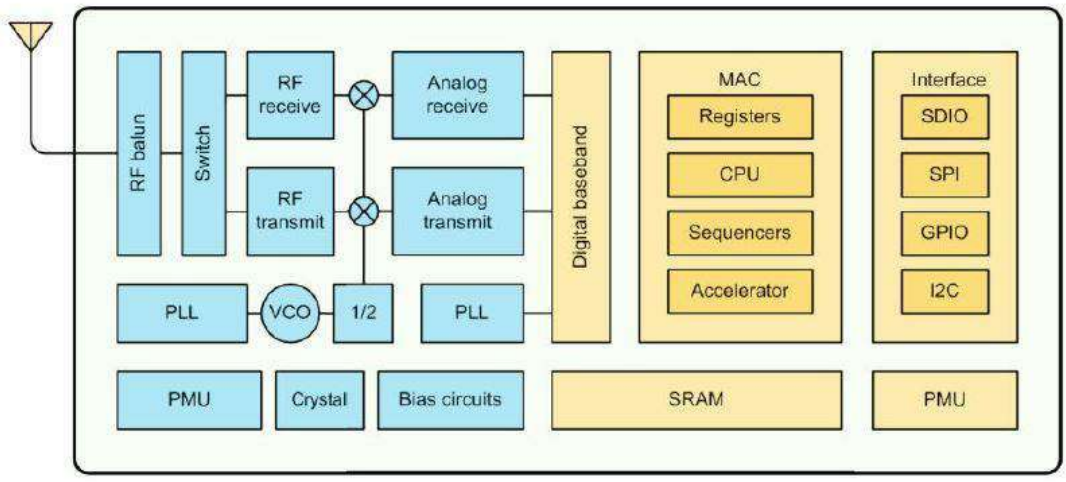


Figure II. 7: *Diagramme de bloque de L'ESP8266*

II.2.3.3 Spécifications du module

Tableau II. 3 : *Spécifications de L'ESP8266-01*

Module Model	ESP-01
Package	DIP-8
Size	24.7*14.4*11.0C±0.23mm
SPI Flash	Default 8Mbit
Interface	UART/GPIO
IO Port	2
UART Baud rate	Support 300 ~ 4608000 bps, Default 115200 bps
Frequency Range	2412 ~ 2484MHz
Antenna	PCB antenna, 2 dBi
Secondary Development	Support
Transmit Power	802.11b: 16±2 dBm (@11Mbps) 802.11g: 14±2 dBm (@54Mbps) 802.11n: 13±2 dBm (@HT20, MCS7)
Receiving Sensitivity	CCK, 1 Mbps: - 90dBm CCK, 11 Mbps: -85dBm 6 Mbps (1/2 BPSK): -88dBm 54 Mbps (3/4 64-QAM): -70dBm HT20, MCS7 (65 Mbps, 72.2 Mbps): -67dBm
Power (Typical Values)	Continuous Transmission => Average : ~ 71mA, Peak : 300mA Modem Sleep : ~20mA Light Sleep: ~2mA Deep Sleep: ~0.02mA
Security	WEP/WPA-PSK/WPA2-PSK
Power Supply	Voltage 3.0V ~ 3.6V， Current >300mA
Operating Temperature	-20 ~ 85
Storage Environment	-40 ~ 90 < 90%RH
Weight	0.65g

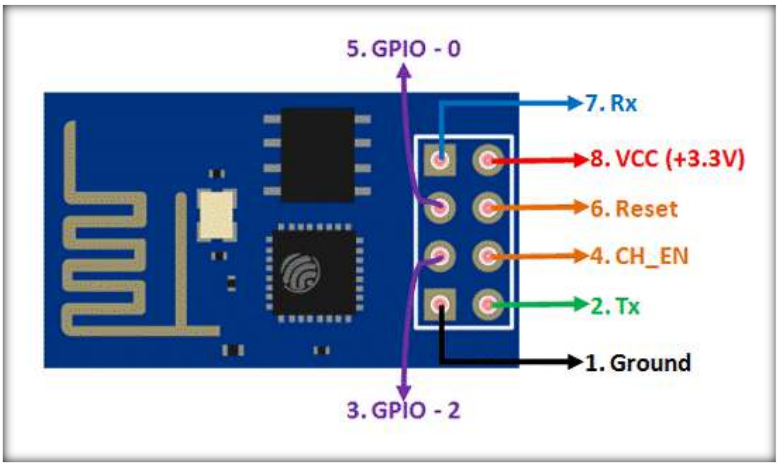


Figure II.6: *Les brochage de l'ESP8266-01*

II.2.3.4 Fonctionnement

L'ESP8266-01 il ne fonctionne qu'en 3.3 V et n'est pas tolérant au 5 V. Il a deux GPIO (0 & 2) qui peuvent être utilisés pour la communication I2C. Il est pourvu d'un port RS232 qu'on peut utiliser pour lui envoyer des commandes AT. On a présente deux façons différentes d'utiliser cet ESP :

- En lui envoyant des commandes AT sur son port RS232 avec un ordinateur ou un microcontrôleur.
- En programmant directement son microcontrôleur interne en Arduino C.

Pour la première partie sur les commandes AT, le firmware NodeMCU doit être installé sur l'ESP. C'est en général le cas pour les ESP neufs. Lorsque l'ESP est programmé en Arduino C, cela écrase le firmware NodeMCU, mais rien n'est perdu, on peut le flasher à nouveau. À noter encore que NodeMCU ne permet pas seulement d'utiliser les commandes AT, mais permet également de programmer l'ESP dans le langage Lua.[37]

II.2.3.5 Les avantages et les inconvénients du module

- **Les avantages :**

Le plus grand avantage de l'ESP8266 est peut-être son coût. Il est tout à fait possible d'en acheter quelques-unes en une fois. Avant que je n'aie appris à le savoir. Les nouveaux ESP8266 sont en cours de sortie. ESPONLY ESP est maintenant en rupture de stock. ESPRITORIAL ESP8266, il est très populaire. En outre, lorsque vous achetez l'ESP8266, il est livré préchargé avec le firmware AT par défaut.[38]

- **Les inconvenient:**

Chaque appareil présente des avantages et des inconvénients, et l'ESP n'est pas différent. L'ESP peut parfois s'avérer être très difficile et frustrant. Avec la nouveauté, il vous sera difficile d'obtenir des informations à ce sujet. Heureusement, une communauté chez esp8266 .com existe, ce qui est très utile. En outre, il commence aussi parfois à faire des choses inattendues, comme jeter des ordures dans la connexion série, etc.

Notez qu'il existe une grande quantité de documentation disponible sur Internet et qu'une partie de celle-ci est en conflit. Cet Inscrutable n'est pas différent. Tout en jouant avec mon ESP8266, j'ai constaté qu'il variait beaucoup de ce qui était mentionné sur Internet (le vôtre aussi) mais ça a bien fonctionné. [38]

II.2.4 Le module RFID

La RFID est basée sur l'échange d'informations véhiculées par des ondes électromagnétiques entre une étiquette (tag) et un lecteur. Dans notre réalisation, on utilise comme système RFID les modules suivants :

II.2.4.1 Le module RC522

Est une interface qui permet l'identification sans contact à partir d'un badge ou une clé RFID, Il est basé sur le circuit intégré MFRC522 de NXP. Le module de lecteur RFID RC522 est conçu pour créer un champ électromagnétique de 13,56 MHz qu'il utilise pour communiquer avec les tags RFID. Le lecteur peut communiquer avec un microcontrôleur via une interface SPI (Serial Peripheral Interface) à 4 broches avec un débit binaire maximal de 10 Mbits / s. Il prend également en charge la communication via les protocoles I2C et UART. La tension de fonctionnement du module est comprise entre 2,5 et 3,3 V, mais la bonne nouvelle est que les broches logiques sont tolérantes à 5 V, ce qui permet de le connecter facilement à un microcontrôleur Arduino ou à tout autre microprocesseur logique 5 V sans passer par aucun convertisseur de niveau logique.[39]

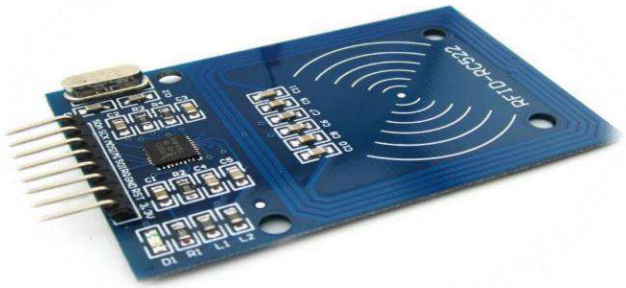


Figure II. 8: *Le module RC522*

➢ **Les spécifications techniques**

- Protocol : SPI.
- Dimensions: 40mm × 60mm.
- Working temperature : -20 _ 80 degree .
- Storage temperature : -40_ 85 degree.
- Humidité: humidité pertinente 5% _ 95%.
- Nom du module : MF522-ED.
- Gamme de fréquences 13,56 MHz Bande ISM.
- Interface hôte SPI / I2C / UART.
- Tension d'alimentation de fonctionnement 2,5 V à 3,3 V.
- Max. Courant de fonctionnement 13-26mA.
- Min. Courant (mise hors tension) 10µA.
- Entrées logiques 5V Tolérant.
- Portée de lecture 5 cm.

- **RC522 RFID Module Pinout**

Le module RC522 a un total de 8 broches qui l'interfèrent avec le monde extérieur. Les connexions sont les suivantes :[40]

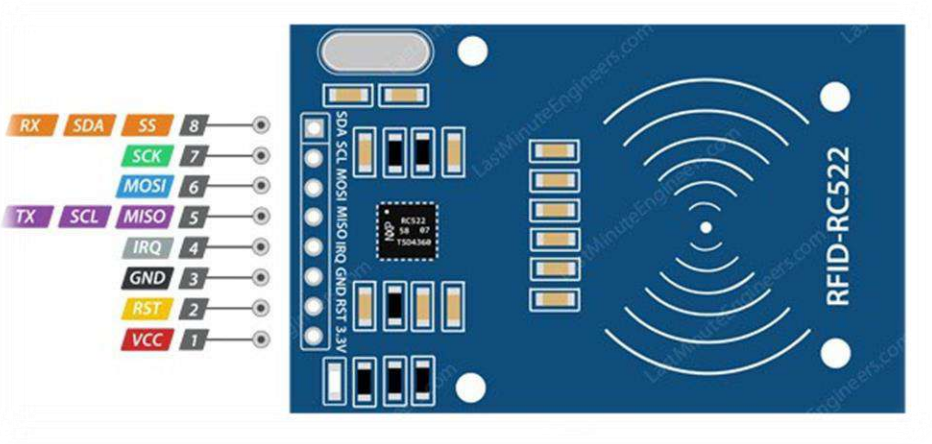


Figure II. 9: *Le brochage de RFID-RC522*

- **VCC** fournit l'alimentation au module. Cela peut être n'importe où de 2,5 à 3,3 volts. Vous pouvez le connecter à la sortie 3.3V de votre Arduino. N'oubliez pas que le connecter à une broche 5V détruira probablement votre module !
- **RST** est une entrée pour la réinitialisation et la mise hors tension. Lorsque cette broche devient faible, la mise hors tension matérielle est activée. Cela désactive tous les récepteurs de courant internes, y compris l'oscillateur et les broches d'entrée, sont déconnectés du monde extérieur. Sur le front montant, le module est réinitialisé.
- **GND** est la broche de terre et doit être connecté à la broche GND de l'Arduino.
- **IRQ** est une broche d'interruption pouvant alerter le microcontrôleur lorsque l'étiquette RFID se trouve à proximité.
- **MISO / SCL / Tx** La broche agit en tant que Master-In-Slave-Out lorsque l'interface SPI est activée, agit en tant qu'horloge série lorsque l'interface I2C est activée et agit en tant que sortie de données série lorsque l'interface UART est activée.
- **MOSI** (Master Out Slave In) Est l'entrée SPI du module RC522.
- **SCK** (Serial Clock) accepte les impulsions d'horloge fournies par le bus maître SPI, c'est-à-dire Arduino.

- **SS / SDA / Rx** La broche agit comme entrée de signal lorsque l'interface SPI est activée, agit comme données série lorsque l'interface I2C est activée et agit comme entrée de données série lorsque l'interface UART est activée. Cette broche est généralement marquée en la recouvrant d'un carré afin de pouvoir être utilisée comme référence pour identifier les autres broches.

II.2.4.2 Porte-clés

Le porte-clés RFID est un produit qui permet un contrôle d'accès fiable et Sécuritaire à l'entrée des immeubles, des parkings ou de portes sécurisées grâce au tag RFID.



Figure II. 10 : *Porte-clés de RFID*

II.2.4.3 Le Badge

Ces badges RFID sont munis d'une antenne et d'une puce aussi, pour permettre la transmission de données avec un lecteur. Le badge RFID est au format : 84 x 56 x 0.76 mm

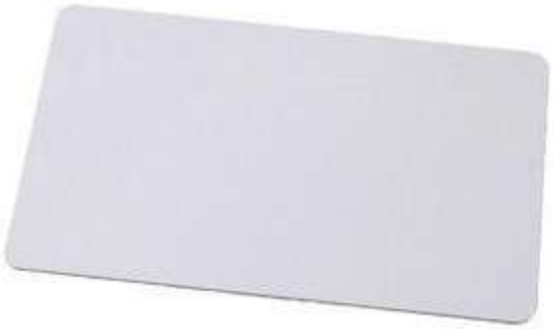


Figure II. 11: *badge de RFID*

II.2.5 Afficheur LCD 16x2

II.2.5.1 Définition

L'écran LCD (écran à cristaux liquides) est un module d'affichage électronique qui permet de trouver un large éventail d'applications. Un écran LCD 16x2 est un module très basique et est très couramment utilisé dans divers dispositifs et circuits. Ces modules sont préférés aux sept segments et aux autres DEL multi-segments . Les raisons étant: les LCD sont économiques; facilement programmable; n'a aucune limitation à afficher des caractères spéciaux et même personnalisés (contrairement à sept segments), des animations , etc.[41]

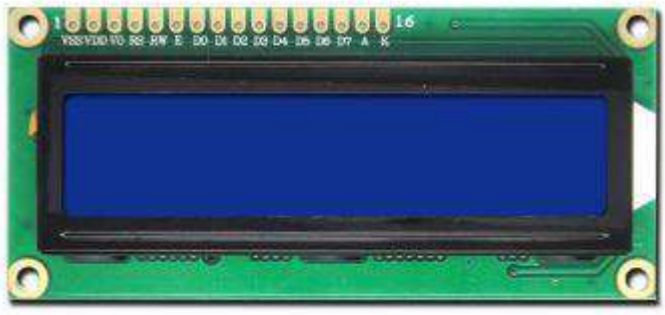


Figure II. 12: *Afficheur LCD 16x2*

II.2.5.2 Fonctionnement

Un écran LCD 16x2 signifie qu'il peut afficher 16 caractères par ligne et qu'il en existe 2. Dans cet écran LCD, chaque caractère est affiché dans une matrice de 5x7 pixels. Cet affichage à cristaux liquides à deux registres, à savoir, commande et données, Le registre de commandes stocke les instructions de commandes données à l'écran LCD. Une commande est une instruction donnée à l'écran LCD d'exécuter une tâche prédéfinie telle que l'initialiser, effacer son écran, régler la position du curseur, contrôler l'affichage, etc. Le registre de données stocke les données à afficher sur l'écran LCD. Les données sont la valeur ASCII du caractère à afficher sur l'écran LCD. [41]

Tableau II. 4 : *explication des pins de LCD 16x2*

Pin No	Function	Name
1	Gnd (0V)	Gnd
2	Alimentation: 5V (4.7V – 5.3V)	Vcc
3	Réglage du contraste ; à travers une résistance variable	V_{EE}
4	Sélectionne le registre de commande quand bas ; et les données enregistrent quand haute	Register Selection
5	Bas pour écrire au registre ; Haut à lire dans le registre	Lire/écrire
6	Envoie des données aux broches de données lorsqu'une impulsion haute à basse est donnée	Activer
7	8-bit broches de données	DB0
8		DB1
9		DB2
10		DB3
11		DB4
12		DB5
13		DB6
14		DB7
15	rétro-éclairage V_{CC} (5V)	Led+
16	rétro-éclairage Gnd (0V)	Led-

- **Adaptateur LCD I2C**

Ces 16 caractères par 2 lignes d'affichage a un contraste texte blanc très clair et très haut sur un fond bleu / rétro-éclairage. Il comprend également une série I2C / IIC carte adaptateur pré-soudé à l'arrière de l'écran LCD. Cela signifie qu'il peut être contrôlé avec seulement 2 I2C broches de données série (SDA et SCL) et nécessite donc beaucoup moins broches numériques IO lorsqu'il est commandé à partir d'un microcontrôleur. Au total, le module ne nécessite 4 fils dont la puissance 5V et GND. Le réglage du contraste est également fourni par la carte fille par l'intermédiaire d'un potentiomètre. Si vous prévoyez d'utiliser avec une carte Arduino.[42]

Les pins de LCD (I2C) sont : GND, VCC (+5V), SDA et SCL.

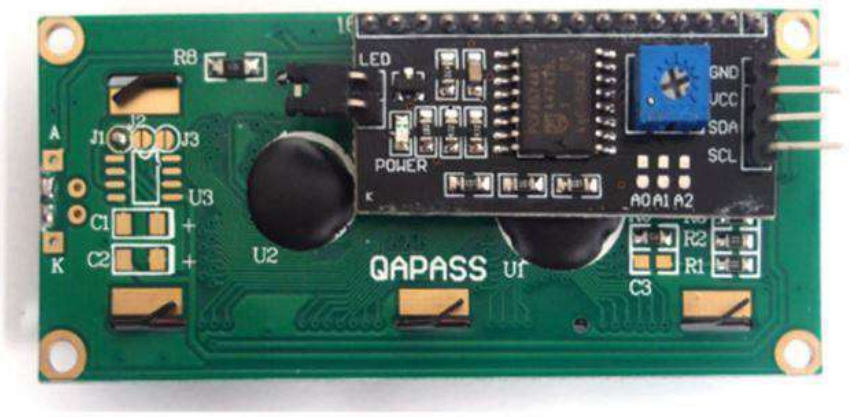


Figure II. 13: *L'Afficheur LCD avec l'adaptateur I2C*

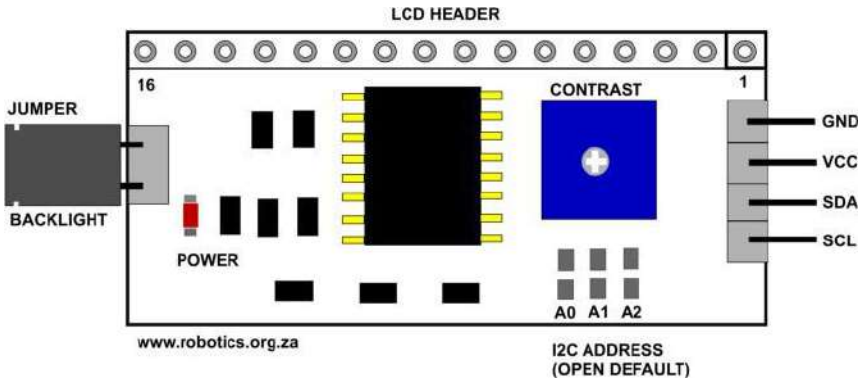


Figure II. 14: *Adaptation I2C*

II.2.6 Le servo-moteur (Sg90)

Les servos-moteurs sont des appareils intelligents. En utilisant une seule broche d'entrée. Sur un servo-moteur, l'angle de l'axe reste fixé dans une position et peut varier entre 0° et 180° En interne, ils ont un pilote de moteur et un circuit de retour qui s'assure que le bras de servo atteigne la position désirée.



Figure II. 15: *Le servo-moteur (Sg90)*

II.2.6.1 Fonctionnement

C'est une onde carrée semblable à PWM. Chaque cycle dans le signal dure 20 millisecondes et la plupart du temps, la valeur est BAS. Au début de chaque cycle, le signal est HAUT pendant une durée comprise entre 1 et 2 millisecondes. À 1 milliseconde, cela représente 0 degré et à 2 millisecondes, 180 degrés. Entre les deux, il représente la valeur comprise entre 0 et 180. C'est une méthode très bonne et fiable.

II.2.6.2 Spécifications du servo moteur

- La tension de fonctionnement est généralement de + 5V.
- Couple : 2.5kg / cm.
- La vitesse de fonctionnement est 0.1s / 60 °.
- Type d'engrenage : plastique.
- Rotation : 0 ° -180 °.
- Poids du moteur : 9gm.
- Le forfait comprend des cornes et des vis.

Les servomoteurs ont trois fils : alimentation, mise à la terre et signal.

- Le fil d'alimentation est généralement rouge et doit être connecté à la broche 5V de la carte Arduino.

- **Le fil de terre** est généralement noir ou marron et doit être connecté à une broche de terre sur la carte Arduino.

- La broche de signal est généralement jaune, orange ou blanche et doit être connectée à une broche numérique sur la carte Arduino

II.2.7 Capteur de lumière (HM-Fliying fish)

Les capteurs à photorésistance sont des capteurs qui vous permettent de détecter la lumière. Ils sont petits, peu coûteux, de faible puissance, faciles à utiliser et ne s'usent pas. Le module de capteur photosensible est le plus sensible pour l'intensité lumineuse de l'environnement et il est généralement utilisé pour détecter la luminosité ambiante et l'intensité de l'environnement.[43]



Figure II. 16: *Capteur de lumière*

II.2.7.1 Fonctionnement

Ce module de capteur de lumière basé sur le photodétecteur GL5528 permet de détecter l'intensité lumineuse de l'environnement. Comme la résistance du capteur varie en fonction de la quantité de lumière à laquelle il est exposé, la tension de sortie varie en fonction de l'intensité de la lumière. Ce module émet des signaux analogiques et numériques qui peuvent être utilisés pour déclencher d'autres modules. Le potentiomètre être utilisé pour ajuster la sensibilité de la sortie numérique. La sortie numérique du terminal est HAUTE lorsque l'intensité de la lumière dépasse la valeur définie par le potentiomètre, et inversement. Les sorties de terminal analogique augmentent avec l'intensité lumineuse.

Le résultat du bloc de lecture analogique sera un nombre compris entre 0 et 1023, où 0 représente 0V et 1023 représente 3V. Cette valeur est stockée dans la lecture de variable. Il est ensuite divisé en un nombre égal à 50. Vous pouvez modifier le nombre 50 pour rendre les lectures plus ou moins sensibles en fonction de la plage de lumière qui vous intéresse. [43]

II.2.7.2 Spécifications

- Utilisation d'un capteur photosensible sensible.
- Le signal de sortie du comparateur est propre, bonne forme d'onde, capacité de conduite, plus de 15mA.

- Avec potentiomètre réglable pour régler la luminosité de la détection de lumière.
- La tension de travail 3.3V-5V.
- Sortie numérique - 0V à 5V, niveau de déclenchement réglable à partir du préréglage.
- Sortie analogique - 0V à 5V basé sur la lumière tombant sur le LDR.
- La forme de sortie : sorties de commutation numériques DO (0 et 1) et sortie de tension analogique AO.
- Trous de boulon fixes pour une installation facile.
- Taille du circuit imprimé : 3.2cm x 1.4cm.
- Utilisation d'un comparateur de tension large LM393.

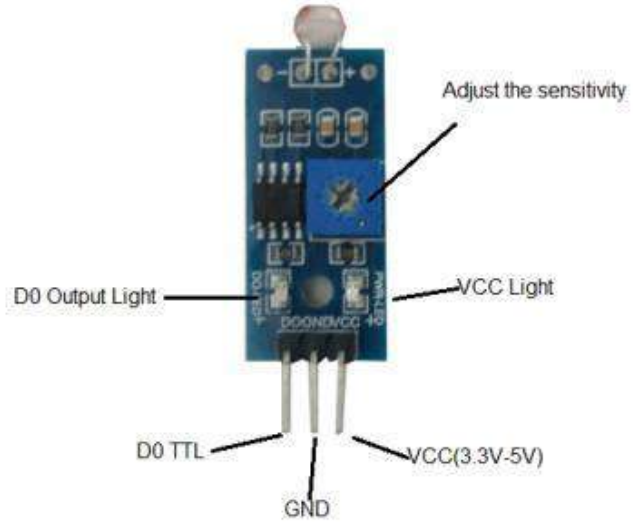


Figure II. 17 : *Explication du module*

II.2.8 Capteur de gaz (MQ-5)

Le module de capteur de gaz MQ5 est utile pour la détection des fuites de gaz, La sensibilité du ce capteur au propane et au méthane est assez élevé et le méthane et le propane peuvent être bien détectés. Ce capteur peut détecter une variété de gaz combustibles, en particulier de gaz naturel, ce qui en fait un capteur à faible coût pour diverses applications. Il a petite sensibilité à l'alcool et la fumée. En raison de sa sensibilité et de son temps de réponse élevés, les mesures peuvent être prises dès que possible. La sensibilité du capteur peut être ajustée à l'aide du potentiomètre.[44]

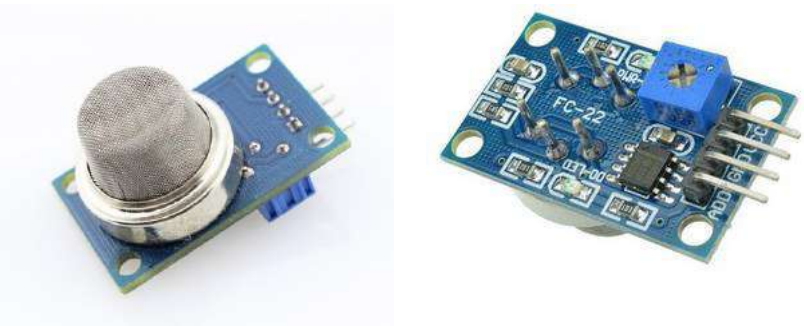


Figure II. 18: *Capteur de gaz (MQ-5)*

II.2.8.1 Fonctionnement

Le capteur de gaz MQ-5 applique du SnO2 qui présente une conductivité inférieure dans l'air clair en tant que matériau de détection de gaz. Dans une atmosphère où il peut y avoir des gaz inflammables, la conductivité du capteur de gaz augmente avec la concentration de gaz inflammable. MQ-5 joue un rôle de premier plan dans la détection du butane, du propane et du méthane et peut identifier à la fois le propane et le méthane. MQ-5 est très sensible au gaz naturel. Il se caractérise par sa capacité à détecter divers gaz inflammables et son coût réduit, ce qui en fait un choix idéal pour différentes applications de détection de gaz.Ce module offre deux possibilités de sortie : une sortie analogique (A0) et une sortie numérique (D0). La sortie analogique peut être utilisée pour détecter les fuites de gaz et pour mesurer le volume de fuites de gaz (en calculant correctement la sortie du capteur dans le programme) en unités spécifiques (par exemple, ppm). La sortie numérique ne donne que deux sorties possibles - Haute et Basse[44]

II.2.8.2 Les caractéristiques du module

- Haute sensibilité au GPL, au gaz naturel, au gaz de ville.
- Petite sensibilité à l'alcool, la fumée.
- Réponse rapide.
- Stable et longue vie.
- Circuit d'entraînement simple

II.2.8.3 Spécifications du module

- Puce de convertisseur PT1301.
- Concentration de détection : 300 à 10 000 ppm.
- Tension de fonctionnement 2.5V-5.0V.
- Courant de fonctionnement 150mA.

- Dimensions 40.0mm X 21.0mm.
- Taille de trou de fixation 2.0mm.
- DOUT Sortie numérique.
- AOUT Sortie analogique.
- GND la terre.
- VCC Alimentation positive.

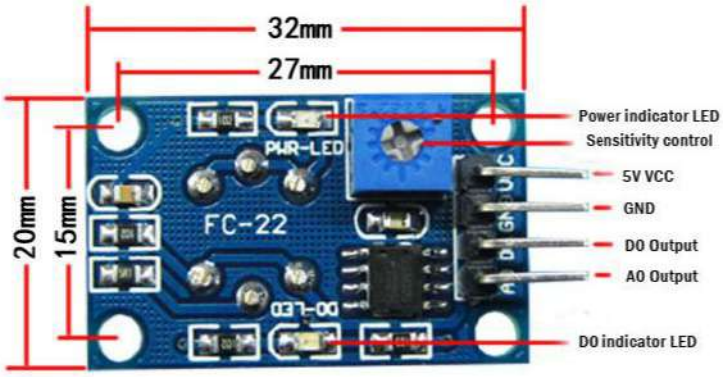


Figure II. 19: *Explication du module MQ5.*

II.2.9 Capteur de flamme (flying Fish)

Le module de capteur de détection de flamme est sensible à la flamme, mais peut également détecter une lumière ordinaire. Habituellement utilisé comme alarme de flamme. Détecte une flamme ou une source de lumière d'une longueur d'onde comprise entre 760 et 1100 nm. Point de détection d'environ 60 degrés, particulièrement sensible au spectre de la flamme. La sensibilité est réglable, performance stable. Le capteur peut être un atout majeur dans un robot de lutte contre l'incendie, il peut être utilisé comme un œil de robot pour trouver la source d'incendie[45]

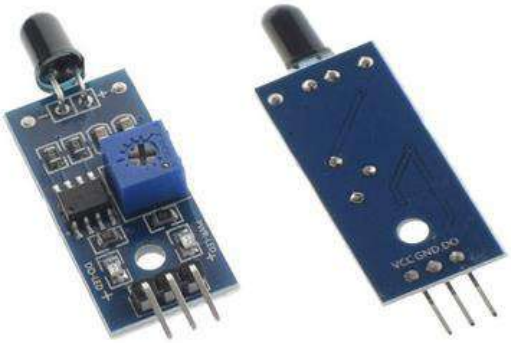


Figure II. 20: *Capteur de flamme*

II.2.9.1 Fonctionnement

Ce capteur de flamme peut être utilisé pour détecter une source d'incendie ou d'autres sources lumineuses de la longueur d'onde comprise entre 760 et 1100 nm. Il est basé sur le capteur YG1006 qui est un phototransistor au silicium NPN à haute vitesse et haute sensibilité. En raison de son époxy noir, le capteur est sensible au rayonnement infrarouge. Lorsque le capteur détecte une flamme, le voyant Signal s'allume et la broche D0 devient BAS. Vous pouvez contrôler la sensibilité en ajustant le potentiomètre. Un test de flamme plus légère peut être déclenché dans un rayon de 0,8 m; si l'intensité de la flamme est élevée, la distance de détection sera augmentée.

Ce capteur ne montre pas les valeurs absolues (telles que la température exacte en ° C ou la force du champ magnétique en mT). C'est une mesure relative : vous définissez une valeur extrême pour une situation environnementale normale donnée et un signal sera envoyé si la mesure dépasse la valeur extrême.[45]

II.2.9.2 Spécifications

- Tension de travail : 3.3v - 5v.
- Plage de détection : 60 degrés.
- Sorties de commutation numériques (0 et 1).
- Avec un trou de vis de montage PCB taille : 3cm x 1.6cm.
- Comparateur puce LM393.
- Sensibilité photo élevée.
- Temps de réponse rapide.
- Sensibilité réglable

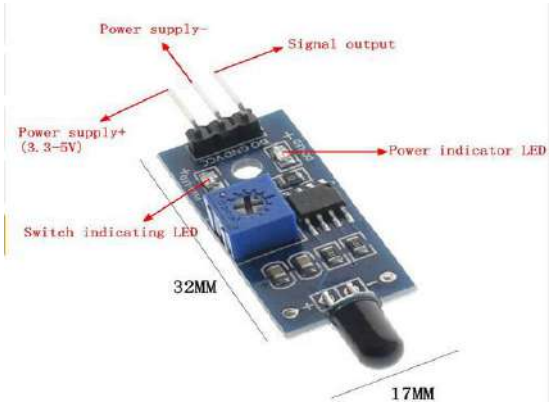


Figure II. 21: *Explication du module de flamme*

II.2.10 Le capteur de température et humidité DHT-22 :

II.2.10.1 Définition

Le DHT-22 (également connu par RHT03) est un capteur d'humidité et de température à faible coût doté d'une interface numérique à un fil. Le capteur est calibré et ne nécessite pas de composants supplémentaires, ce qui vous permet de mesurer directement l'humidité relative et la température. Le DHT22 est plus précis et a une plage dynamique plus grande que ses capteur DHT11.[46]

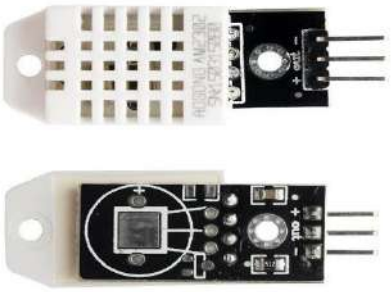


Figure II. 22: *Le capteur DHT-22*

II.2.10.2 Spécifications Techniques

- Entrée 3.3-5.5V
- Courant de mesure 1-1,5mA
- Courant de veille 40-50 uA
- Humidité de 0-100% HR
- Plage de température -40 - 80 degrés C
- Précision de + -2% HR
- + -0,5 degrés C

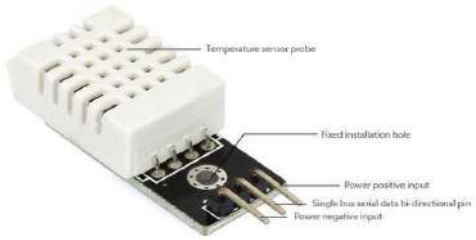


Figure II.11 : *Le capteur DHT-22 schématique.*

II.2.11 Le capteur ultrasonic HC-SR04

II.2.11.1 Introduction

Le capteur à ultrasons envoie une impulsion sonore haute fréquence, puis le temps nécessaire pour que l'écho du son se réfléchisse. Le capteur a 2 ouvertures sur le devant. Une ouverture transmet des ondes ultrasonores, l'autre les reçoit.[47]

II.2.11.2 Les caractéristiques du module

- Alimentation : + 5V DC.
- Courant de repos : <2mA.
- Courant de travail : 15mA.
- Angle effectif : <15 °.
- Distance : 2cm - 400 cm / 1 ″ - 13ft.
- Résolution : 0.3 cm.
- Angle de mesure : 30 degrés.
- Largeur d'impulsion d'entrée de déclenchement : 10uS
- Dimension : 45mm x 20mm x 15mm.

II.2.11.3 Fonctionnement

Le capteur à ultrasons HC-SR04 utilise un sonar pour déterminer la distance à un objet comme le font les chauves-souris. Il offre une excellente détection de plage sans contact avec une

précision élevée et des lectures stables dans un boîtier facile à utiliser. De 2 cm à 400 cm ou 1 ”à 13 pieds. Son fonctionnement n’est pas affecté par la lumière du soleil ou un matériau noir comme les télémètres Sharp (bien que des matériaux acoustiquement doux comme le tissu puissent être difficiles à détecter). Il vient complet avec module émetteur et récepteur à ultrasons.[47]

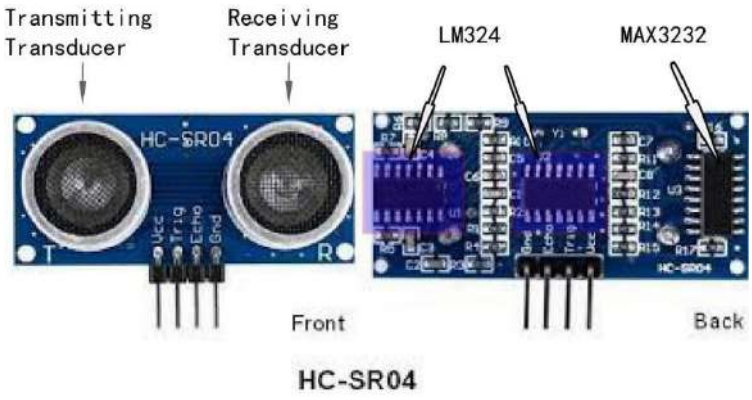


Figure II.12 : Explication du module *HC-SR04*

À l'avant du télémètre à ultrasons se trouvent deux cylindres en métal. Ce sont des transducteurs. Les transducteurs convertissent les forces mécaniques en signaux électriques. Dans le télémètre à ultrasons, il existe un transducteur émetteur et un transducteur récepteur. Le transducteur émetteur convertit un signal électrique en impulsion ultrasonore et le transducteur récepteur reconvertit l'impulsion ultrasonore réfléchie en un signal électrique. Si vous regardez à l'arrière du télémètre, vous verrez un CI derrière le transducteur de transmission appelé MAX3232. C'est l'IC qui contrôle le transducteur émetteur. Derrière le transducteur récepteur se trouve un circuit intégré appelé LM324. Il s'agit d'un ampli-op à quatre qui amplifie le signal généré par le transducteur récepteur en un signal suffisamment puissant pour être transmis à l'Arduino.[47]

- **Chronogramme de fonctionnement :**

Le chronogramme de HC-SR04 est présenté. Pour démarrer la mesure, Trig de SR04 doit recevoir une impulsion haute (5V) pendant au moins 10us. Le capteur émettra 8 cycles de rafale ultrasonore à 40 kHz et attendra la rafale ultrasonore réfléchie. Lorsque le capteur détecte les ultrasons depuis le récepteur, la broche Echo est réglée sur haut (5 V) et retarde pendant une période (largeur) proportionnelle à la distance. Pour obtenir la distance, mesurez la largeur (Ton) de la broche Echo.[47]

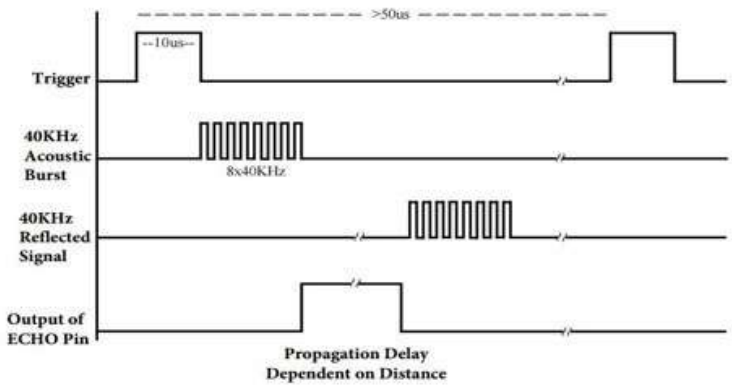


Figure II. 23*: Le chronogramme de HC-SR04*

- **Comment calculer la distance :**

- Temps = largeur d'impulsion d'écho, en nous (micro seconde)
- Distance en centimètres = Temps / 58
- Distance en pouces = Temps / 148

Vous pouvez également utiliser la vitesse du son, car il est connu que le son voyage dans les airs à environ 344 m / s (1129 ft / s), vous pouvez prendre le temps que l'onde sonore revienne et la multiplier par 344 mètres (ou 1129 pieds) pour trouver la distance aller-retour totale de l'onde sonore. Aller-retour signifie que l'onde sonore a parcouru 2 fois la distance de l'objet avant d'être détectée par le capteur ; il inclut le "trajet" du capteur sonar à l'objet ET le "trajet" de l'objet au capteur à ultrasons (après que l'onde sonore ait rebondi sur l'objet). Pour trouver la distance à l'objet, divisez simplement la distance aller-retour en deux.[47]

$$distance = \frac{la\ vitess\ de\ son \text{ X le temp passé}}{2}$$

La variable de temps est le temps nécessaire à l'impulsion ultrasonore pour quitter le capteur, rebondir sur l'objet et revenir au capteur. En fait, nous divisons ce temps par deux puisqu'il ne nous reste plus qu'à mesurer la distance à l'objet, et non la distance à l'objet et au capteur. La variable de vitesse est la vitesse à laquelle le son se déplace dans l'air.[47]

II.2.12 Le clavier 4x4 (keypad)

Un clavier est un ensemble de boutons disposés dans un bloc ou un « pavé » qui porte des chiffres, des symboles ou des lettres alphabétiques. Les pads contenant principalement des chiffres sont appelés un clavier numérique.

Le clavier à matrice 4 x 4 est généralement utilisé comme entrée dans un projet. Il a 16 clés au total, ce qui signifie les mêmes valeurs d'entrée. Il est ultra-fin, facile à connecter à n'importe quel microcontrôleur et dispose d'un endos adhésif pour faciliter le montage dans diverses applications.[48]



Figure II. 24: Le clavier 4x4

II.2.12.1 Les caractéristiques

- Design ultra-mince.
- Support adhésif.
- Excellent rapport qualité / prix.
- Interface facile à n'importe quel microcontrôleur.
- Exemples de programmes fournis avec les microcontrôleurs BASIC Stamp 2 et Propeller P8X32A.
- Estimation maximale : 24 VDC, 30 mA.
- Interface : accès à 8 broches à matrice 4 × 4.
- Température de fonctionnement : 0 à 50 ° C (32 à 122 ° F).
- Dimensions : clavier, 6,9 x 7,6 cm (2,7 x 3,0 po) Câble : 2,0 x 8,8 cm (0,78 x 3,5 po).

II.2.12.2 Fonctionnement

Les claviers Matrix utilisent une combinaison de quatre rangées et de quatre colonnes pour fournir des états de bouton au périphérique hôte, généralement un microcontrôleur. Sous chaque touche se trouve un bouton-poussoir, une extrémité étant connectée à une rangée et l'autre extrémité à une colonne.

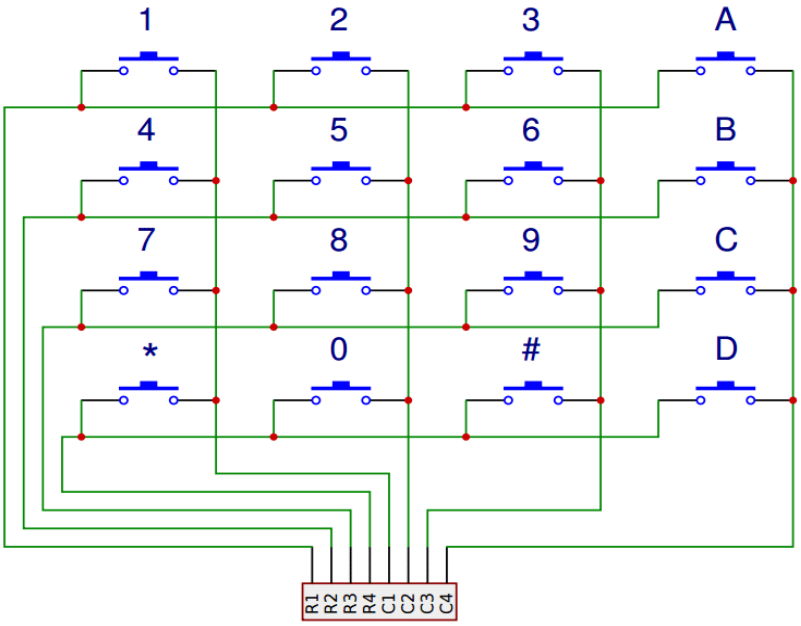


Figure II.15 : *Le circuit intégré du clavier 4x4*

Pour que le microcontrôleur détermine quel bouton est enfoncé, il doit d'abord tirer sur chacune des quatre colonnes (broches 1 à 4), le plus faible ou le plus élevé, puis interroger les états des quatre rangées (broches 5 et 5). 8). En fonction de l'état des colonnes, le microcontrôleur peut indiquer quel bouton est enfoncé. Par exemple, supposons que votre programme élève les quatre colonnes sur le bas, puis la 2eme rangée plus haut. Il lit ensuite les états d'entrée de chaque colonne et lit la broche 3 à la hauteur. Cela signifie qu'un contact a été établi entre la colonne 2 et la rangée 2, le bouton « 5 » a donc été actionné.[48]

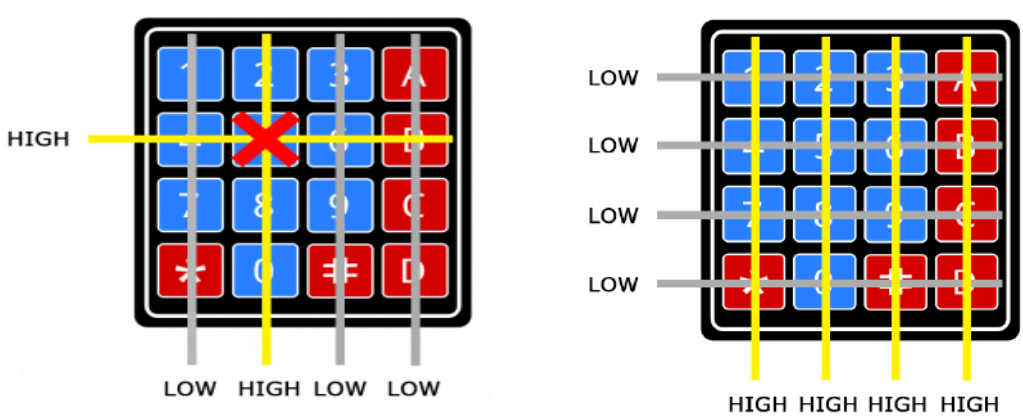


Figure II. 25: *Fonctionnement du clavier 4x4*

II.2.13 Capteur de mouvement (PIR)

Les capteurs de mouvement PIR vous permettent de détecter un mouvement, presque toujours utilisé pour détecter si un humain est entré ou non dans la plage de capteurs. Ils sont petits, peu coûteux, de faible puissance, faciles à utiliser et ne s'usent pas. Pour cette raison, ils se trouvent couramment dans les appareils et les gadgets utilisés dans les maisons ou les entreprises. Ils sont souvent appelés capteurs PIR, « Infrarouge passif », « Pyroélectrique » ou « Mouvement IR ». [49]



Figure II. 26: *Capteur de mouvement (PIR).*

II.2.13.1 Caractéristique du module

- Tension de travail : 4.5V à 20V
- Sortie : haute : 3,3 V, basse : 0 V
- Angle de détection : environ 120 degrés
- Portée : réglable, jusqu'à 7m
- Modes de déclenchement : déclencheur non répétable / déclencheur répétable H (par défaut)

- Temps de pause : (temps restant) réglable entre 5-300 secondes. — il peut être encore augmenté en augmentant la valeur du condensateur CY1-Timing sur la broche 4 du CI
- Température de fonctionnement : -20 - +80 degrés C.
- Dimensions de la carte : 33x25mm, 14mm de haut sans l'objectif ; Objectif : 11 mm de haut, 23 mm de diamètre.
- Poids : 6g

II.2.13.2 Fonctionnement :

Capteur de mouvement PIR. PIR signifie InfraRed Passif. Ce capteur de mouvement comprend une lentille de Fresnel, un détecteur infrarouge et des circuits de détection associés. La lentille du capteur concentre tout rayonnement infrarouge présent autour du détecteur infrarouge. Notre corps produit de la chaleur infrarouge qui est captée par le capteur de mouvement. Le capteur émet un signal 5V pendant une minute dès qu'il détecte la présence d'une personne. Il offre une plage de détection provisoire d'environ 6 à 7 m et est très sensible. Lorsque le détecteur de mouvement PIR détecte une personne, il envoie un signal 5V à l'Arduino. Ainsi, une interruption sur Arduino est déclenchée. Nous définissons ce que l'Arduino doit faire lorsqu'il détecte un intrus. [49]

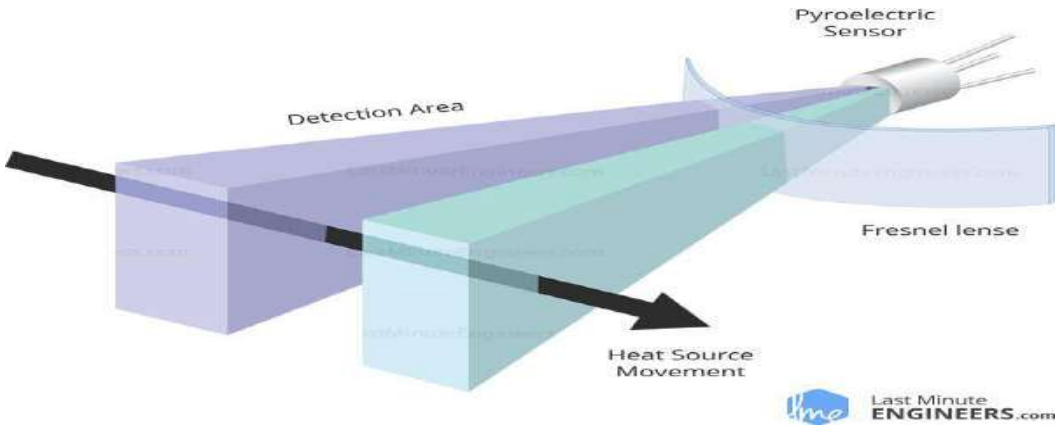


Figure II. 27: *Fonctionnement de l'infrarouge.*

- **Sensibilité :**

Comme mentionné précédemment, la plage de réglage est comprise entre 3 et 7 mètres environ. L'illustration ci-dessous montre ce réglage. Vous pouvez cliquer pour agrandir l'illustration.

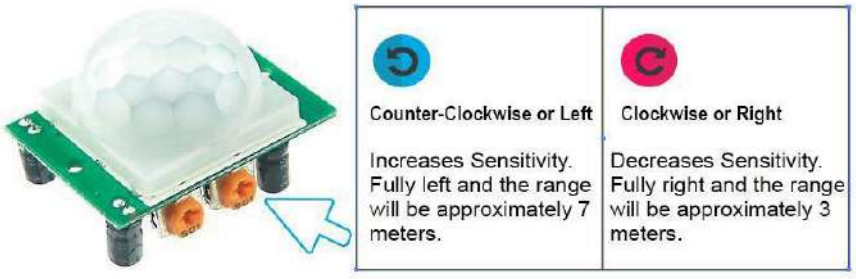


Figure II. 28: *Réglage de sensibilité*

- **Temps :**

Le réglage de la temporisation détermine la durée pendant laquelle la sortie du module de capteur PIR restera élevée après le mouvement de détection. La plage est d'environ 5 secondes à cinq minutes. L'illustration ci-dessous montre ce réglage.[49]



Figure II. 29: *Réglage du temp*

II.2.14 Buzzer et Ventilateur

- **Le buzzer**

Il s'agit d'une petite enceinte ronde de 12 mm qui fonctionne autour de la plage audible de 2 kHz. Vous pouvez utiliser ces haut-parleurs pour créer de la musique simple ou des interfaces utilisateur. Ce n'est pas un vrai haut-parleur piézoélectrique mais se comporte de la même manière. Au lieu d'un cristal piézoélectrique qui vibre avec un courant électrique, ce minuscule haut-parleur utilise un électroaimant pour entraîner une mince feuille de métal.

Cela signifie que vous devez utiliser une forme de courant alternatif pour obtenir un son. La bonne nouvelle est que ce haut-parleur est réglé pour répondre au mieux avec une onde carrée (par exemple à partir d'un microcontrôleur).[50]

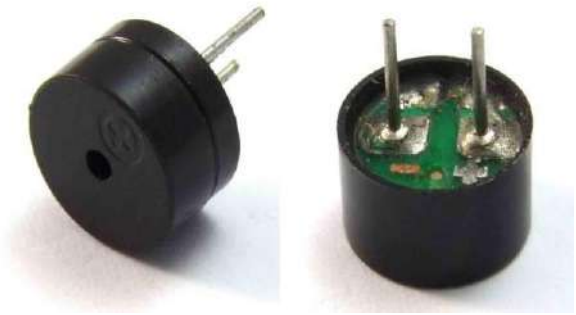


Figure II. 30: *le buzzer*

II.2.14.1. Les caractéristiques du buzzer

- Type de prise de son : son continu.
- Tension nominale : DC 5V.
- TENSION DE FONCTIONNEMENT : DC 4 ~ 7V.
- Taille du corps (borne non comprise) : 12 x 9,5 mm / 0,47 "x 0,37" (D * T).
- Hauteur du terminal : 7,6 mm / 0,29 ″.
- Sortie audio : ≥85dB.
- Matériau externe : plastique.
- Fréquence : 2300 +/- 300Hz.
- Courant : 30mA.
- Température de fonctionnement : -20 ~ + 85.
- La couleur noire.
- Poids : 8g.
- Dimension : (UNIT : mm).[25]

➢ **La ventilateur 12v**

- Débit d'air maximal 49.7CFM.
- Maximum noise 44dBA.
- Température de fonctionnement -10 ~ 75 degrés C.

- Vitesse nominale 4000 - 5000RPM.
- Tension nominale 12 V.
- Humidité relative 65% +/- 20%.



Figure II. 31: *La ventilateur 12v.*

II.3 Logiciel

II.3.1 Arduino IDE

IDE signifie « Integrated Development Environment » ou en bon français Environnement de Développement Intégré. Cette application propose effectivement tout ce qui est nécessaire pour éditer un programme, vérifier sa syntaxe, puis le téléverser dans une carte Arduino, c'est-à-dire programmer le microcontrôleur de la carte. Très simple à utiliser, l'IDE est très facile à prendre en main en quelques minutes du fait de sa grande convivialité, et l'apprentissage d'un débutant se fait naturellement sans qu'il soit nécessaire de lire et comprendre à l'avance une quelconque documentation. Cependant, l'IDE offre tout de même de plus en plus de possibilités au fur et à mesure de ses évolutions. Cet explication fait le point sur certaines options qui peuvent se révéler bien utiles. Les figures et les explications sont faites avec un environnement Windows, mais ceux qui travaillent sous Mac ou Linux transposeront aisément car il n'y a quasiment pas de différence.[51]

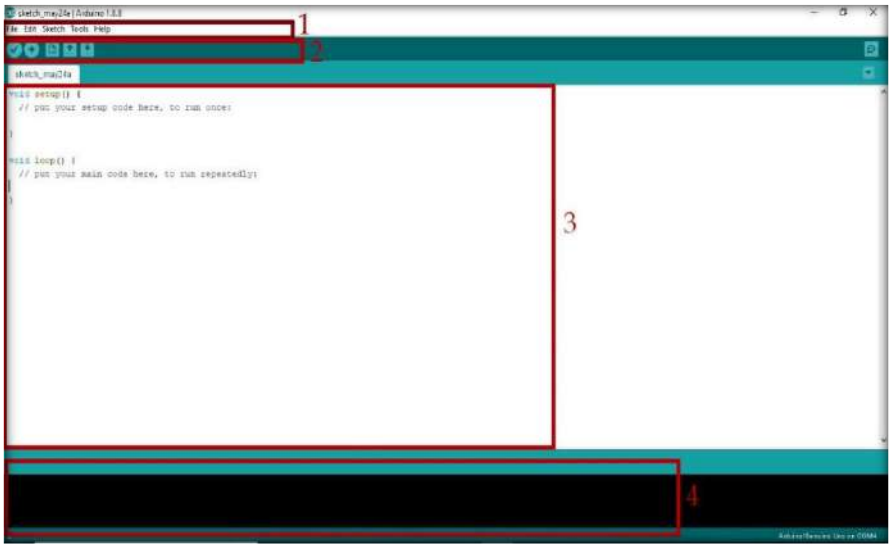


Figure II. 32: *L'interface du logiciel IDE.*

1 → ce sont les options de configuration du logiciel.

2 → il contient les boutons qui vont nous servir lorsque l'on va programmer nos cartes.

3 → ce bloc va contenir le programme que nous allons créer.

4 → celui-ci est important, car il va nous aider à corriger les fautes dans notre programme.

C'est le débogueur.



Figure II. 33: *La barre des tâches.*

1 → Ce bouton permet de vérifier le programme, il actionne un module qui cherche les erreurs dans votre programme.

2 → Charge (téléverse) le programme dans la carte Arduino.

3 → Crée un nouveau fichier.

4 → Ouvre un fichier.

5 → Enregistre le fichier.

6 → Ouvre le moniteur série.[52]

II.3.2 Proteus professionnel 8

II.3.2.1 Définition

Proteus 8 est un logiciel de conception de cartes de circuits électroniques, d'assemblage de cartes de circuits imprimés et de prototypage qui peut également être utilisé pour la simulation en temps réel de microcontrôleurs tels que Arduino, la conception de schémas électroniques et de circuits électriques externes et la conception de cartes de circuits imprimés.

Se compose de deux parties importantes ISIS et ARES [53]

II.3.2.2 Proteus ISIS

ISIS est le logiciel utilisé pour dessiner des schémas et simuler les circuits en temps réel. La simulation permet un accès humain pendant l'exécution, offrant ainsi une simulation en temps réel. ISIS possède une large gamme de composants dans sa bibliothèque. Il possède des sources, des générateurs de signaux, des outils de mesure et d'analyse tels qu'oscilloscope, voltmètre, ampèremètre, etc., des sondes pour la surveillance en temps réel des paramètres du circuit, des commutateurs, des affichages, des charges telles que des moteurs et des lampes, des composants discrets tels que des résistances, des condensateurs et des inductances, transformateurs, circuits intégrés numériques et analogiques, commutateurs à semi-conducteurs, relais, microcontrôleurs, processeurs, capteurs, etc…[53]

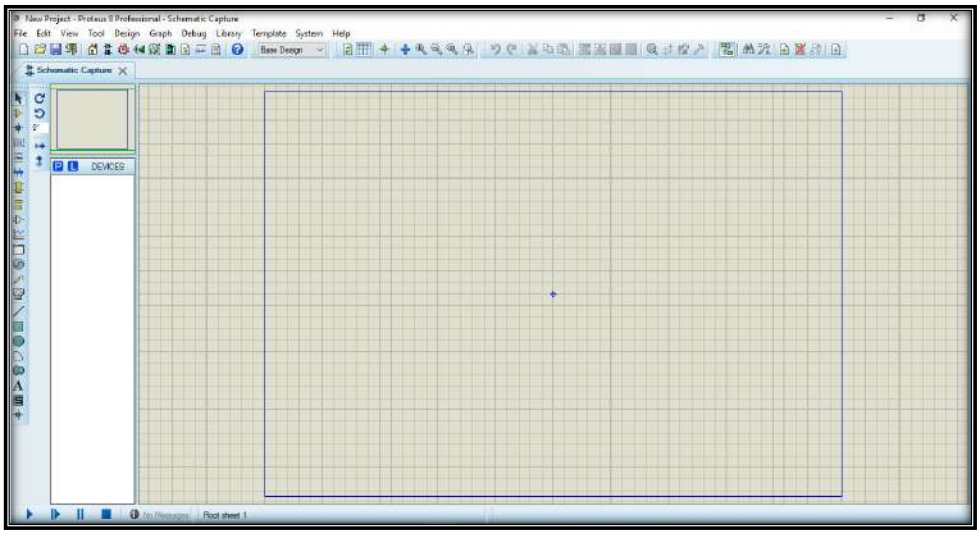


Figure II. 34: *L'interface du ISIS.*

II.3.2.3 Proteus ARES

ARES a est utilisé pour la conception de circuits imprimés. Il offre la possibilité de visualiser la sortie en 3D du circuit conçu ainsi que des composants. ARES propose des circuits imprimés pour la conception de jusqu'à 14 couches internes, avec montage en surface et paquets traversants. Il est intégré aux empreintes de pas de différentes catégories de composants tels que les circuits intégrés, les transistors, les en-têtes, les connecteurs et autres composants discrets. Il offre des options de routage automatique et manuel au PCB designer. Le schéma dessiné dans ISIS peut être transféré directement ARES.[53]

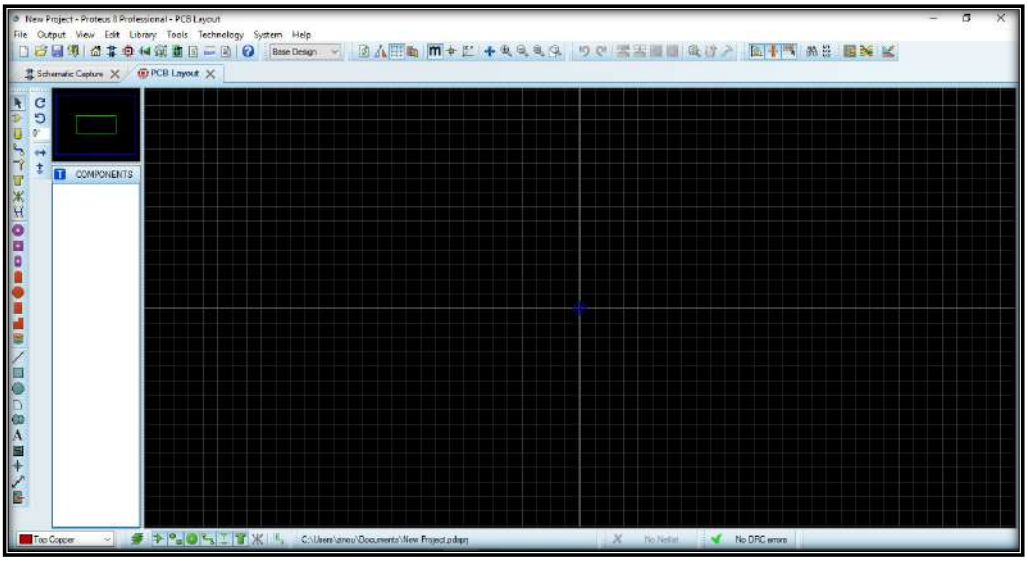


Figure II. 35: *L'interface du ARES.*

II.3.3 MIT App Inventor

MIT App Inventor est un environnement de programmation visuel et intuitif qui permet à tout le monde, de créer des applications entièrement fonctionnelles pour smartphones et tablettes. Les nouveaux utilisateurs de MIT App Inventor peuvent avoir une première application simple opérationnelle en moins de 30 minutes. De plus, notre outil basé sur des blocs facilite la création d'applications complexes et à fort impact en beaucoup moins de temps que les environnements de programmation traditionnels. [54]

Le projet MIT App Inventor vise à démocratiser le développement de logiciels en permettant à tous, en particulier les jeunes, de passer de la consommation de technologies à la création de technologies. Pour ceux qui souhaitent développer des applications Android, App

Inventor pour Android de MIT est le meilleur moyen de le faire sans connaissances en programmation. Comme il dispose déjà d'une interface Web pour la conception d'applications, les utilisateurs n'ont pas besoin d'entrer dans les spécificités de la programmation Java avec le kit de développement logiciel. Cela incitera d'autres établissements d'enseignement à utiliser l'open source comme moyen de créer des applications Android. À son tour, cela suscitera l'intérêt d'étudiants pour le développement d'applications Android.[54]

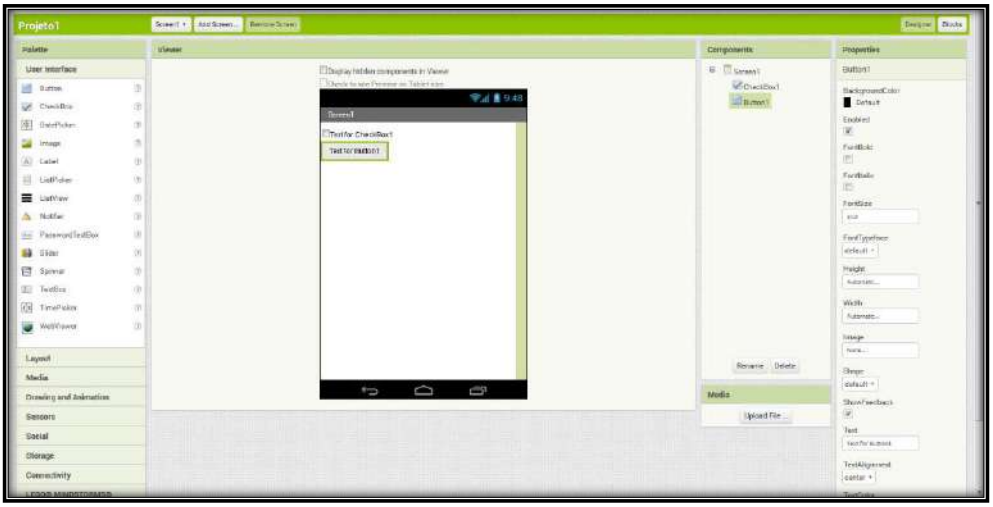


Figure II. 36: *L'interface du MIT.*

II.4 Conclusion

Après avoir parlé sur le matériel et les logiciels utilisés dans ce projet de la domotique, on va montrer comment les utiliser dans notre maison intelligente dans le chapitre suivant.


Chapitre III
la réalisation final

III.1 Introduction

Dans ce chapitre nous exposons en détail l'aspect réalisation de notre maison intelligente. La chronologie sera la suivante :

Dans un premier temps, la maison intelligente est décrite du point de vue architectural avec la disposition des périphériques dans les différents compartiments de la maison. L'aspect programmation de chaque module séparé est présenté par des organigrammes ; pour ensuite présenter la totalité du système dans un organigramme unique. La réalisation du circuit imprimé est ensuite illustrée dans tous ses aspects. Le chapitre est enfin terminé par une conclusion.

III.1.1 Description de périphérique et leur position

A. Le garage

Pour le garage, une antenne RFID est installée pour la détection d'une carte RFID qui a été mise sur le véhicule ; un servomoteur ouvre le garage quand Le véhicule couvre le champ de l'antenne RFID. La porte s'ouvrira donc automatiquement et après que le véhicule pénètre dans le garage la porte se referme automatiquement.

B. Le jardin

On a conçu trois systèmes pour le jardin, dont le premier est le système de sécurité connecté à un capteur ultrasonique installé à proximité de la porte d'entrée qui va émettre des ondes ultrasoniques jusqu'au deuxième mur de la maison, en cas d'intrusion il déclenche le buzzer et la LED rouge, il est aussi possible d'activer ou de désactiver ce système-là.

Le deuxième est le système d'éclairage automatique composé d'un capteur qui mesure la quantité de la lumière externe pour allumer la LED pendant la nuit et l'éteindre durant la journée.

Le troisième est aussi un système de sécurité sous forme de clavier 4X4 contenant des numéros de 0 à 9 et des lettres ABCD, en tapant un code correct la porte s'ouvre et se referme automatiquement.

C. Le couloir

Pour le couloir de la maison on a créé deux systèmes, l'un est le détecteur de mouvement basé sur l'infrarouge, lorsqu'un mouvement est détecté, le capteur met la ligne électrique sous tension ce qui allume la lumière et quand la personne quitte cette zone la lumière sera éteinte après un lapse de temps, dans le but d'économiser l'énergie et lutter contre le gaspillage.

L'autre système est composé d'un capteur de température, un afficheur LCD et un ventilateur. Le capteur mesure la température et l'humidité ensuite les affiche sur l'afficheur LCD, si la température dépasse notre degré de chaleur désiré qui a été déjà réglé dans notre système le ventilateur commence à tourner afin de nous offrir une climatisation.

D. Les chambres

Pour chaque chambre il y a une LED qui nous permet d'allumer et d'éteindre la lumière à distance avec une application Android sur Smartphone ou bien tablette.

E. La salle de bain

Dans la salle de bain on a un capteur de mouvement qui allume la lumière dès la détection d'un mouvement, et en quittant la salle de bain la lumière sera éteinte automatiquement après une période de temps.

F. La cuisine

Dans notre cuisine, on a réalisé deux systèmes de sécurité afin de se protéger contre deux dangers mortels, le premier système est le détecteur de fuite du gaz et le deuxième système le détecteur de feu.

Concernant le détecteur de fuite de gaz il est basé sur le capteur de gaz mq5, une LED rouge, un buzzer et un ventilateur. Le capteur mesure le pourcentage de gaz dans l'air et s'il atteint le niveau de danger, L'alarme se lance et la LED rouge commence à clignoter et en même temps l'extracteur dégage le gaz perdu à l'extérieur, et il va aérer la cuisine.Et concernant le système de flamme, il contient un capteur de flamme qui peut détecter le feu rapidement, le buzzer se déclenche et une LED rouge commence à clignoter pour nous avertir qu'il y a un danger.

G. Le balcon

Au niveau de notre balcon il y a un capteur de luminosité intégré à l'enrouleur de rideau, quand la luminosité est trop élevée l'appareil s'active automatiquement pour ouvrir les rideaux et lorsque la luminosité est insuffisante ou bien il fait nuit les rideaux se ferment automatiquement, avec la possibilité d'activation ou de désactivation de ce système par un seul bouton.

III.1.2 La programmation

Nous allons expliquer la partie programmation de chaque système par un organigramme et à la fin on va montrer l'organigramme général.

III.1.2.1 Le système d'économie d'énergie

Il est illustré sur l'organigramme de la figure III.1

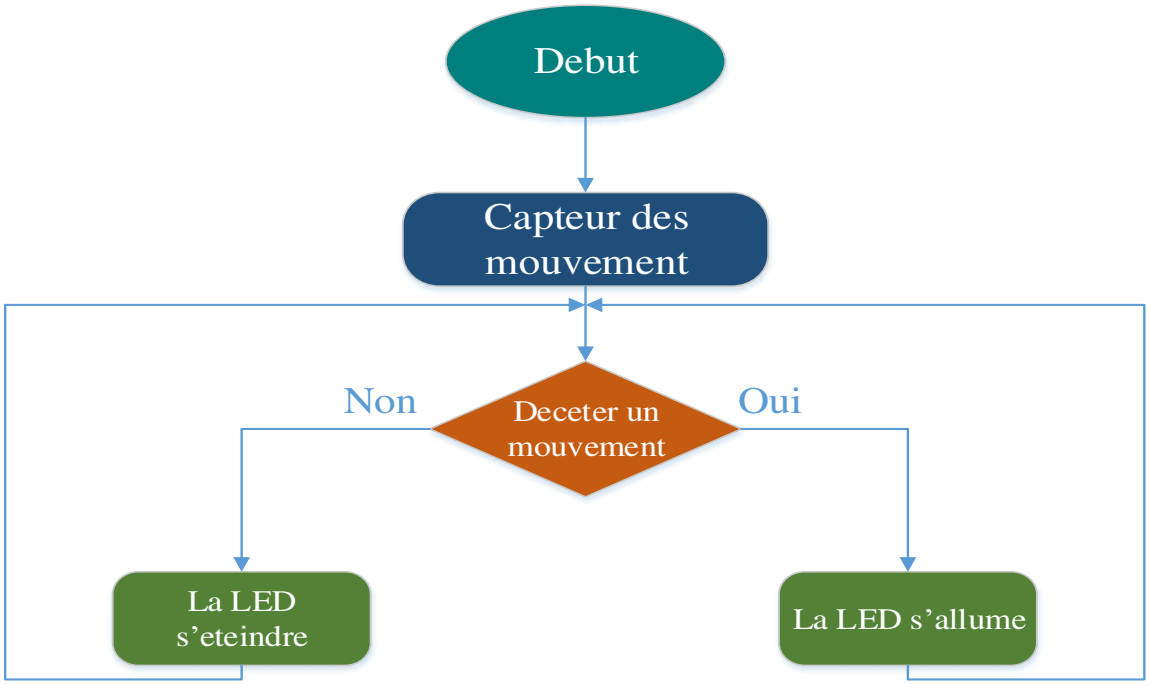


Figure III. 1: *L'organigramme de capteur de mouvement (PIR)*

III.1.2.2 Les systèmes de sécurité externe :

Ces systèmes ont pour mission d'assurer la sécurité contre le vol et l'infraction du domicile. Ils sont illustrés dans les organigrammes des figures III.2 et III.3.

A. Porte d'entrée

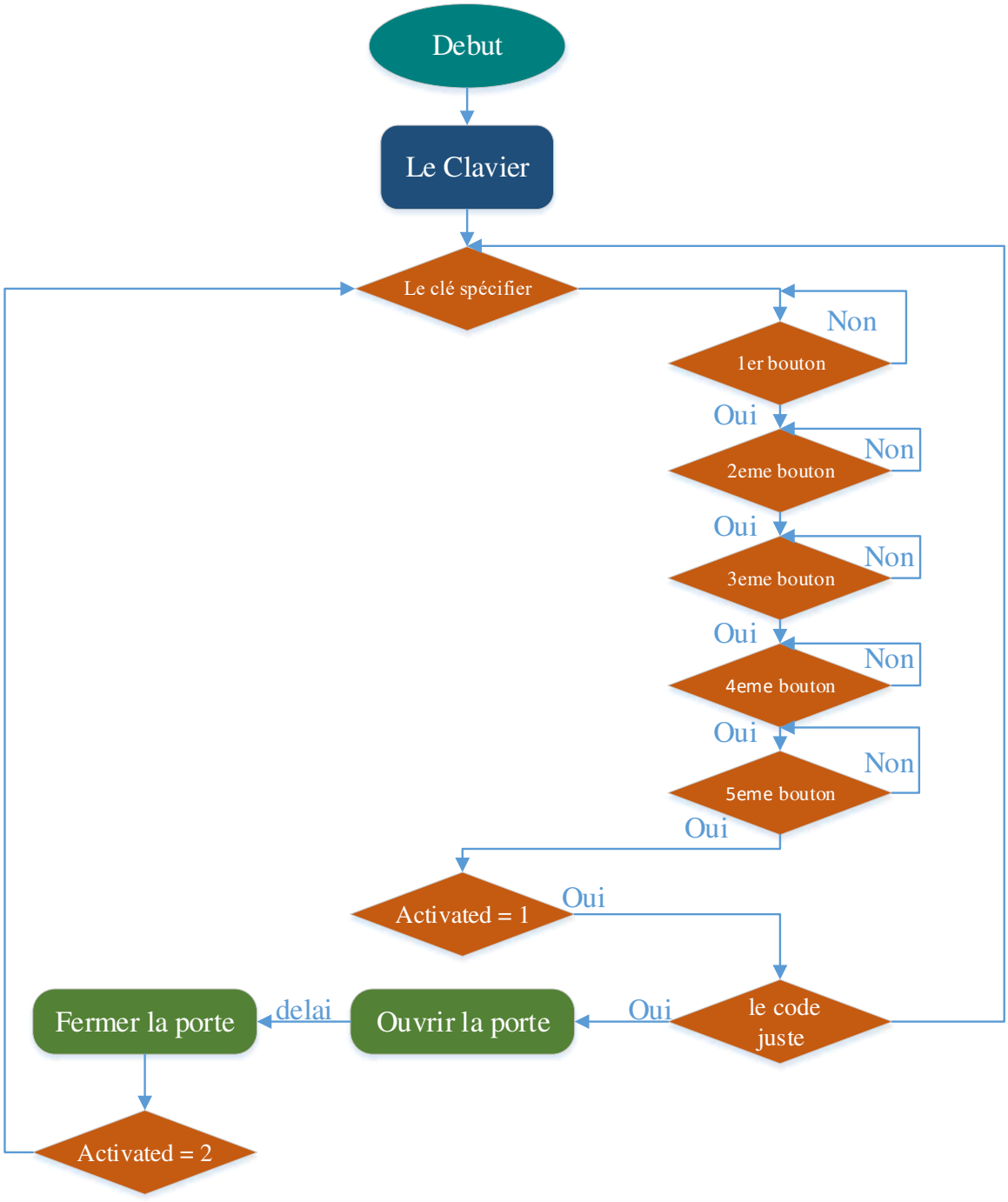


Figure III. 2: *L'organigramme de système d'ouverture et fermeture de la porte.*

B. Système d'alarme

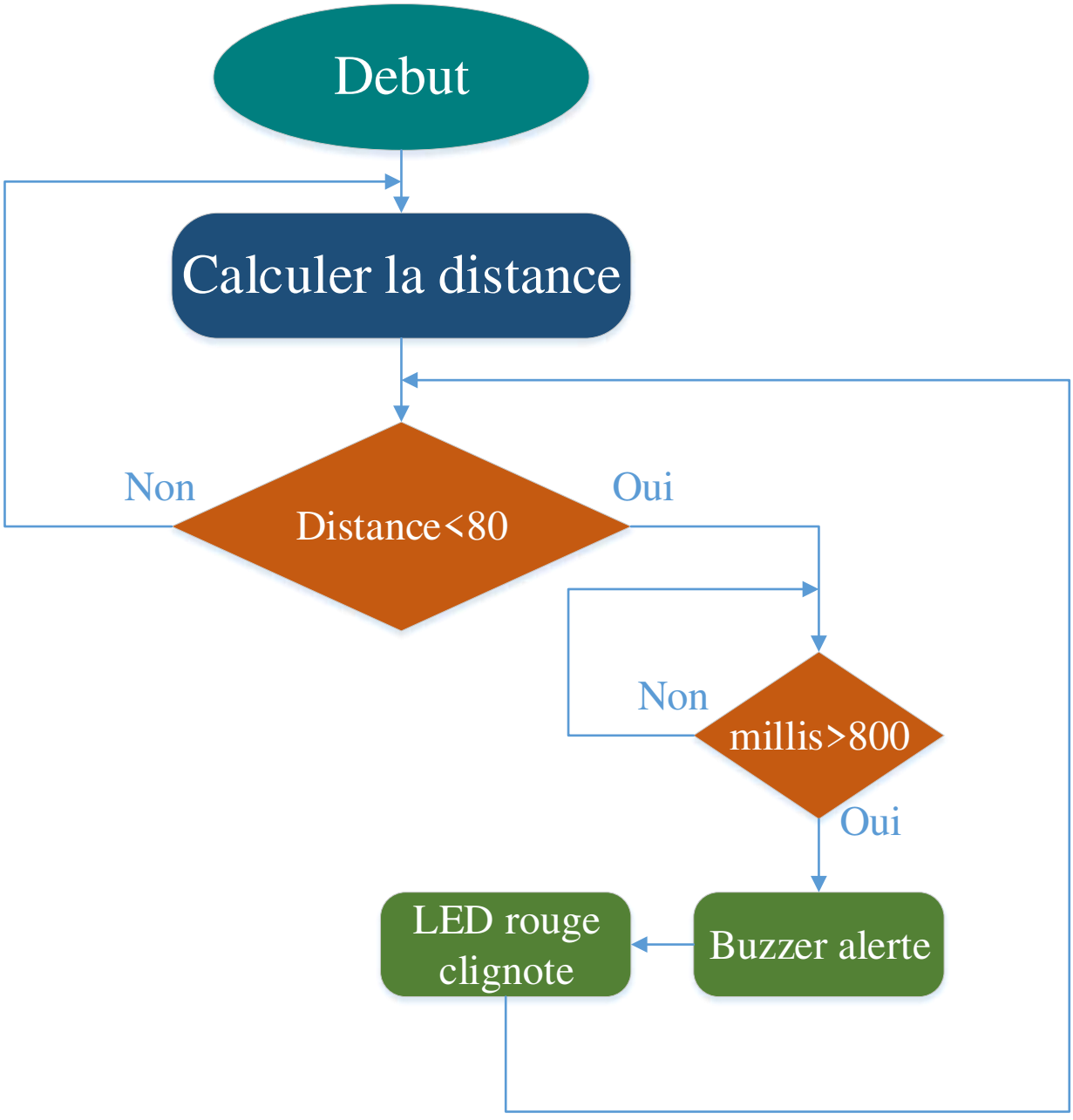


Figure III. 3: *L'organigramme du capteur ultrasonique HC-SR04.*

III.1.2.3 Système de température et climatisation

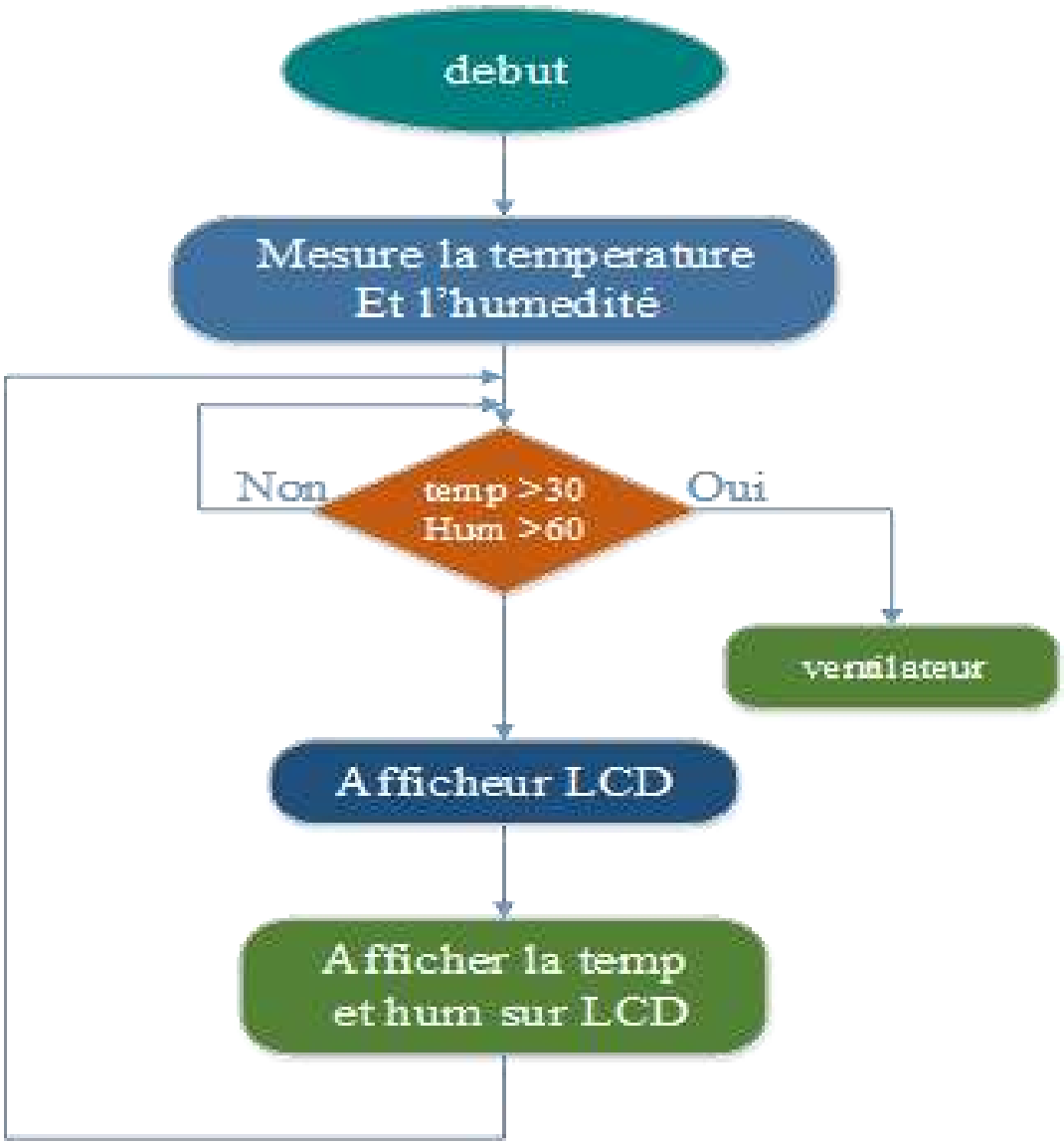


Figure III. 4: *L'organigramme de capteur DHT.*

III.1.2.4 Les systèmes de sécurité interne

Ces systèmes ont pour mission d'assurer la sécurité contre les dangers de flamme et les fuites de gaz. Ils sont illustrés sur les organigrammes des figures III.5 et III.6.

A. Système de détection de fuite de gaz

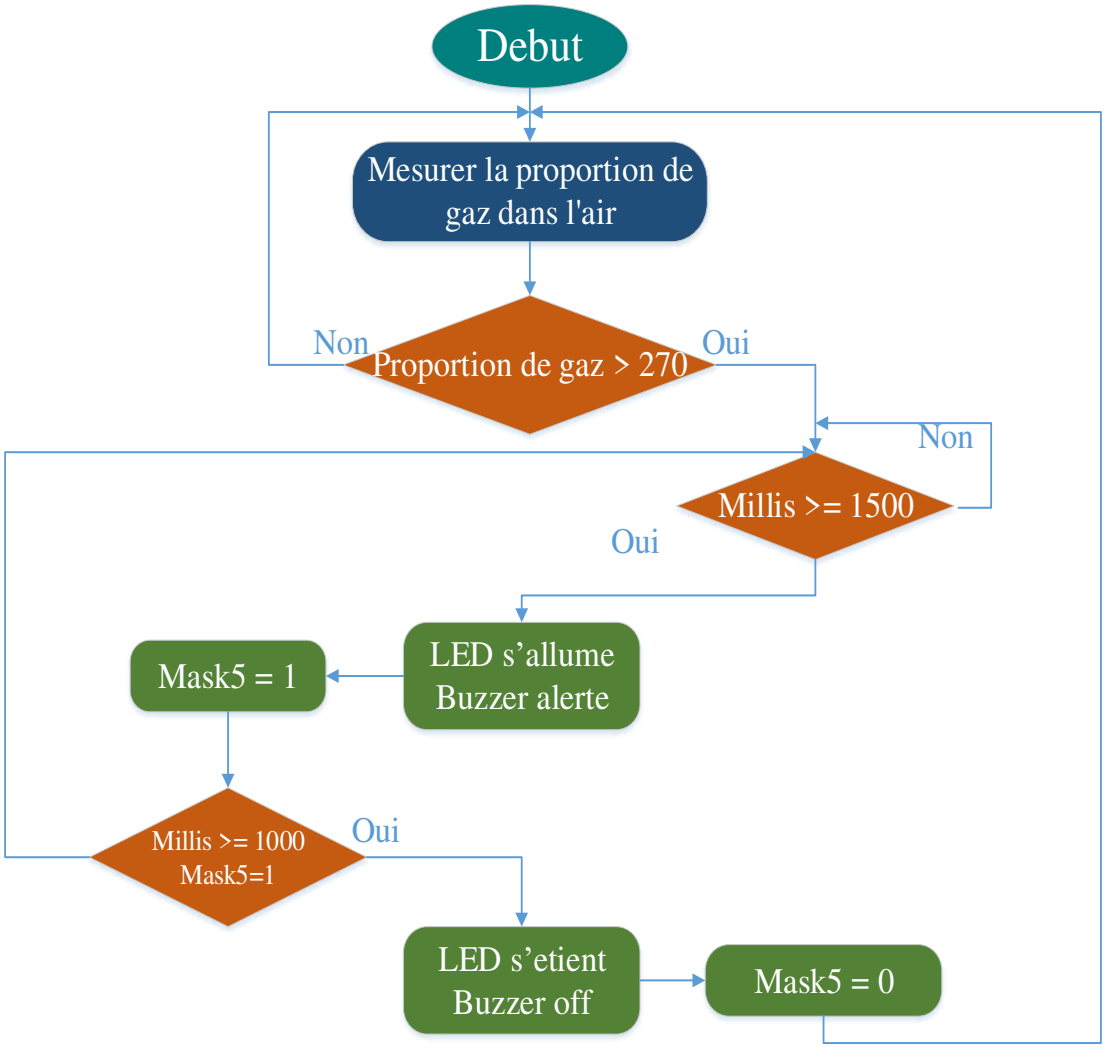


Figure III. 5: *L'organigramme de capteur de gaz MQ5*

B. Système de détection des flammes

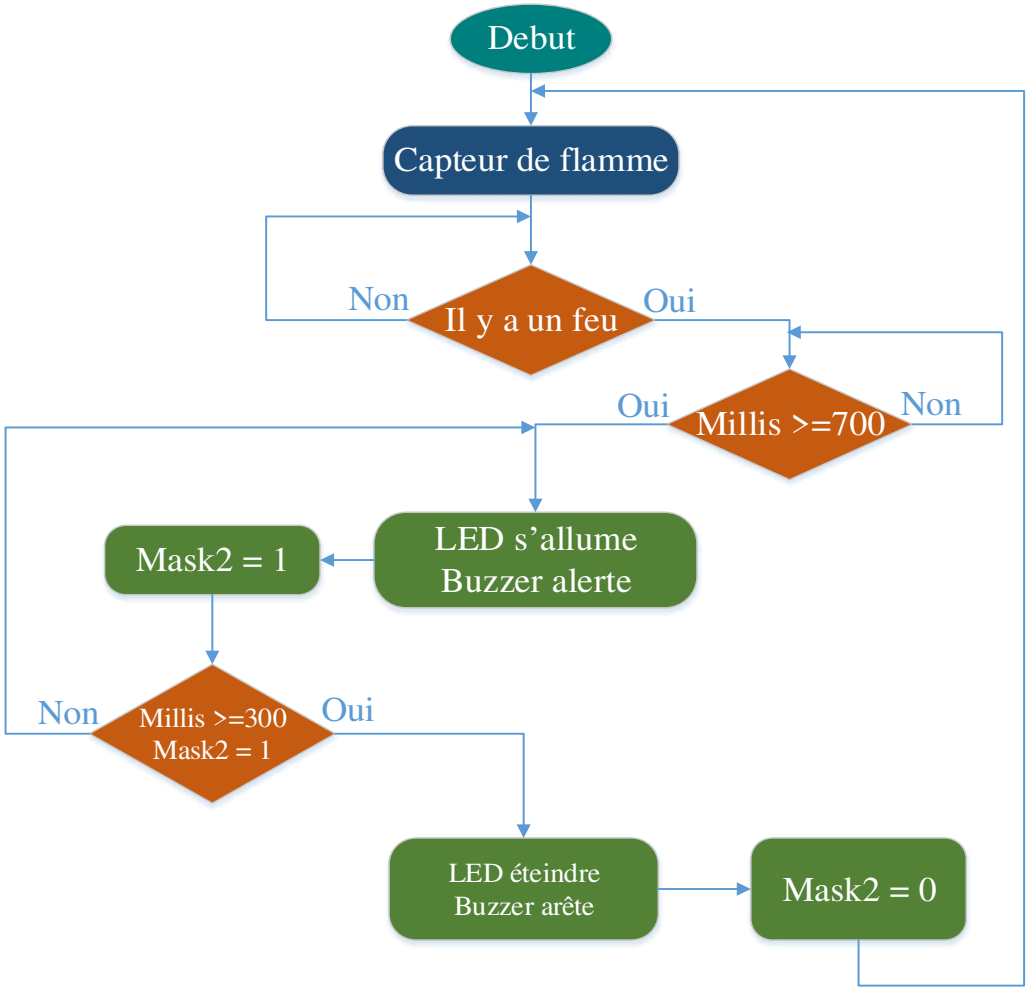


Figure III. 6: *L'organigramme de capteur de flamme.*

III.1.2.5 Le système d'accès au garage

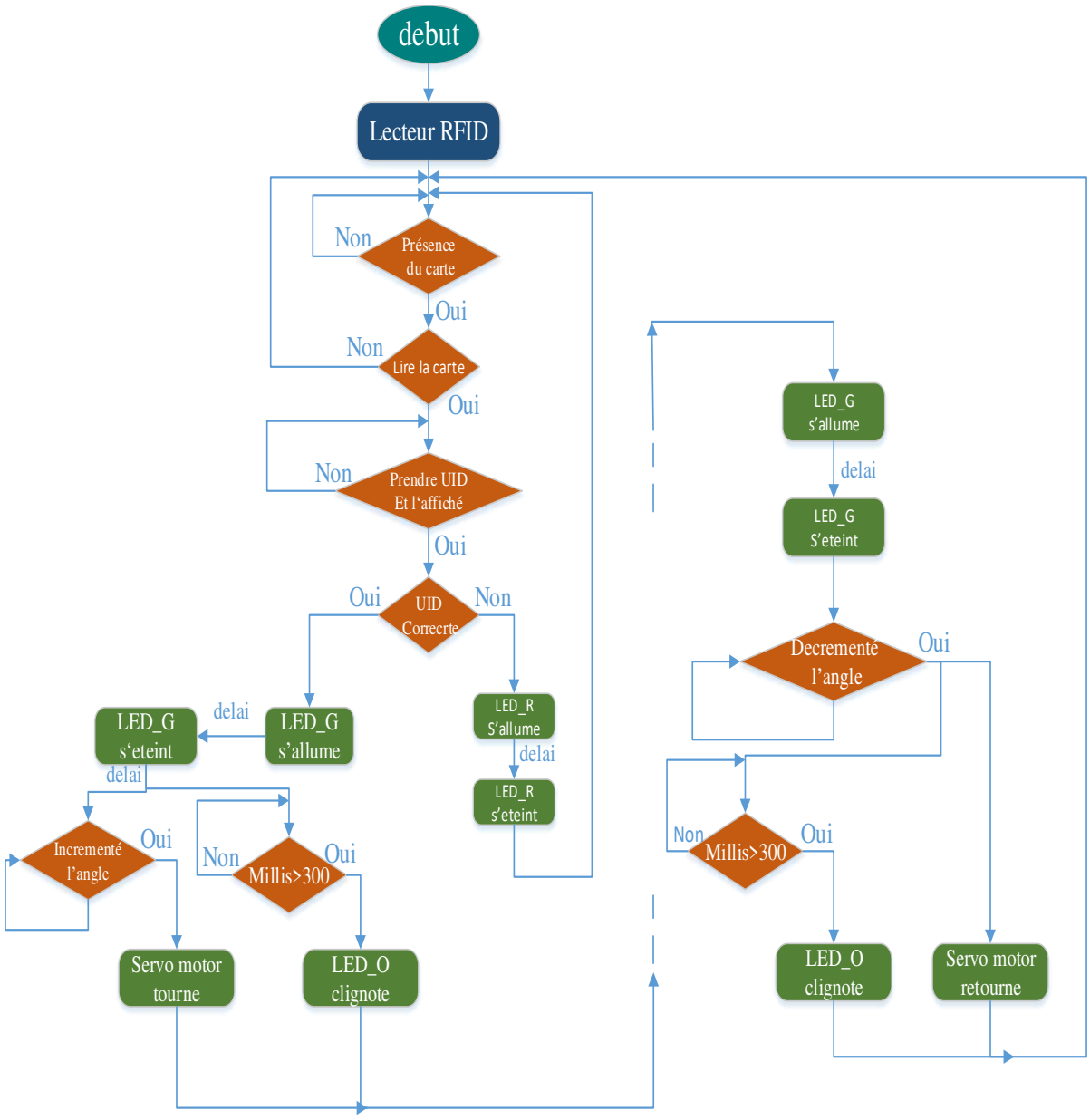


Figure III. 7: *L'organigramme de module RFID.*

III.1.2.6 Le système de commande du rideau balcon :

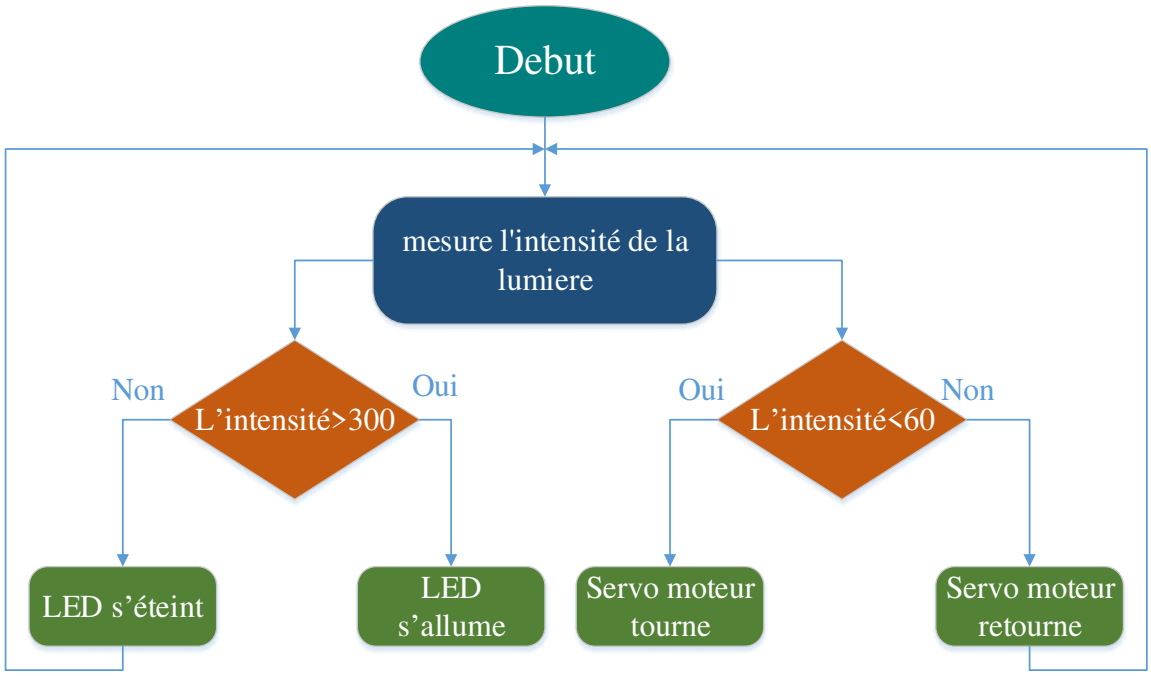


Figure III. 8: *L'organigramme du capteur de la lumière.*

III.1.2.7 Le système de contrôle à distance

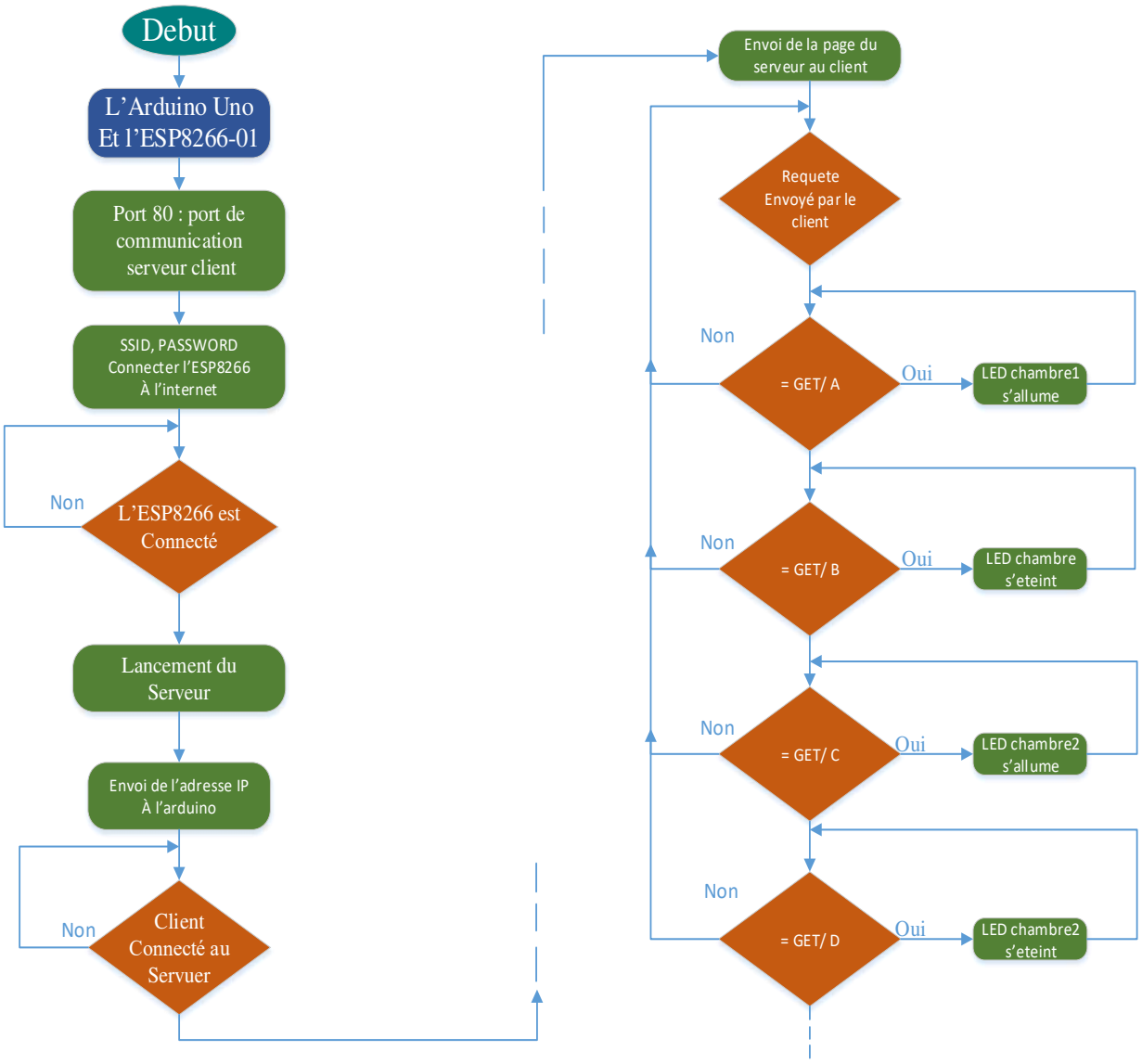


Figure III. 9: *L'organigramme de module ESP8266-01.*

III.1.2.8 Le système global : il est illustré sur la figure III.10

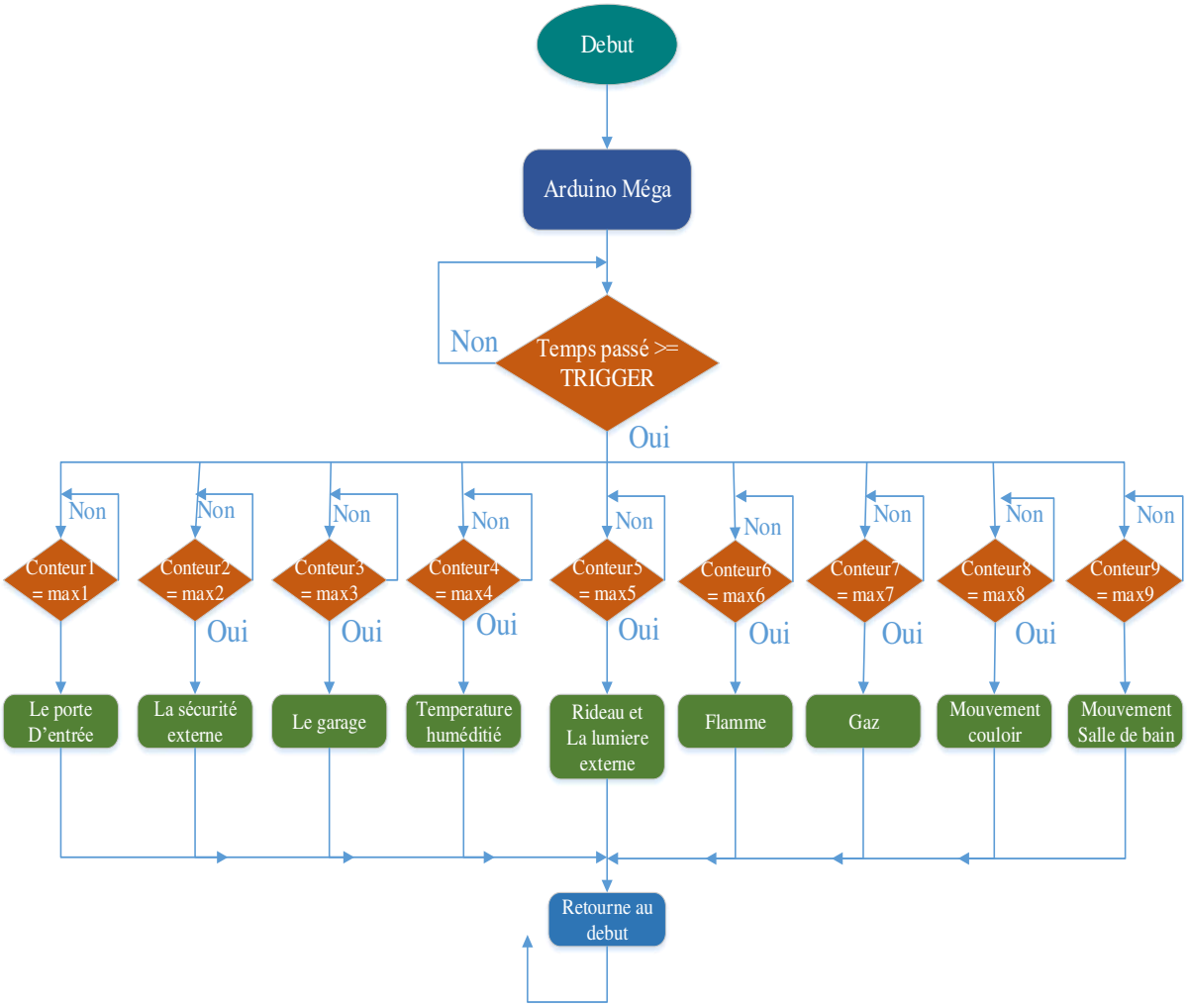


Figure III. 10: *L'organigramme general.*

III.1.3 Application sur android

On a créé une application Android sur le site MIT qui contient des boutons on/off de chaque chambre pour faire la commande de l'allumage et l'éteigne de la lumière à distance par le smartphone, sans se déplacer ou faire un effort physique dans le but de faciliter notre quotidien

- Interface:

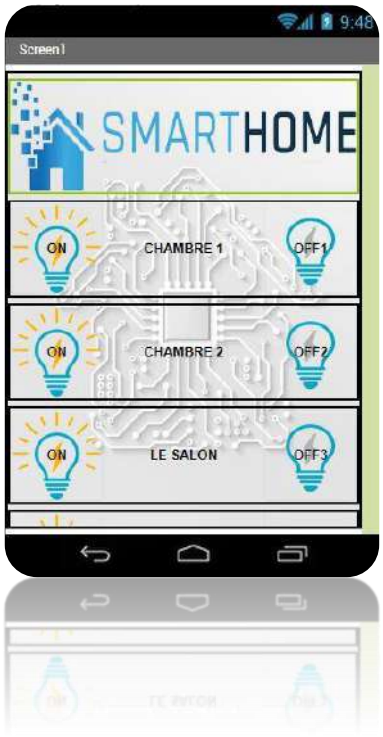


Figure III. 11: *L'interface de l'application Android.*

- Le programme de l'application:

- Le block de la chambre 1 :

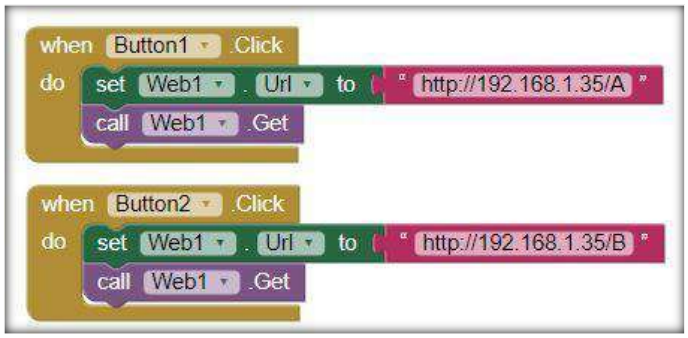


Figure III. 12: *les bouton ON/OFF de la chambre 1.*

- Le block de la chambre 2 :

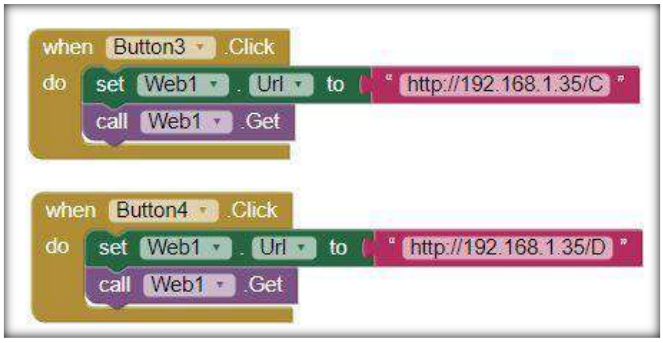


Figure III. 13: *les bouton ON/OFF de la chambre 2.*

- Le block du salon :

-

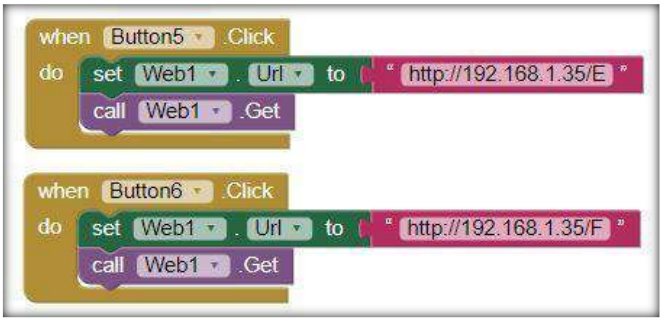


Figure III. 14: *Les bouton ON/OFF du salon 3.*

- Le block de la cuisine :

-

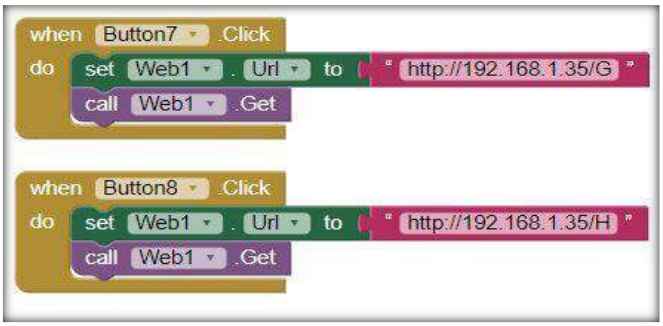


Figure III. 15: *les boutons ON/OFF de la cuisine.*

III.2 Aspect hardware de la réalisation.

Avant de fabriquer le circuit imprimé on à fait une simulation avec ISIS pour organiser notre travaille et aussi pour confirmer qu'il n'y a aucune erreur et pour on peut créer les packages qui on a utilisé dans ARES

III.2.1 Simulation par le logiciel Proteus (ISIS)

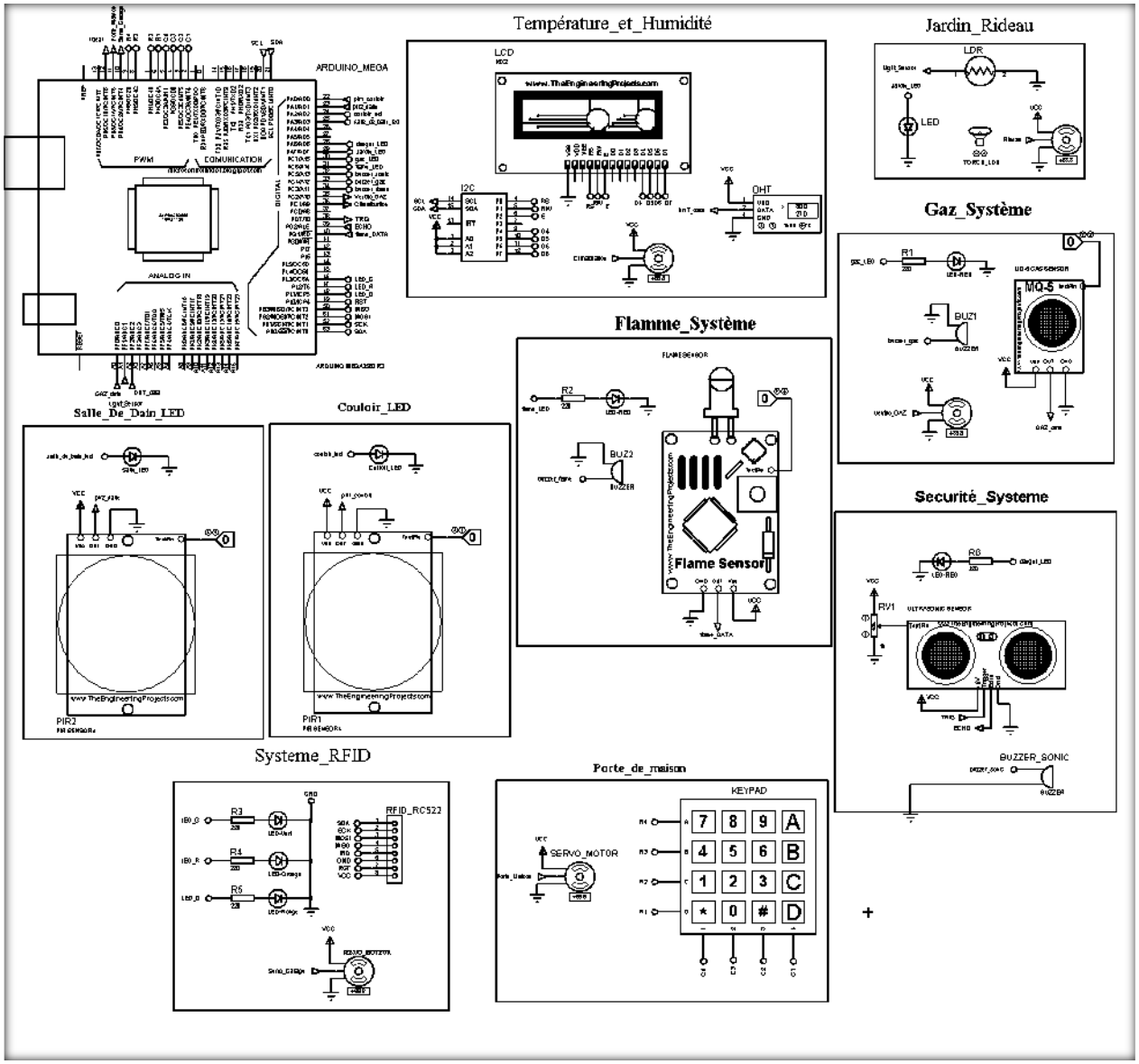


Figure III. 16: *Le schéma général de la simulation.*

- Le système de sécurité externe

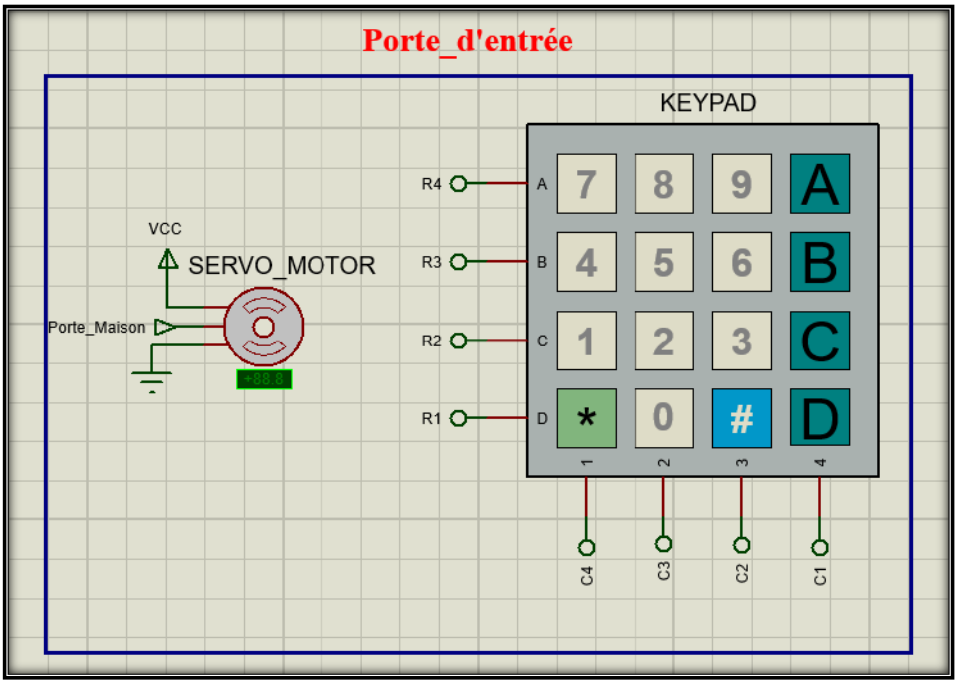


Figure III. 17: *système d'ouverture et fermeture de la porte.*

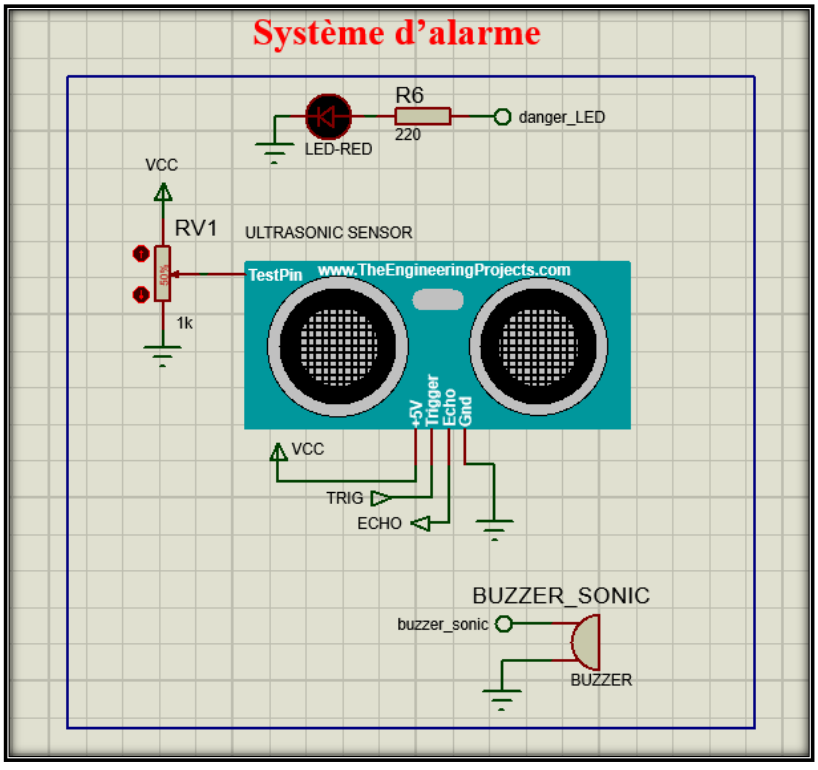


Figure III. 18: *système de capteur ultrasonique HC-SR04.*

- Le système de sécurité interne

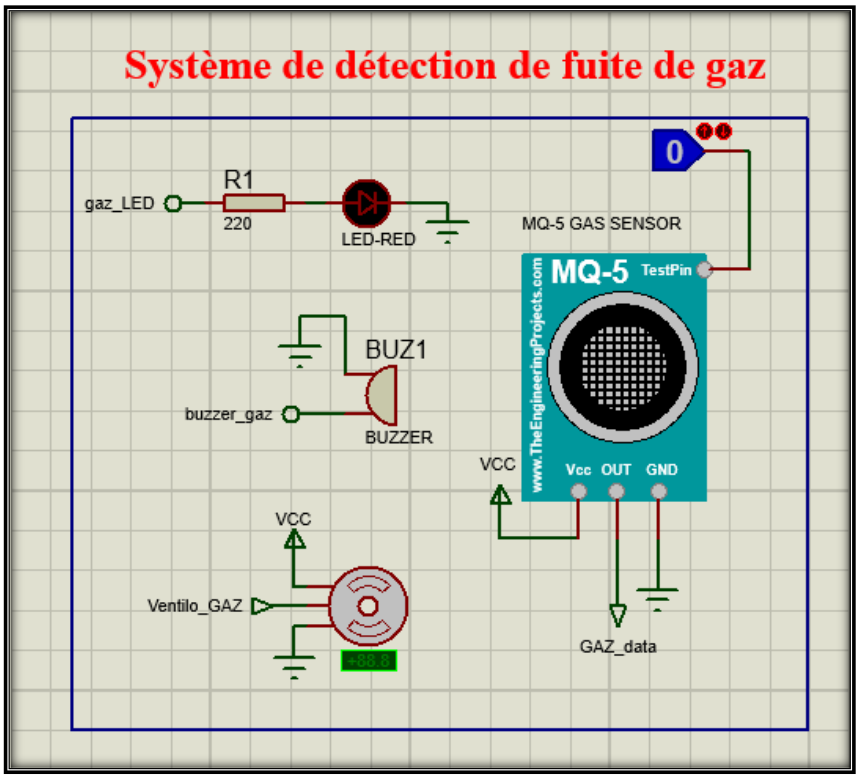


Figure III. 19: *système de capteur de gaz MQ5.*

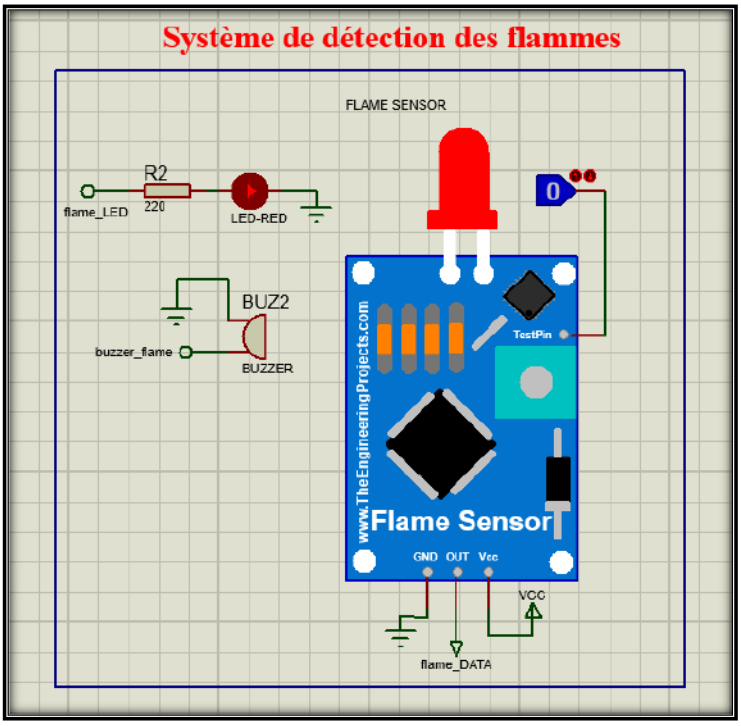


Figure III. 20: *système de capteur de flamme.*

- Le système d'économie d'énergie

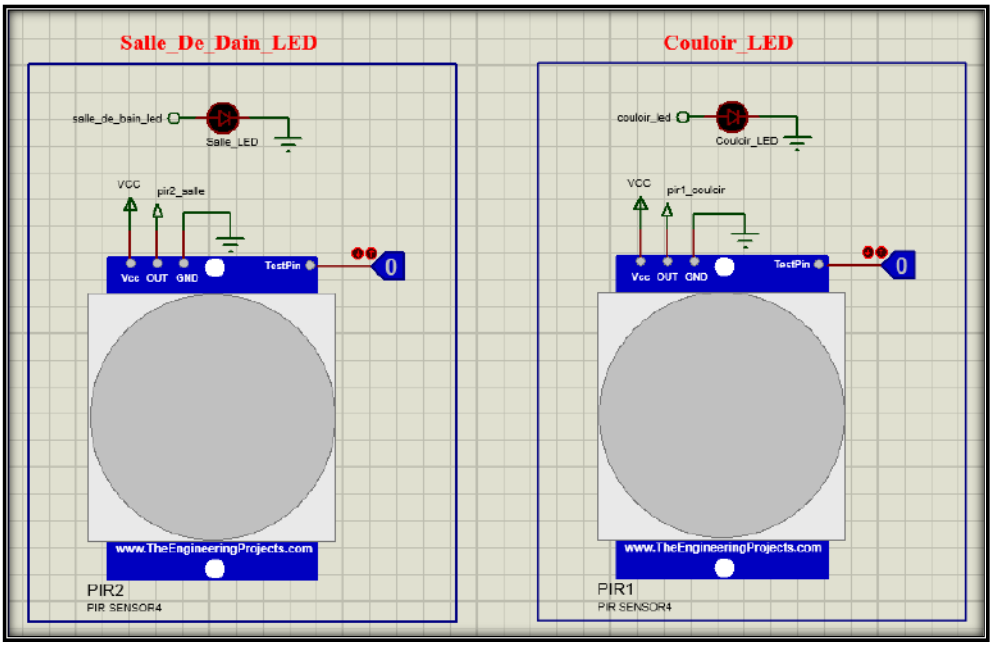


Figure III. 21: *Système de capteur de mouvement (PIR).*

- Le système d'accès au garage

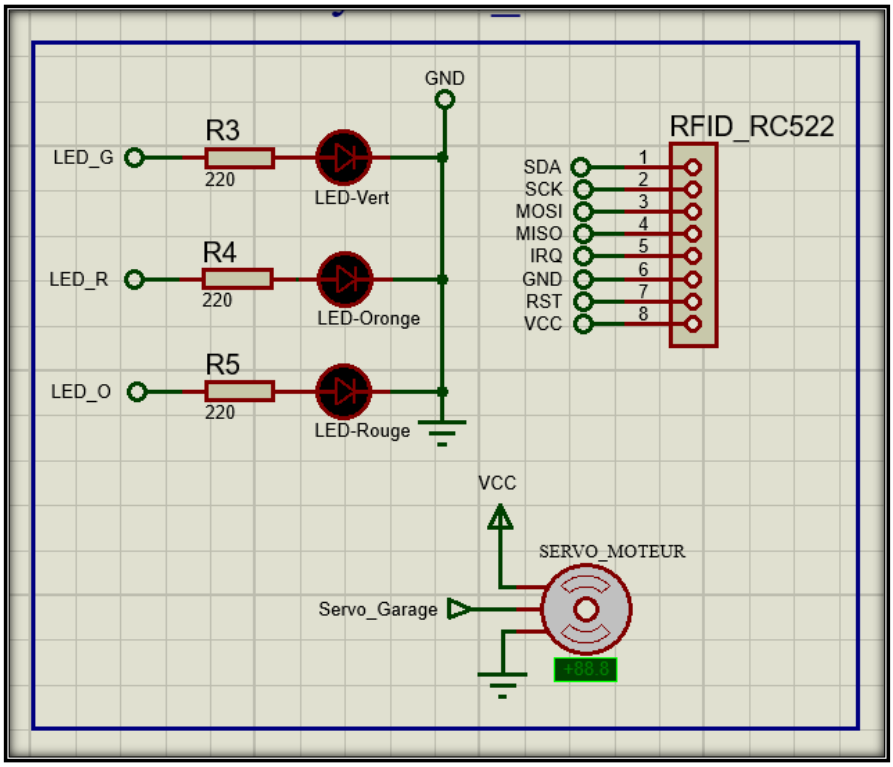


Figure III. 22: *Systeme RFID.*

- Système de température et climatisation

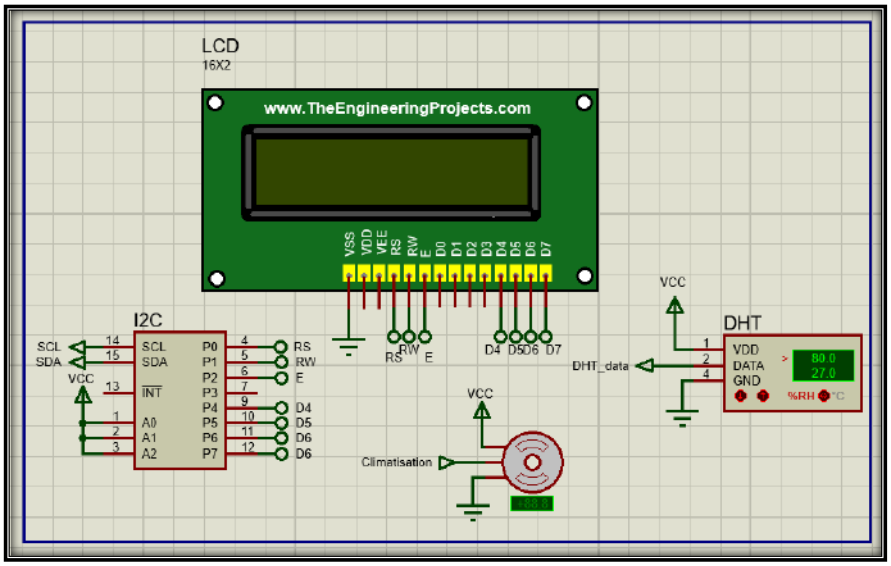


Figure III. 23: *Système de capteur DHT22 avec un afficheur LCD.*

- Le systeme de l'éxterieur

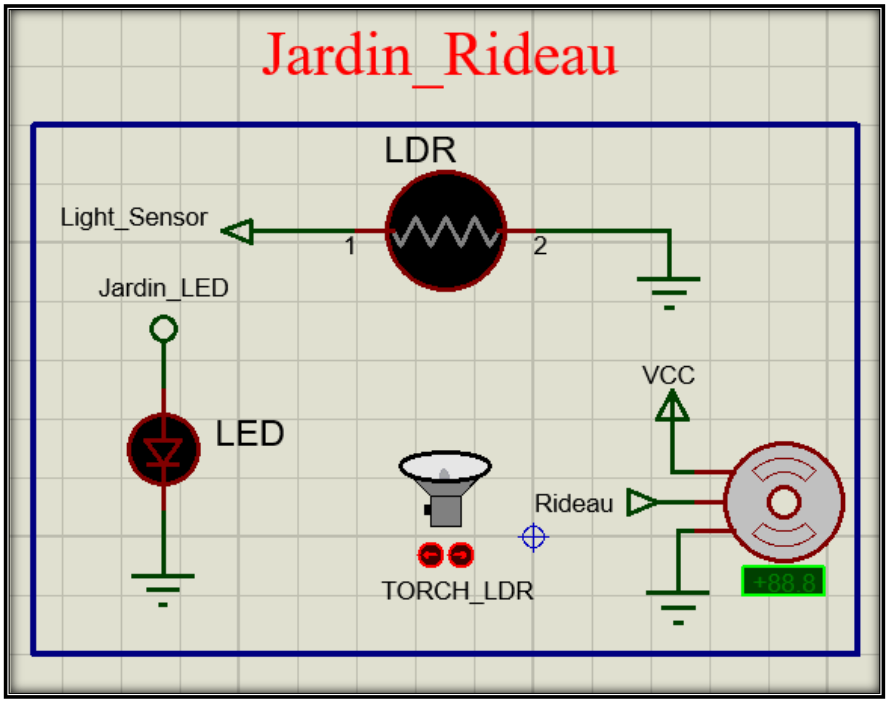


Figure III. 24: *Système de capteur de Lumière.*

- Le système de contrôle à distance

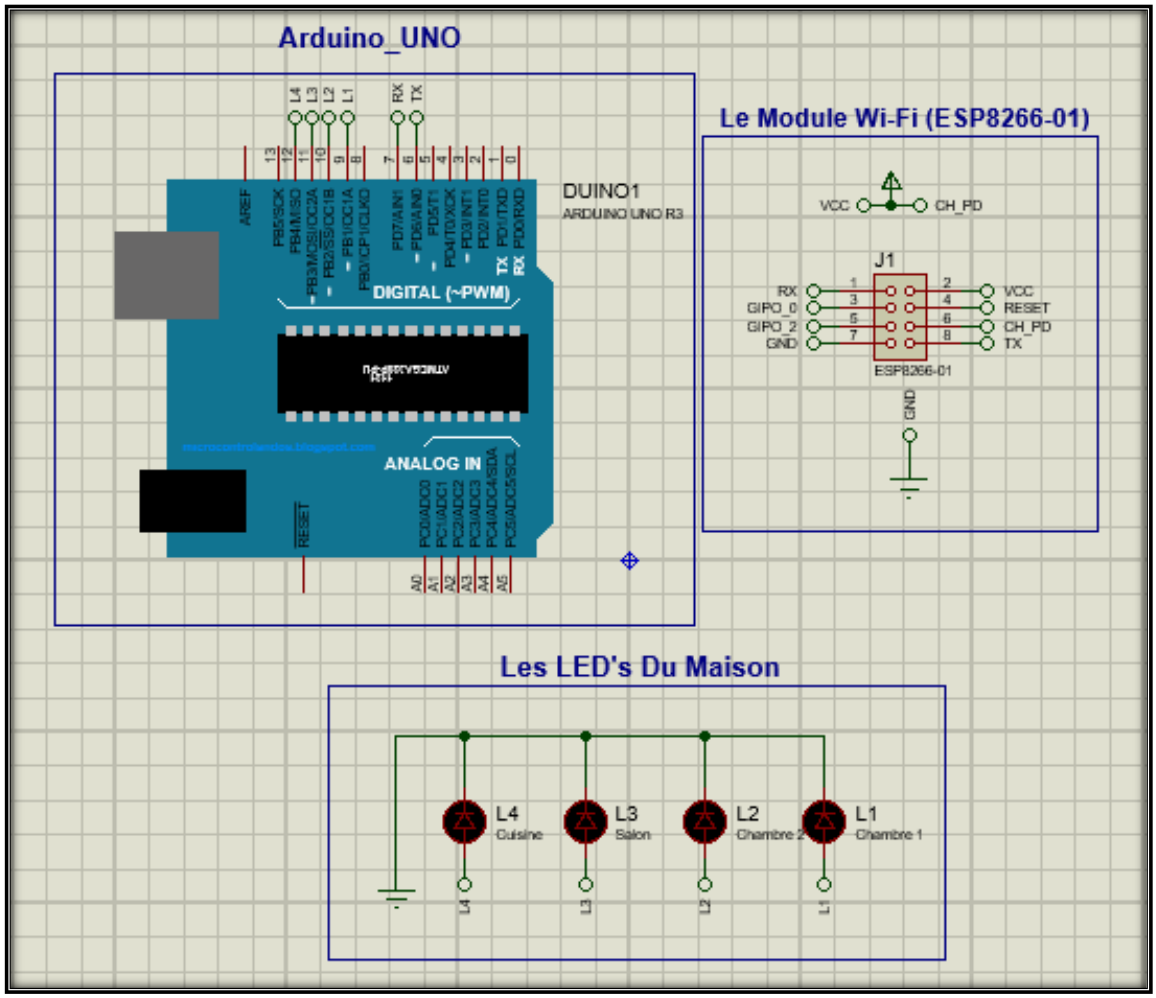


Figure III. 25: *Le système de module ESP8266-01 avec Arduino Uno.*

III.2.2 Assemblage du matériel sur ARES

- Schéma de routage

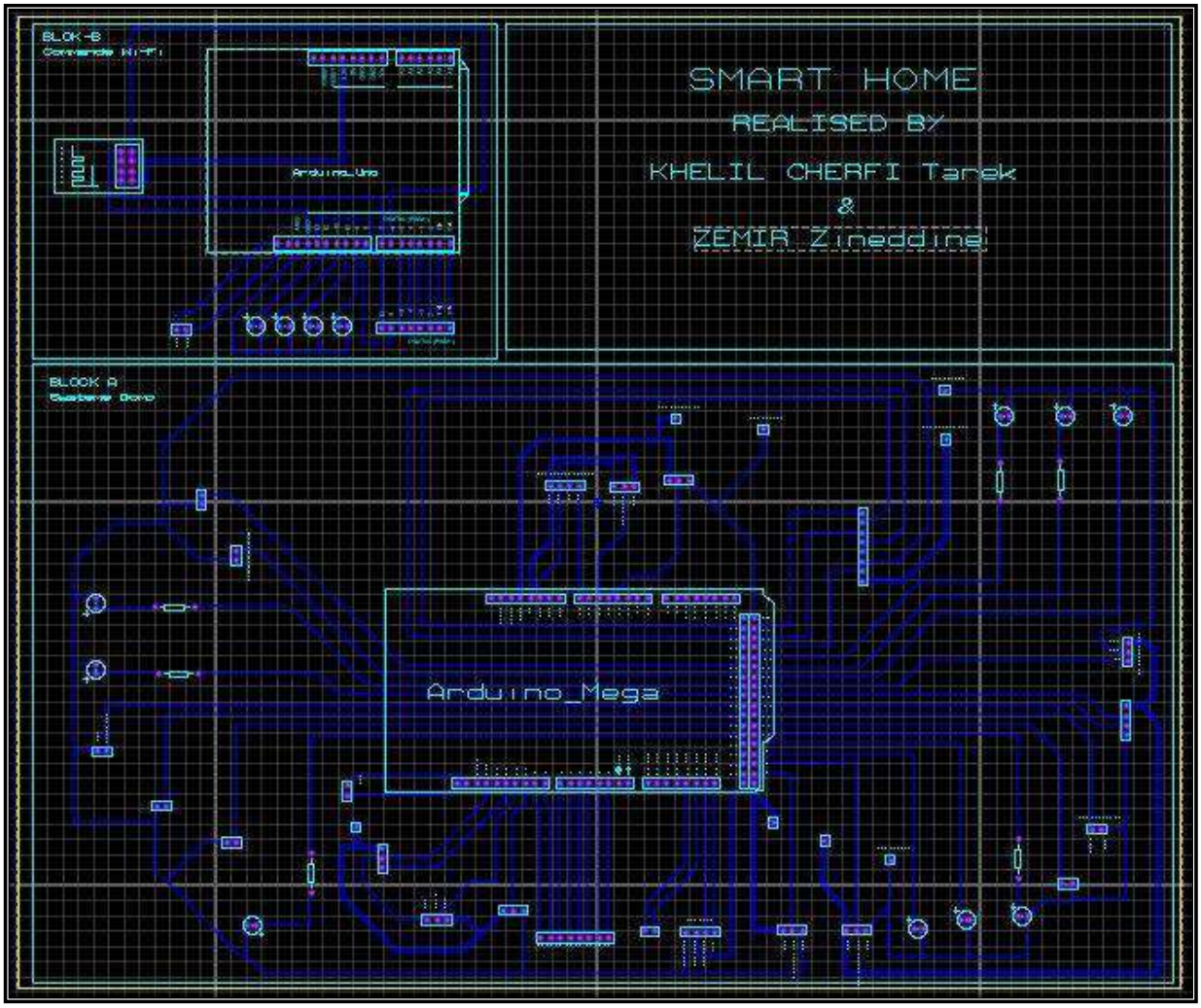


Figure III. 26: *schéma de routage ARES*

- Schéma de routage fichier PDF

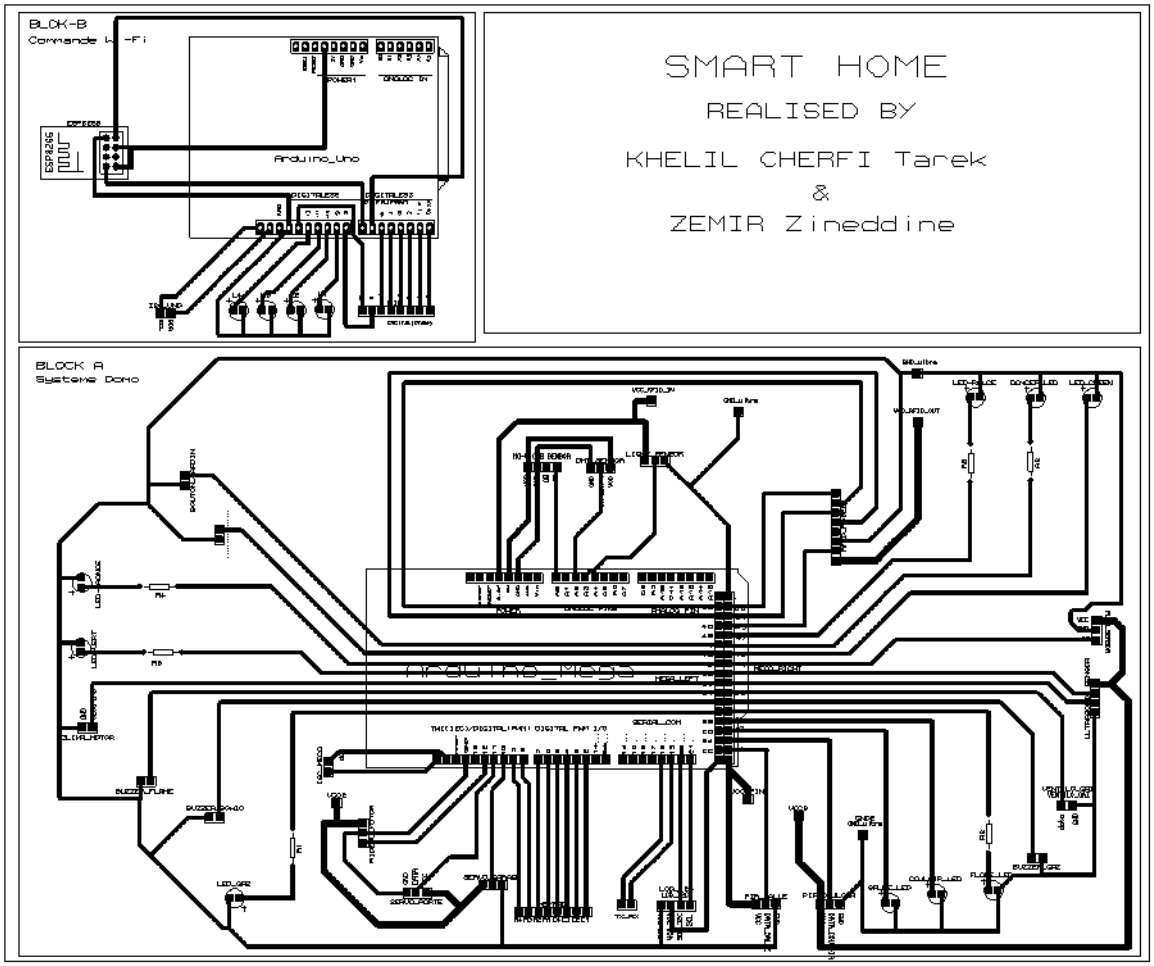


Figure III. 27: *schéma de routage PDF.*

- Le typon final de la carte :

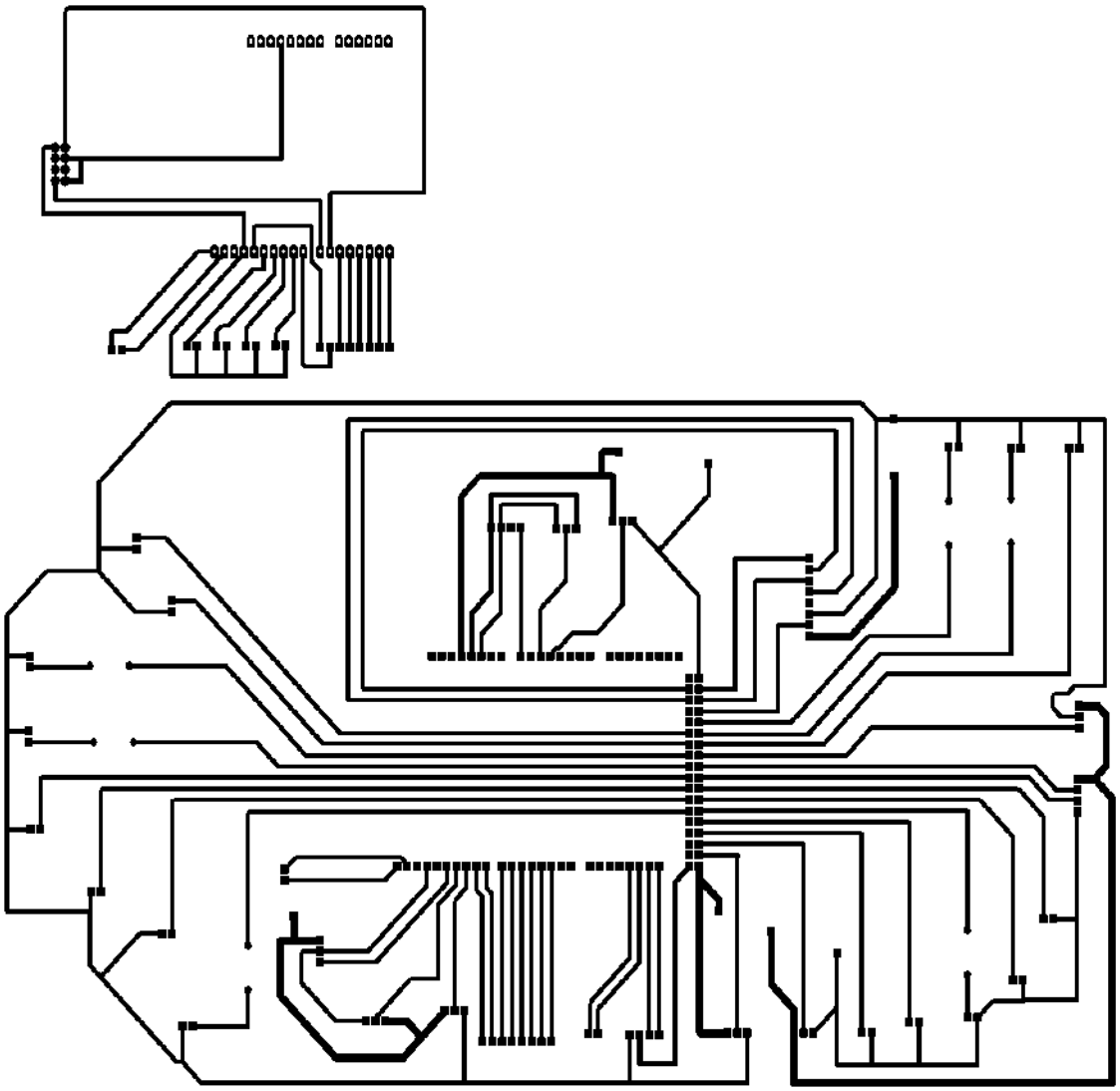


Figure III. 28: *Le typon.*

III.2.3 Assemblage du matériel sur la plaquette d'essaie

- Soudage de module I2C avec afficheur LCD

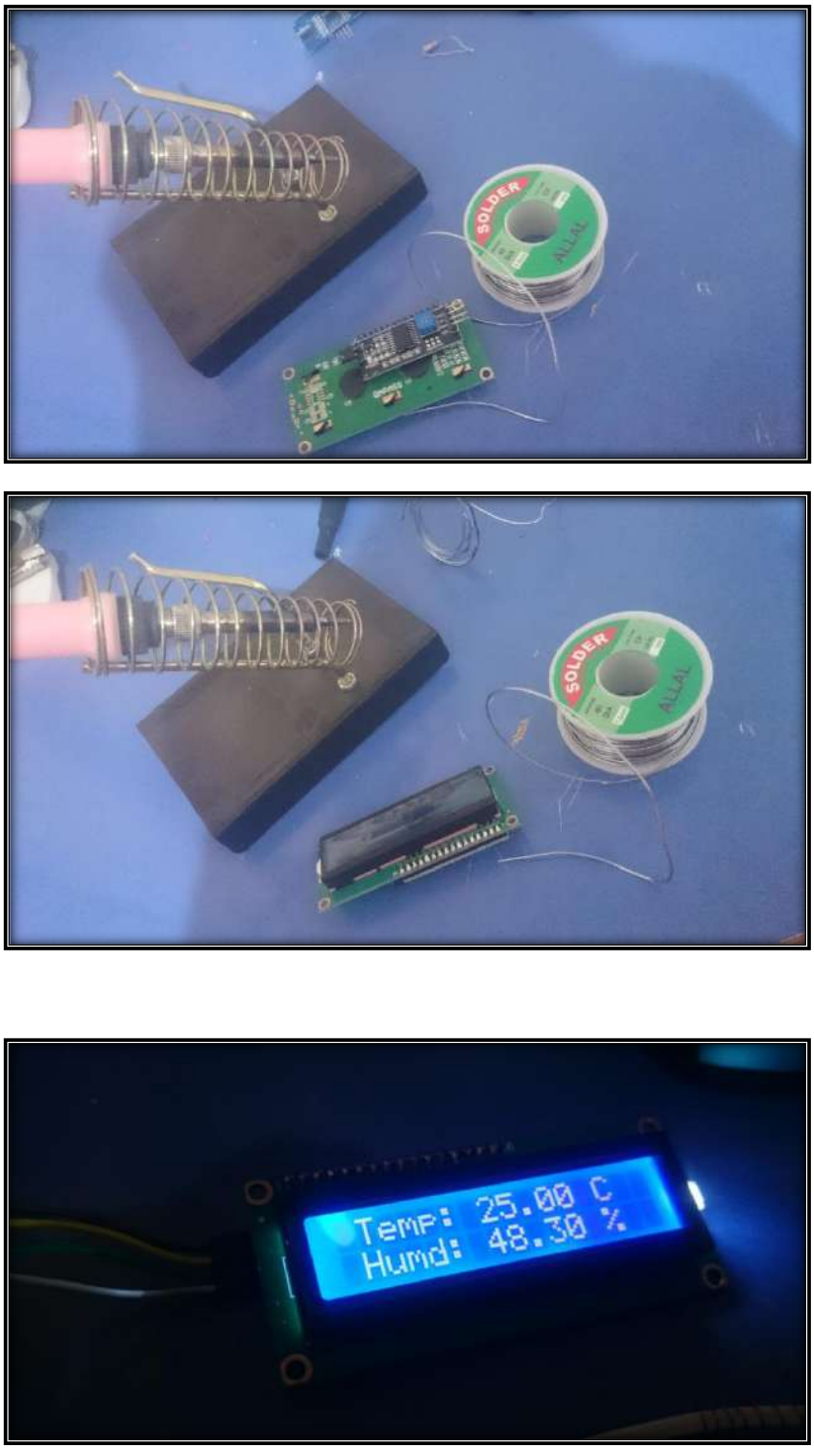
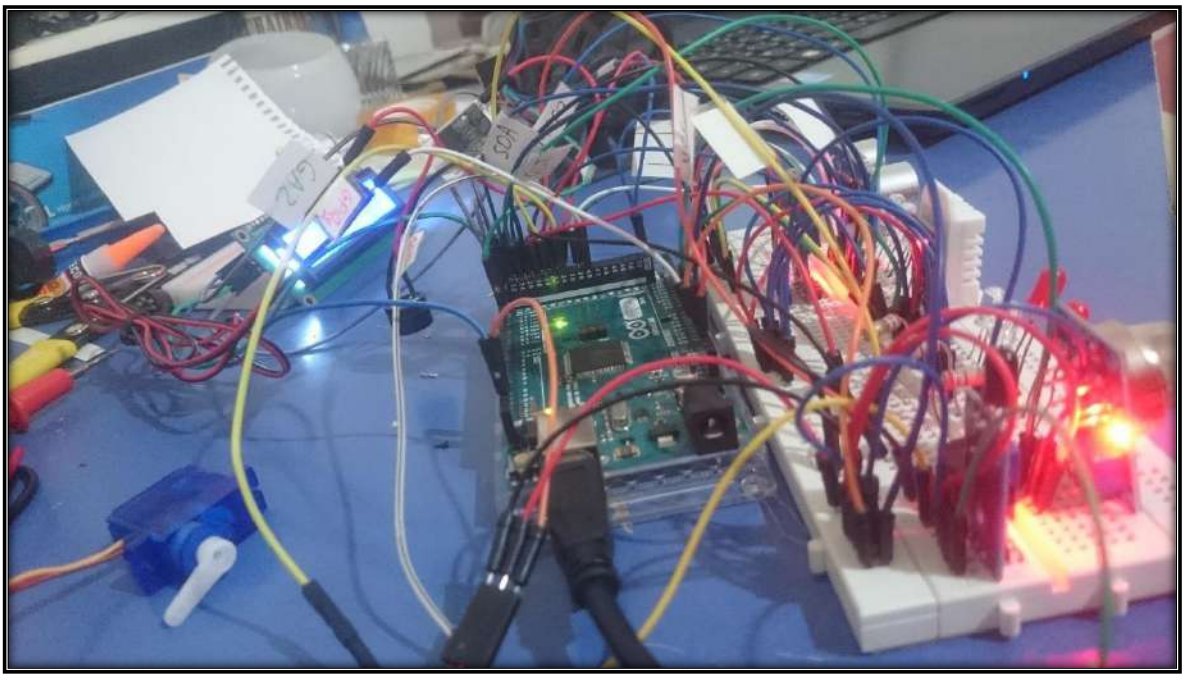


Figure III. 29: *Adaptateur I2c avec LCD* 16x2.

- Montage du matériel sur la plaquette



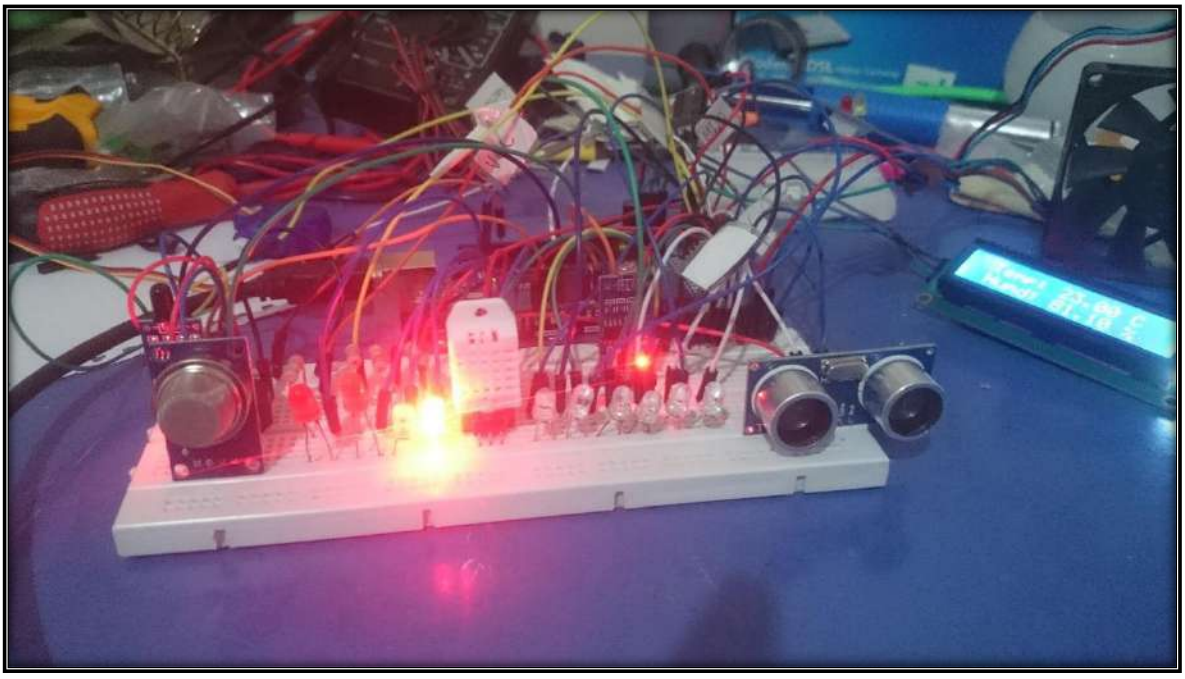


Figure III. 30: *Assemblage sur la plaquette d'essaie*

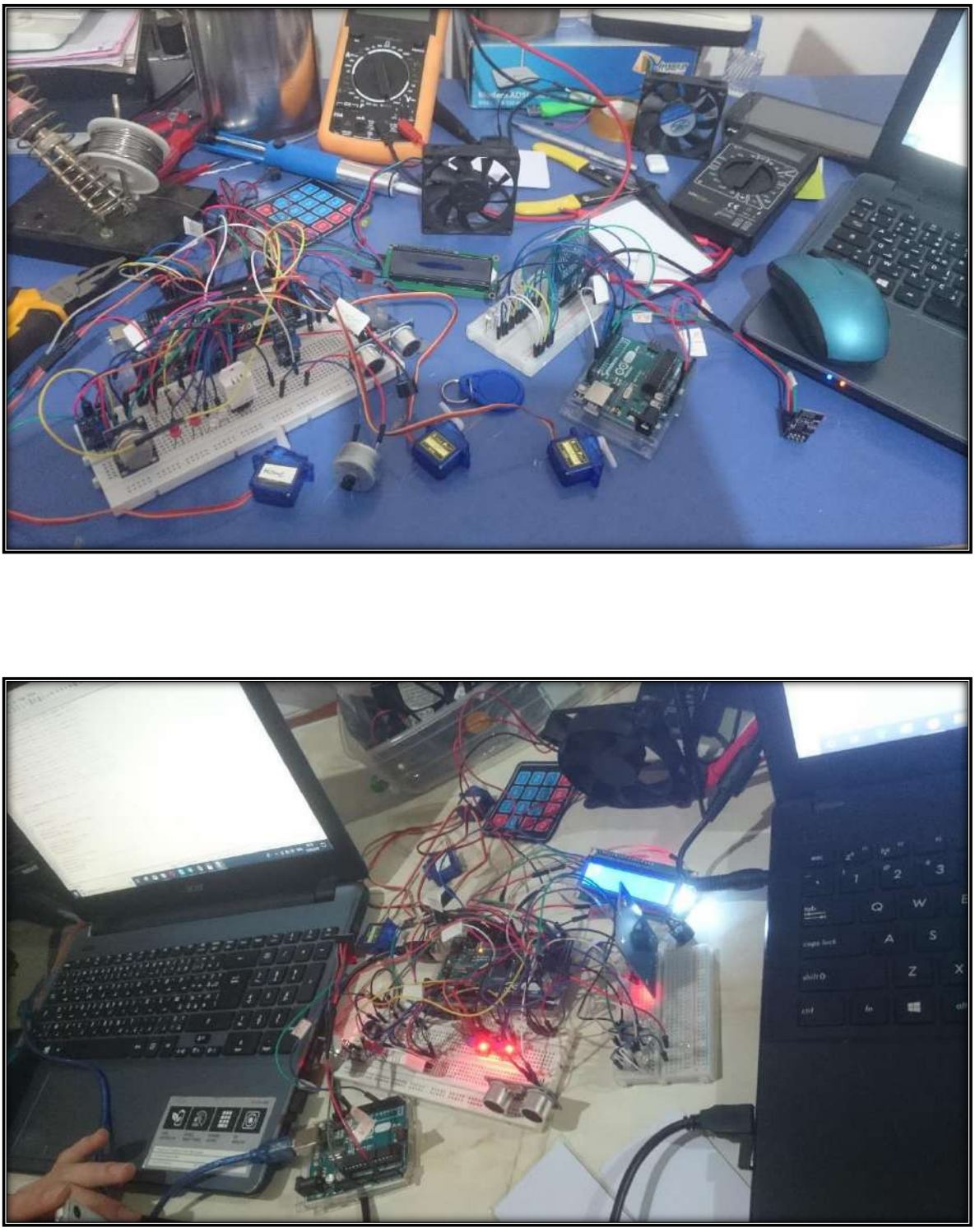


Figure III. 31: *Assemblage finale sur la plaquette.*

III.2.4 Structure de la maison

On a construit une maquette d'une maison pour réaliser notre travail en réalité

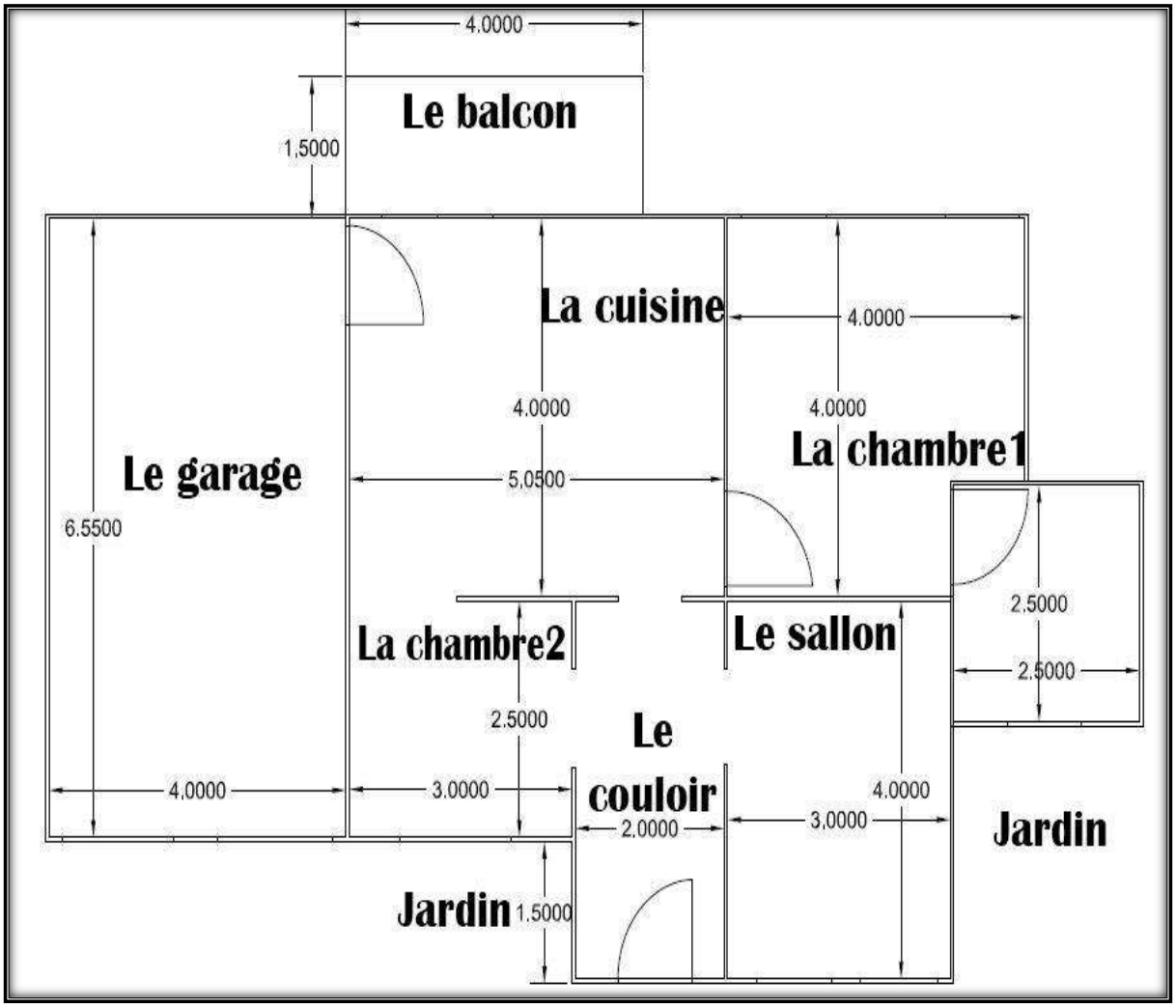


Figure III. 32: *le plan de la maison*

III.3 Le produit fini

III.3.1 La plaque électronique de le circuit imprimé

- La vision en 3D de Block A

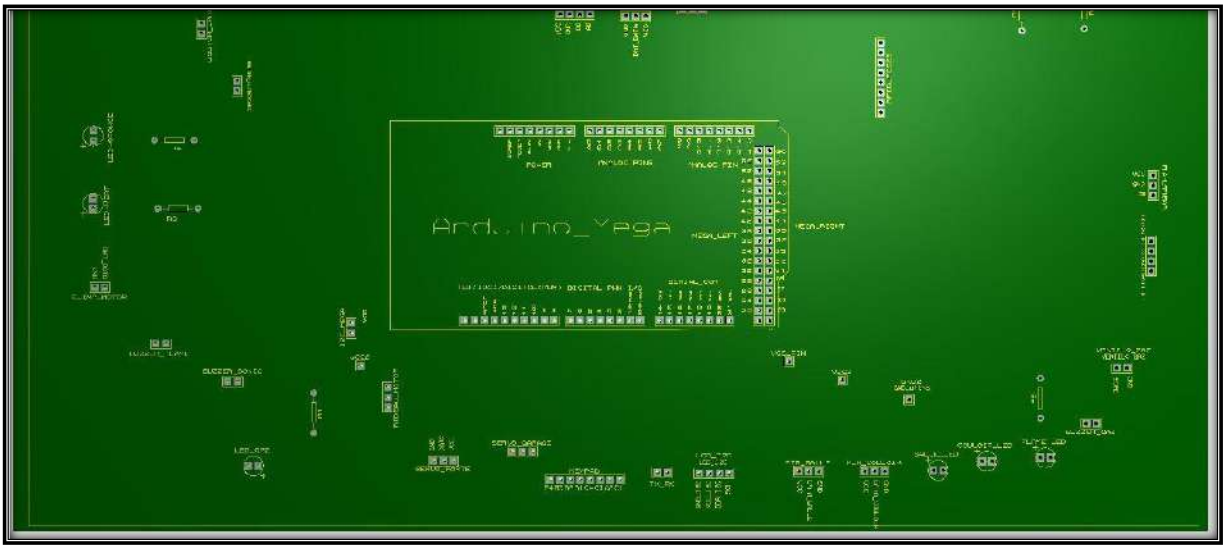


Figure III. 33: *Block A de la carte.*

- La vision 3D de Block B :

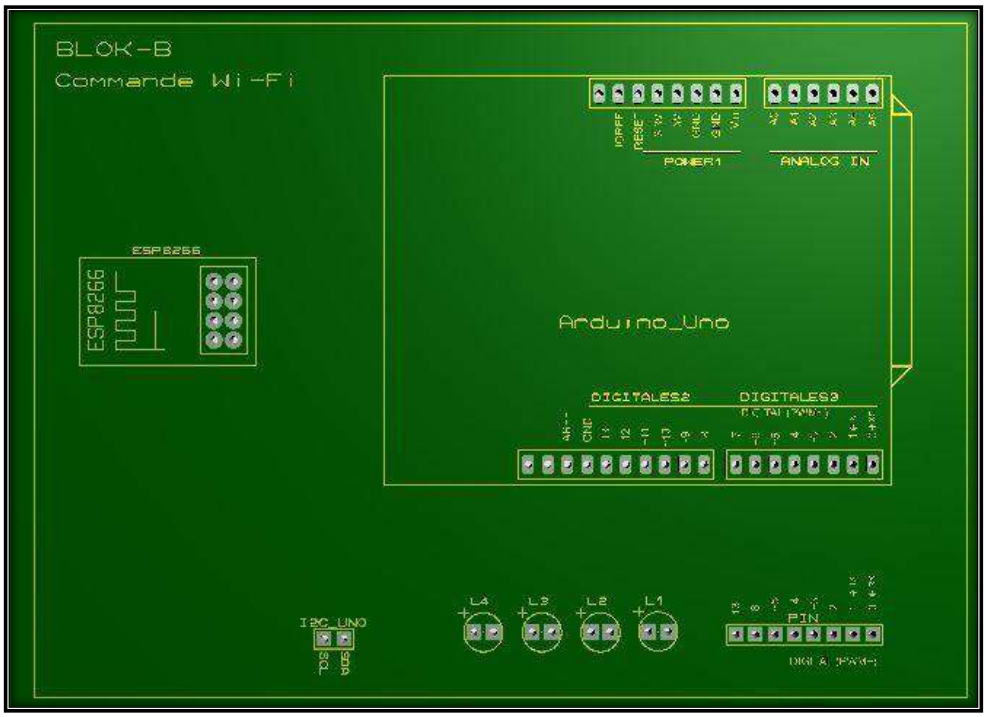


Figure III. 34: *Block B de la carte.*

- La vison 3D de l'arrière-plan de la plaquette :

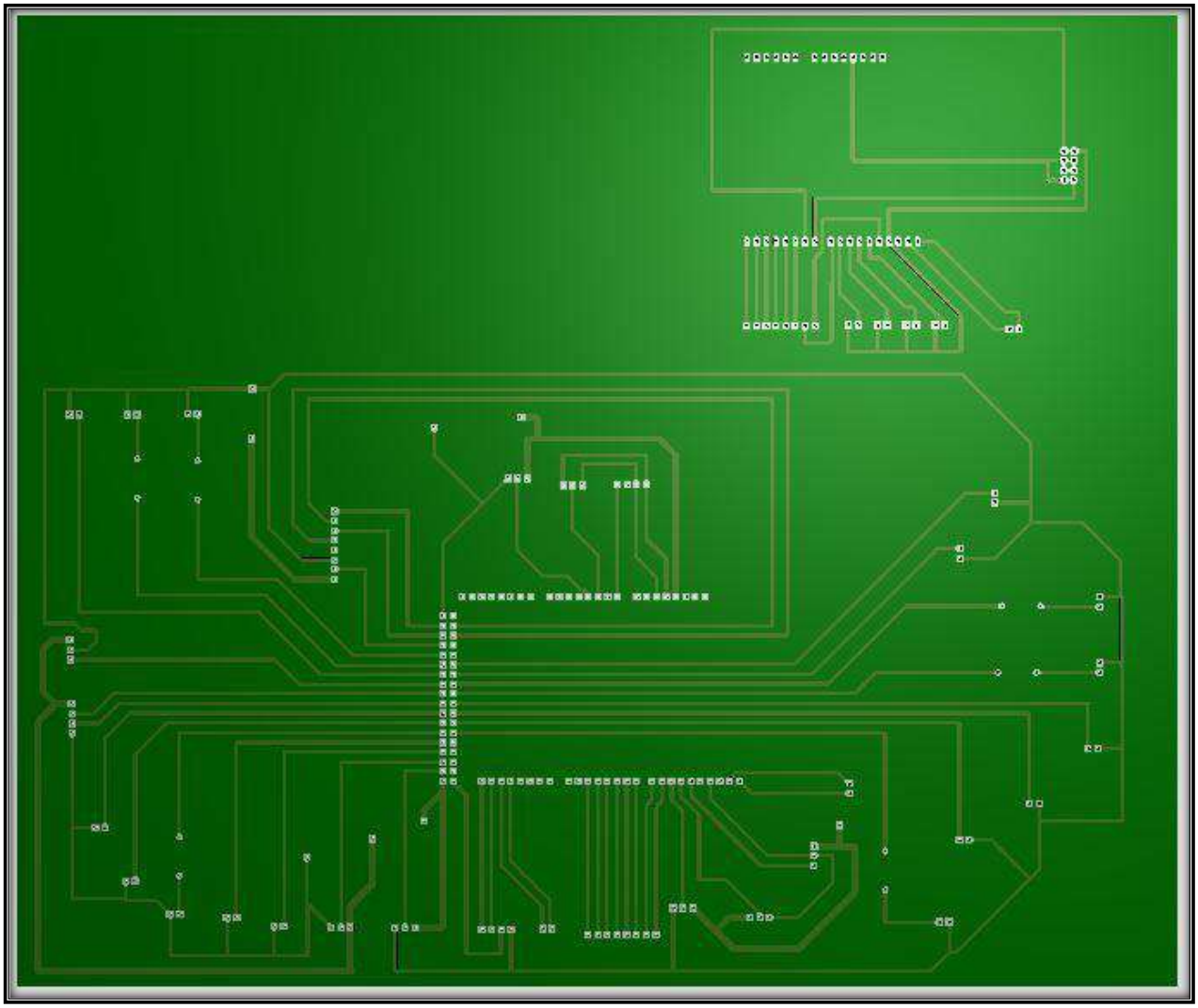


Figure III. 35: *l'arrière-plan de la carte.*

- Le circuit imprimer

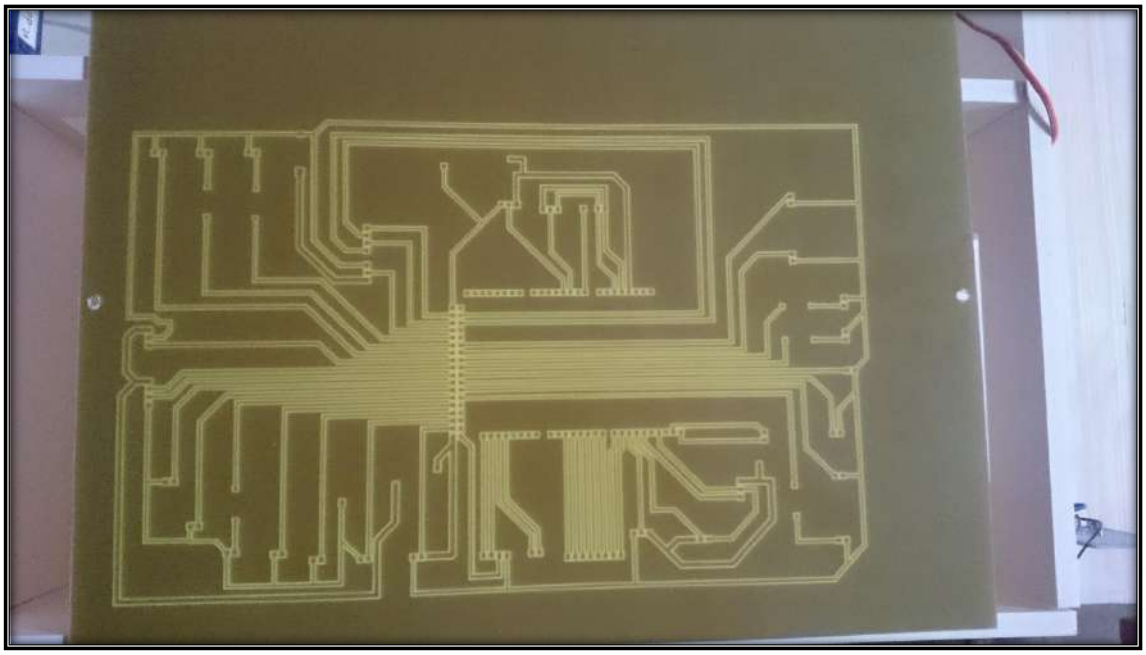


Figure III. 36 *: circuit final.*

III.3.2 La maquette de la maison



Figure III. 37: la maquette de la realisation



Figure III. 38: *la façade de la maquette.*

III.4 Conclusion

Nous avons exposé dans ce chapitre l'aspect réalisation de notre maison intelligente. Le maximum de concision a été ciblé sans pour autant perdre de clarté dans l'exposé. Il importe de noter quelques difficultés rencontrées pendant la réalisation de ce travail, le plus important étant celui de la saturation de L'Arduino, les timers (la fonction millis) nous ont été d'un grand secours, chaque fonction est périodiquement testée permettant ainsi de gérer les composants en parallèle virtuellement. Parmi ces problèmes aussi le module wifi et le module rfid les deux travaillent avec le bus de communication SPI et l'Arduino a un seul bus SPI donc on était obligé d'utiliser le 2eme Arduino (UNO) pour le module ESP8226-01(Wi-Fi). Nous avons aussi présenté notre travaille rée avec des figures que nous avant photographiée sur le champ.

Conclusion générale

Conclusion générale

Les systèmes domotiques permettent de contrôler et d'automatiser un grand nombre de fonctionnalités dans nos maisons permettant à la fois bien être, sécurité et économie d'énergie. La possibilité d'utiliser des Smartphones ou des tablettes rend les systèmes domotiques très utiles dans le futur. Bien que les installations de la domotique en ce moment peuvent s'avérer couteuses, la logique du marcher fera sans doute que les prix vont baisser et devenir à la portée de tout le monde.

Il va sans dire que la pièce maitresse dans cette réalisation qui est l'ARDUINO joue un rôle prépondérant dans les applications électroniques de nos jours. Sa disponibilité sur le marché, son prix abordable et surtout sa documentation accessible à tout le monde en font un dispositif de choix pour ceux qui désirent concevoir et mettre au point des réalisations aussi performantes que bon marché.

Ce projet nous a permis de comprendre et d'approfondir tant de notions dans ce domaine. Les problèmes auxquels nous avons fait face pendant la réalisation nous ont appris tant de choses notamment l'importance de l'organisation, et de la méthodologie de travail qui sont deux éléments clés dans la réussite.

Pour terminer, il importe de souligner qu'un travail est toujours perfectible et d'autres améliorations peuvent être apportées. Ceci est d'autant plus vrai que les possibilités offertes par la domotique sont loin d'être limitées.

Références Bibliographiques

[1] https://www.climamaison.com/domotique/definition.htm le 2019-03-03 à 10:31:01

[2] https://www.climamaison.com/domotique/les-principes.htm le 2019-03-03 à 11:10:33

[3] https://www.maison-et-domotique.com/47895-la-domotique-cest-quoi/ le 2019-03-03 à 2:05:48

[4] http://igm.univ-mlv.fr/~dr/XPOSE2007/aessaidi-ndiop_LA-DOMOTIQUE/intro.htm le2019-03-04 à 09:15:13

[5] https://domotique.ooreka.fr/comprendre/maison-communicante le 2019-03-10 à 10:12:50

[6] https://www.techno-science.net/definition/3920.html le 2019-05-15 à 02:05:09

[7] http://www.pcpratique.com/ftechniques/fethernet_print.html le 2019-05-15 à 02:20:19

[8] https://www.abix.fr/cable-ethernet le 2019-05-16 à 01:59:28

[9] https://adsl.ooreka.fr/astuce/voir/747627/rj45 le 2019-05-16à 01:55:50

[10] technologie-bluetooth.html le 2019-04-20 à 16:56:22

[11] https://www.domotique-info.fr/technologies-domotique/zigbee/ le 2019-05-08 à 23:42:17

[12] www.leadertelecom.fr/borne_wifi_113.htm le 2019-05-17 à 02:51:19

[13] https://slideplayer.fr/slide/13650581/ le 2019-05-13 à1 :26 :10

[14] https://le-routeur-wifi.com/normes-wifi-802-11/ le 2019-05-07 à 2:51:30

[15] www.netacad.com ccna1 (cour CCNA1 chapitre 4) le 2019-05-08 à 12 :30 :06

[16] www.afrikanet.com le2019-05-08 à 1:02 :22

[17] www.netacad.com ccna1 (cour CCNA1 chapitre 4) le 2019-05-08 à 01 :30 :06

[18] https://www.commentcamarche.net/faq/3020-wifi-cours-d-introduction 2019-05-19 02:20:06

[19] https://rfid.ooreka.fr/comprendre/systeme-rfid 2019-05-20 a 01:42:32

[20] Nicolas Seriot. Les systèmes d'identification radio (RFID) — fonctionnement, applications et dangers —, IL–2005b, Yverdon-les-Bains.

[21] Tania Martin Le b.a.-ba de la RFID origines, Technologie et Applications Publication périodique de Smals Janvier 2014

[22] https://fr.wikipedia.org/wiki/Protocoles_de_communication_RFID 2019-05-19 à 22:37:50

[23] https://rfid.ooreka.fr/comprendre/systeme-rfid 2019-05-19 à 23:15 :09

[24]http://www.banq.qc.ca/services/services_professionnels/milieux_doc/dossiers_thematiques/services_techniques/rfid/ 2019-05-20 a 01:42:32

[25]https://www.companeo.com/portiques-de-securite/guide/technologie-rfid 2019-05-20 01:42:32

[26] https://www.gotronic.fr/art-carte-arduino-mega-2560-12421.htm 2019-05-16 02:47:42

[27] Le grand livre d'Arduino Erik Bartmann la page numéro 16

[28] Betrand.Cottenceau, Carte ARDUINO UNO Microcontrôleur Microcontrôleur ATMega328, page 6/23, sept 2007

[29] https://www.iot-jungle.com/fr/shop/cartes-cpu/arduino-fr/arduino-gbx00066/

[30] Lycee_Eugene_IONESCO_TSTI2D_-SIN_STI2D_- la page numéro 7

[31] Le grand livre d'Arduino Erik Bartmann la page numéro 13

[32] https://www.crepp.org/technique/informatique/cartes-embarquees/esp8266/

[33] https://www.vs-elec.fr/fr/communication/144-robodyn-module-wifi-esp-07-esp8266-8mb-6970622931980.html

[34] http://f-leb.developpez.com/tutoriels/arduino/esp8266/debuter/ le 2019-05-20 01:58:53

[35] https://projetasgarddiy.fr/2018/07/18/flasher-un-esp-01-avec-espeasy-mega/ le 2019-05-20 à 01:42:31

[36] Shenzhen Ai-thinker Technology Co,2017.

[37] https://ouilogique.com/esp8266-01/ le 2019-05-21 à 21:54:12

[38] https://www.instructables.com/id/Beginners-Guide-to-ESP8266-and-Tweeting-Using-ESP8/

Le 2019-05-19 à 04:16:51

[39] https://www.moussasoft.com/product/module-rfid-rc522-lecteur-rfid le 2019-05-22 à 02:31:10

[40] https://lastminuteengineers.com/how-rfid-works-rc522-arduino-tutorial/ le 2019-05-22 à 02:31:10

[41]https://www.engineersgarage.com/electronic-components/16x2-lcd-module-datasheet le 2019-05-20 02:33:41

[42] Mantech Electronics (Pty), A MOBICON Company.

[43] http://osoyoo.com/2017/09/13/photoresistor-sensor-module/ le 2019-05-23 à 22:34:55

[44] http://osoyoo.com/2018/11/15/arduino-lesson-mq-5-gas-sensor/ le 2019-05-24 à 01:58:31

[45] http://osoyoo.com/2018/01/17/arduino-lesson-flame-sensor/ le 2019-05-24 à 02:10:36

[46] http://osoyoo.com/2018/03/15/arduino-lesson-dht22-humidity-and-temperature-sensor/ 2019-05-22 02:05:09

[47] http://osoyoo.com/2017/07/23/arduino-lesson-ultrasonic-sensor-hc-sr04/ le 2019-05-22 à 03:11:25

[48] http://osoyoo.com/2017/09/13/arduino-lesson-4x4-matrix-keypad/ le 2019-05-23 à 02:05:34

[49] http://osoyoo.com/2017/07/27/arduino-lesson-pir-motion-sensor/ le 2019-05-24 à 02:29:57

[50] http://osoyoo.com/2017/05/05/buzzer-5v-breadboard-friendlytmb12a05/ le 2019-05-25 à 00:43:54

[51] https://www.locoduino.org/spip.php?article210 le 2019-05-24 à 15:00:06

[52] https://eskimon.fr/tuto-arduino-103-le-logiciel le 2019-05-24 à 02:59:29

[53] http://www.circuitstoday.com/proteus-software-introduction le 2019-05-25 à 01:28:00

[54] http://appinventor.mit.edu/explore/about-us.html le 2019-05-24 à 02:57:03

[55] https://www.androidauthority.com/android-app-inventor-open-source-by-google-and-mit-is-here-47357/ le 2019-05-24 à 02:57:15

Annexe

Annexes

- **Le programme de l'Arduino Méga :**

```
////////////////////////////////////////////////////////////////////////////*/
#include <Servo.h>                                                    //--
#include <LiquidCrystal_I2C.h>                                        //--
#include <Wire.h>                                                     //--
#include "DHT.h"                                                      //--
#include <Keypad.h>                                                   //--
#include <SPI.h>                                                      //--
#include <MFRC522.h>                                                  //--
Servo PorteMaison;                                                    //--
Servo rideau ;                                                        //--
//-------------millis-------------------------------------------------------------
unsigned long ultrasonic_timer = 0;                                   //--
int mask1=0;                                                          //--
unsigned long flame_timer = 0;                                        //--
int mask2 = 0;                                                        //--
unsigned long home_door = 0;                                          //--
int mask3 = 0;                                                        //--
unsigned long redLed = 0;                                             //--
int mask4 = 0;                                                        //--
unsigned gas_timer = 0;                                               //--
int mask5 = 0;                                                        //--
//---------------multi-task-------------------------------------------------------------------------------------------
---------
long timeCounter; // milliseconds counter                                                    //--
long timePassed; // The actual milliseconds time                                            //--
```

```
// We need a "loop" doing event 1 every 10 ms, one doing event 2 every 5 ms
//--
// and one doing event 3 every 25 ms                                    //--
long delay1 = 40;    // ms for cycle of event 1                          //--
long delay2 = 30;    // ms for cycle of event 2                          //--
long delay3 = 25;    // ms for cycle of event 3                          //--
long delay4 = 35;    // ms for cycle of event 4                          //--
long delay5 = 45;    // ms for cycle of event 5                          //--
long delay6 = 25;    // ms for cycle of event 6                          //--
long delay7 = 25;    // ms for cycle of event 7                          //--
long delay8 = 25;    // ms for cycle of event 8                          //--
long delay9 = 10;    // ms for cycle of event 9                          //--
// Counters for the number of triggered events before the
//--
// corresponding event starts                                           //--
int countEvent1, countEvent2, countEvent3, countEvent4, countEvent5, countEvent6,
countEvent7,countEvent8,countEvent9;  //--
int TRIGGER = 5;    // ms to trigger the milliseconds timer
//--
int MAX1 = 2;       // Number or triggered milliseconds to start event1
//--
int MAX2 = 1;       // Number or triggered milliseconds to start event2
//--
int MAX3 = 5;       // Number or triggered milliseconds to start event3
//--
int MAX4 = 7;       // Number or triggered milliseconds to start event4
//--
int MAX5 = 9;       // Number or triggered milliseconds to start event5
//--
int MAX6 = 11;      // Number or triggered milliseconds to start event6
//--
int MAX7 = 13;      // Number or triggered milliseconds to start event7
//--
```

```
int MAX8 = 15;      // Number or triggered milliseconds to start event8
//--
int MAX9 = 17;      // Number or triggered milliseconds to start event9
//--
//--------------------RFID_Garage-------------------------------------------------------------------------------------
#define SS_PIN 53                                                   //--
#define RST_PIN 49                                                  //--
#define LED_G 38 //define green LED pin                             //--
#define LED_R 47 //define red LED                                   //--
#define LED_O 40 // define oronge led                               //--
MFRC522 mfrc522(SS_PIN, RST_PIN);   // Create MFRC522 instance.     //--
Servo myServo; //define servo name                                  //--
int posg ;                                                  //--
int ledState = LOW;                                                 //--
unsigned long previousMillis = 0;                                   //--
const long interval = 300;                                          //--
//----------------------porte_d'entrer-----------------------------------------------------------------------------
int character = 0;                                                  //--
int activated =0;                                                   //--
char Str[16] = {' ', ' ', ' ', ' ', ' ', ' ', '-', '*', '*', '*', ' ', ' ', ' ', ' ', ' ', ' '};   //--
int i=0;                                                            //--
//Pins                                                              //--
//Keypad config                                                     //--
const byte ROWS = 4; //four rows                                    //--
const byte COLS = 4; //four columns                                 //--
//define the cymbols on the buttons of the keypads                  //--
char hexaKeys[ROWS][COLS] = {                                       //--
{'1','2','3','A'},                                                  //--
{'4','5','6','B'},                                                  //--
{'7','8','9','C'},                                                  //--
```

```
{'*','0','#','D'}                                                    //--
};                                                                   //--
byte rowPins[ROWS] = {9, 8, 7, 6}; //connect to the row pinouts of the keypad
//--
byte colPins[COLS] = {5, 4, 3, 2}; //connect to the column pinouts of the keypad
//--
//initialize an instance of class NewKeypad                          //--
Keypad customKeypad = Keypad( makeKeymap(hexaKeys), rowPins, colPins, ROWS, COLS);
//--
//--------------LED-----------------------------------------------------------------------------------
int danger_LED= 45;                                                  //--
int jardin_LED= 43;                                                  //--
int gaz_LED=30;                                                      //--
int flame_LED = 31;                                                  //--
//-----------Buzzers-------------------------------------------------
int buzzer_sonic = 32;                                               //--
int buzzer_gaz = 33;                                                 //--
int buzzer_flame = 34;                                               //--
//---------ventilo-------------------------------------------------
int ventilo_GAZ=35;                                                  //--
int climatisation = 36;                                              //--
//-----------Variable_Capteurs-----------------------------------------
const int trigPin=37;                                                //--
const int echoPin=39;                                                //--
long duration, cm;                                                   //--
int distance;                                                        //--
int pos = 0;                                                         //--
int angl = 0;                                                        //--
int val = 0;                                                         //--
int MQ5 = A0;                                                        //--
int light_cap = A3;                                                  //--
```

```
int flame_sensor = 41;                                  //--
int flame_detect ;                                      //--
LiquidCrystal_I2C lcd(0x27, 16, 2);                     //--
#define DHTPIN A2                                       //--
#define DHTTYPE DHT22   // DHT 22  (AM2302)              //--
int maxHum = 60;                                        //--
int maxTemp = 30;                                       //--
DHT dht(DHTPIN, DHTTYPE);                               //--
//---------------------------PIR1-------------------------------------------------
int pir1 = 25 ;                                         //--
int couloir_led = 29;                                   //--
int val_couloir = 0;                                    //--
//---------------------------PIR2-------------------------------------------------
int pir2 = 23 ;                                         //--
int salle_de_bain_led = 27;                             //--
int val_salle = 0;                                      //--
//--------------------------------------------------------------------------------
void setup() {                                          //--
// put your setup code here, to run once:               //--
pinMode(23, INPUT);                                     //--
pinMode(25, INPUT);                                     //--
pinMode(27, OUTPUT);                                    //--
pinMode(29, OUTPUT);                                    //--
pinMode(45, OUTPUT);                                    //--
pinMode(43, OUTPUT);                                    //--
pinMode(30, OUTPUT);                                    //--
pinMode(31, OUTPUT);                                    //--
pinMode(32, OUTPUT);                                    //--
pinMode(33, OUTPUT);                                    //--
pinMode(34, OUTPUT);                                    //--
```

```
pinMode(35, OUTPUT);                                    //--
pinMode(36, OUTPUT);                                    //--
pinMode(trigPin, OUTPUT);                               //--
pinMode(echoPin, INPUT);                                //--
pinMode(A0, INPUT);                                     //--
pinMode(A3, INPUT);                                     //--
pinMode(A2, INPUT);                                     //--
pinMode(41, INPUT);                                     //--
pinMode(42, INPUT);                                     //--
pinMode(44, INPUT);                                     //--
Serial.begin(9600);   // initialize serial for debugging          //--
dht.begin();                                            //--
lcd.begin();                                            //--
SPI.begin();      // Initiate  SPI bus                  //--
mfrc522.PCD_Init();   // Initiate MFRC522                 //--
pinMode(LED_G, OUTPUT);                                 //--
pinMode(LED_R, OUTPUT);                                 //--
pinMode(LED_O, OUTPUT);                                 //--
Serial.println("Put your card to the reader...");       //--
Serial.println();                                       //--
myServo.attach(10); //servo pin                         //--
myServo.write(posg); //servo start position             //--
rideau.attach(12); //servo pin                          //--
rideau.write(0); //servo start position                 //--
PorteMaison.attach(11); //servo pin                      //--
PorteMaison.write(0); //servo start position            //--
//------------------------multi-task-----------------------------------------
timeCounter = millis(); // read the actual internal time  //--
timePassed = 0; // we start with no time passed          //--
countEvent1 = 0;                                        //--
```

```
countEvent2 = 0;                                    //--
countEvent3 = 0;                                    //--
countEvent4 = 0;                                    //--
countEvent5 = 0;                                    //--
countEvent6 = 0;                                    //--
countEvent7 = 0;                                    //--
countEvent8 = 0;                                    //--
countEvent9 = 0;                                    //--
ultrasonic_timer = millis();                         //--
flame_timer = millis();                             //--
home_door = millis();                               //--
}                                                   //--
//-------------------------------------------------------------------------------------------
void loop() {                                                   //--
// Read the actual time                                         //--
timePassed = millis() - timeCounter;                                    //--
// The smallest unit of time that should trigger an event is for event 2 (5 ms)         //--
// so our trigger should count this smaller unit                        //--
// Check if the time passed after last reading is the trigger time              //--
if(timePassed >= TRIGGER) {                                     //--
// It's time to manage the events                               //--
if(++countEvent1 == MAX1){                                      //--
// Reset the trigger and start the event                        //--
door();                                                 //--
countEvent1 = 0;                                        //--
} // Event 1                                             //--
//-------------------------------------------------------------------------------------------//--
// It's time to manage the events                               //--
if(++countEvent2 == MAX2){                                      //--
// Reset the trigger and start the event                        //--
```

```
ultrasonic();                                                    //--
countEvent2 = 0;                                                 //--
} // Event2                                                      //--
//------------------------------------------------------------------------------------------//--
// It's time to manage the events                                //--
if(++countEvent3 == MAX3){                                       //--
// Reset the trigger and start the event                         //--
garage();                                                        //--
if ( mask4 == 1&& millis()- redLed>=500)                         //--
{                                                                //--
digitalWrite(LED_R, LOW);                                        //--
mask4=0;                                                         //--
}                                                                //--
countEvent3 = 0;                                                 //--
} // Event3                                                      //--
//------------------------------------------------------------------------------------------//--
// It's time to manage the events                                //--
if(++countEvent4 == MAX4){                                       //--
// Reset the trigger and start the event                         //--
dht22_temp();                                                    //--
countEvent4 = 0;                                                 //--
} // Event4 ???                                                  //--
//------------------------------------------------------------------------------------------//--
// It's time to manage the events                                //--
if(++countEvent5 == MAX5){                                       //--
// Reset the trigger and start the event                         //--
jardin_rid();                                                    //--
countEvent5 = 0;                                                 //--
} // Event5                                                      //--
//------------------------------------------------------------------------------------------//--
```

```
// It's time to manage the events                                    //--
if(++countEvent6 == MAX6){                                           //--
// Reset the trigger and start the event                             //--
flame();                                                          //--
countEvent6 = 0;                                                     //--
} // Event6                                                          //--
//-------------------------------------------------------------------------------------//--
// It's time to manage the events                                    //--
if(++countEvent7 == MAX7){                                           //--
// Reset the trigger and start the event                             //--
gaz_cap();                                                        //--
countEvent7 = 0;                                                     //--
} // Event7                                                          //--
//-------------------------------------------------------------------------------------//--
// It's time to manage the events                                    //--
if(++countEvent8 == MAX8){                                           //--
// Reset the trigger and start the event                             //--
couloir();                                                        //--
countEvent8 = 0;                                                     //--
} // Event8                                                          //--
//-------------------------------------------------------------------------------------//--
// It's time to manage the events                                    //--
if(++countEvent9 == MAX9){                                           //--
// Reset the trigger and start the event                             //--
salle_de_bain();                                                  //--
countEvent9 = 0;                                                     //--
} // Event9                                                          //--
} // Trigger                                                         //--
//-------------------------------------------------------------------------------------//--
} //--end void loop--//
```

```
//-----------------------------------------------------------------
void dht22_temp ()                                          //--
{                                                           //--
float h = dht.readHumidity();                               //--
// Read temperature as Celsius                              //--
float t = dht.readTemperature();                            //--
if(h > maxHum || t > maxTemp) {                             //--
digitalWrite(climatisation, HIGH);                          //--
} else {                                                    //--
digitalWrite(climatisation, LOW);                           //--
}                                                           //--
//--------------afficher la temp sur lcd-----------------------------------------
lcd.setCursor(0,0); //Setting the cursor at the start of the LCD            //--
lcd.print(" Temp: "); //Writing the temperature on the LCD                  //--
lcd.print(t); //Writing the value of temperature on the LCD                 //--
lcd.print(" C");                                                            //--
lcd.setCursor(0,1); //setting the cursor at next line                       //--
lcd.print(" Humd: "); //Writing the humidity on the LCD display             //--
lcd.print(h); //Writing the humidity value stored in hum on the LCD display //--
lcd.print(" %");                                                            //--
}                                                                           //--
//-------------------------------------------------------------------------------------
void ultrasonic()                                                           //--
{                                                                           //--
digitalWrite(trigPin, LOW);                                                 //--
delayMicroseconds(5);                                                       //--
digitalWrite(trigPin, HIGH);                                                //--
delayMicroseconds(10);                                                      //--
digitalWrite(trigPin, LOW);                                                 //--
pinMode(echoPin, INPUT);                                                    //--
```

```
duration = pulseIn(echoPin, HIGH);                                    //--
// Convert the time into a distance                                   //--
cm = (duration/2) / 29.1;     // Divide by 29.1 or multiply by 0.0343          //--
if ( cm < 20){                                                        //--
digitalWrite (danger_LED, HIGH);                                      //--
// ...for 1sec                                                        //--
if(millis()-ultrasonic_timer >=1000)                                  //--
{                                                                     //--
tone(buzzer_sonic,1000);                                              //--
ultrasonic_timer=millis();                                            //--
mask1=1;                                                              //--
}                                                                     //--
if(millis()-ultrasonic_timer>=500&&mask1==1)                          //--
{                                                                     //--
noTone(buzzer_sonic);                                                 //--
mask1=0;                                                              //--
}                                                                     //--
}                                                                     //--
else {                                                                //--
digitalWrite (danger_LED,LOW);                                        //--
noTone(buzzer_sonic);                                                 //--
}                                                                     //--
}                                                                     //--
//-------------------------------------------------------------------------------
void jardin_rid()                                                     //--
{                                                                     //--
val = analogRead(A1);                                                 //--
if ( val < 60){                                                       //--
for ( pos ; pos<=110 ; pos++)                                         //--
{                                                                     //--
```

```
rideau.write(pos);                                  //--
delay(20);                                          //--
}                                                   //--
}                                                   //--
else{                                               //--
for ( pos ; pos>=0 ; pos--)                                 //--
{                                                   //--
rideau.write(pos);                                  //--
delay(20);                                          //--
}                                                   //--
}                                                   //--
if(val>300){                                                //--
digitalWrite(jardin_LED, HIGH);                                     //--
}                                                   //--
else{                                               //--
digitalWrite(jardin_LED, LOW);                                      //--
}                                                   //--
}                                                   //--
//-------------------------------------------------------------------------------
void flame ()                                               //--
{                                                           //--
flame_detect = digitalRead(flame_sensor) ;                                  //--
if (flame_detect == 0)                                              //--
{                                                           //--
if (millis()-flame_timer>=700)                                          //--
{                                                           //--
digitalWrite(buzzer_flame, HIGH);                                       //--
digitalWrite(flame_LED, HIGH);                                          //--
flame_timer=millis();                                               //--
mask2=1;                                                    //--
```

```
}                                                   //--
if (millis()-flame_timer>=300&&mask2==1)                    //--
{                                                   //--
digitalWrite(flame_LED, LOW);                               //--
digitalWrite(buzzer_flame,LOW);                             //--
mask2=0;                                            //--
}                                                   //--
}                                                   //--
else {                                              //--
digitalWrite(buzzer_flame,LOW);                             //--
digitalWrite(flame_LED,LOW);                                //--
}                                                   //--
}                                                   //--
//--------------------------------------------------------------------------------
void gaz_cap ()                                             //--
{                                                   //--
int temp;                                           //--
temp=analogRead(MQ5);                                       //--
if(temp >270)                                       //--
{                                                   //--
if(millis()-gas_timer >=1500){                              //--
digitalWrite (gaz_LED,HIGH);                                //--
tone (buzzer_gaz, 700); // tone(buzpin,1000)   Send 1KHz sound signal...     //--
gas_timer= millis();                                        //--
mask5 = 1;                                          //--
}                                                   //--
if(millis()-gas_timer >=1000 && mask5 == 1){                //--
digitalWrite(gaz_LED,LOW);                                  //--
noTone(buzzer_gaz);     // noTone Stop sound...             //--
gas_timer= millis;                                          //--
```

```
mask5=0;                                                    //--
digitalWrite (ventilo_GAZ, HIGH);                           //--
}                                                           //--
}                                                           //--
else                                                        //--
{                                                           //--
digitalWrite(gaz_LED,LOW);                                  //--
noTone(buzzer_gaz);     // Stop sound...                    //--
digitalWrite(ventilo_GAZ,LOW);                              //--
}                                                           //--
}                                                           //--
//-------------------------------------------------------------------------------------------------
void door()                                                 //--
{                                                           //--
///////////////EMERGENCY OPEN/CLOSE////////                  //--
///////////////KEYPAD OPEN/CLOSE////////////                 //--
char customKey = customKeypad.getKey(); //this function reads the presed key   //-
-
if (customKey){                                             //--
if (character ==0)                                          //--
{                                                           //--
Str[6]= customKey;                                          //--
}                                                           //--
if (character ==1)                                          //--
{                                                           //--
Str[7]= customKey;                                          //--
}                                                           //--
if (character ==2)                                          //--
{                                                           //--
Str[8]= customKey;                                          //--
```

```
}                                                                    //--
if (character ==3)                                                   //--
{                                                                    //--
Str[9]= customKey;                                                   //--
}                                                                    //--
if (character ==4)                                                   //--
{                                                                    //--
Str[10]= customKey;                                                  //--
activated=1;                                                         //--
}                                                                    //--
character=character+1;                                               //--
}                                                                    //--
if (activated == 1)                                                  //--
{                                                                    //--
/*Change your password below!!!                                      //--
Change each of Str[6], Str[7], Str[8], Str[9]*/                      //--
if(Str[10]='A' && character==5 && Str[6]=='3' && Str[7]=='0' && Str[8]=='0' && Str[9]=='7' )
//--
{                                                                    //--
//myservo.write(mot_max);                                            //--
for (angl ; angl<=110 ; angl++)                                      //--
{                                                                    //--
PorteMaison.write(angl);                                             //--
delay(20);                                                           //--
}                                                                    //--
delay(4000);                                                         //--
for (angl ; angl>=0 ; angl--)                                        //--
{                                                                    //--
PorteMaison.write(angl);                                             //--
delay(20);                                                           //--
```

```
}                                                                                    //--
}                                                                                    //--
activated = 2;                                                                       //--
}                                                                                    //--
if (activated == 2)                                                                  //--
{                                                                                    //--
activated = 0;                                                                       //--
character=0;                                                                         //--
}                                                                                    //--
}                                                                                    //--
//-------------------------------------------------------------------------------------------------------------
void garage()                                                                        //--
{                                                                                    //--
// Look for new cards                                                                //--
if ( ! mfrc522.PICC_IsNewCardPresent())                                              //--
{                                                                                    //--
return;                                                                              //--
}                                                                                    //--
// Select one of the cards                                                           //--
if ( ! mfrc522.PICC_ReadCardSerial())                                                //--
{                                                                                    //--
return;                                                                              //--
}                                                                                    //--
//Show UID on serial monitor                                                         //--
Serial.print("UID tag :");                                                           //--
String content= "";                                                                  //--
byte letter;                                                                         //--
for (byte i = 0; i < mfrc522.uid.size; i++)                                          //--
{                                                                                    //--
Serial.print(mfrc522.uid.uidByte[i] < 0x10 ? " 0" : " ");                            //--
```

```
Serial.print(mfrc522.uid.uidByte[i], HEX);                                    //--
content.concat(String(mfrc522.uid.uidByte[i] < 0x10 ? " 0" : " "));                    //--
content.concat(String(mfrc522.uid.uidByte[i], HEX));
}
Serial.println();
Serial.print("Message : ");
content.toUpperCase();
//change here the UID of the card/cards that you want to give access
if (content.substring(1) == "14 41 DA 1F" )
{
Serial.println("Authorized access");
Serial.println();
delay(500);
digitalWrite(LED_G, HIGH);
delay(1000);
digitalWrite(LED_G,LOW);
delay(1000);
for ( posg = 0 ; posg<110 ; posg++)
{
unsigned long currentMillis = millis();
if (currentMillis - previousMillis >= interval) {
// save the last time you blinked the LED
previousMillis = currentMillis;
// if the LED is off turn it on and vice-versa:
if (ledState == LOW) {
ledState = HIGH;
} else {
ledState = LOW;
}
// set the LED with the ledState of the variable:
```

```
digitalWrite(LED_O, ledState);
}
myServo.write(posg);
delay(20);
}
digitalWrite(LED_G,HIGH);
delay(3000);
digitalWrite(LED_G,LOW);
for (posg = 110 ; posg >=0; posg--)
{
unsigned long currentMillis = millis();
if (currentMillis - previousMillis >= interval) {
// save the last time you blinked the LED
previousMillis = currentMillis;
// if the LED is off turn it on and vice-versa:
if (ledState == LOW) {
ledState = HIGH;
} else {
ledState = LOW;
}
// set the LED with the ledState of the variable:
digitalWrite(LED_O, ledState);
}
myServo.write(posg);
delay(20);
}
}
else
{  mask4=1;
Serial.println(" Access denied");
```

```
if (millis()- redLed>=50)
{
digitalWrite(LED_R, HIGH);
redLed=millis();
}
}
}
//-----------------------------------------------------------------------------------------------
void couloir ()
{
val_couloir = digitalRead(pir1);    // read sensor value
if (val_couloir == 1) {    // check if the sensor is HIGH
digitalWrite (couloir_led,HIGH);
}
else {
digitalWrite (couloir_led,LOW);
}
}
//-----------------------------------------------------------------------------------
void salle_de_bain ()
{
val_salle = digitalRead(pir2);    // read sensor value
if (val_salle == HIGH) {          // check if the sensor is HIGH
digitalWrite (salle_de_bain_led,HIGH);
}
else {
digitalWrite (salle_de_bain_led,LOW);
}
}
//-----------------------------------------------------------------------------------
```

- **Le progremme de l’arduino uno et esp8266-01 :**

```
#include "WiFiEsp.h"
// Emulate Serial1 on pins 6/7 if not present
#ifndef HAVE_HWSERIAL1
#include "SoftwareSerial.h"
SoftwareSerial Serial1(6, 7); // RX, TX
#endif

char ssid[] = "DJAWEB_6C34";            // your network SSID (name)
char pass[] = "94D723866C34";        // your network password
int status = WL_IDLE_STATUS;

int Chambre1_LED = 9;
int Chambre2_LED = 10;
int Salon_LED = 11;
int Cuisine = 12;

WiFiEspServer server(80);

// use a ring buffer to increase speed and reduce memory allocation
RingBuffer buf(32);

void setup()
{
pinMode(10, OUTPUT);
pinMode(11, OUTPUT);
pinMode(4, OUTPUT);
pinMode(12, OUTPUT);
```

```
Serial.begin(115200);   // initialize serial for debugging
Serial1.begin(9600);    // initialize serial for ESP module
WiFi.init(&Serial1);    // initialize ESP module

// check for the presence of the shield
if (WiFi.status() == WL_NO_SHIELD) {
Serial.println("WiFi shield not present");
// don't continue
while (true);
}

// attempt to connect to WiFi network
while (status != WL_CONNECTED) {
Serial.print("Attempting to connect to WPA SSID: ");
Serial.println(ssid);
// Connect to WPA/WPA2 network
status = WiFi.begin(ssid, pass);
}
Serial.println("You're connected to the network");
printWifiStatus();
// start the web server on port 80
server.begin();
}
//--------------------------------------------------------------------------------------------
void loop()
{
leds();
cuis();
}
//--------------------------------------------------------------------------------------------
```

```
void leds ()
{
WiFiEspClient client = server.available();  // listen for incoming clients

if (client) {                             // if you get a client,
Serial.println("New client");             // print a message out the serial port
buf.init();                               // initialize the circular buffer
while (client.connected()) {              // loop while the client's connected
if (client.available()) {                 // if there's bytes to read from the client,
char c = client.read();                   // read a byte, then
buf.push(c);                              // push it to the ring buffer

// printing the stream to the serial monitor will slow down
// the receiving of data from the ESP filling the serial buffer
//Serial.write(c);

// you got two newline characters in a row
// that's the end of the HTTP request, so send a response
if (buf.endsWith("\r\n\r\n")) {
sendHttpResponse(client);
break;
}
salon();
chamb1();
chamb2();
}
}
// close the connection
client.stop();
Serial.println("Client disconnected");
```

```
}


}
//-------------------------------------------------------------------------------------------
void chamb1()
{
// Check to see if the client request was "GET /H" or "GET /L":
if (buf.endsWith("GET /A"))
digitalWrite(Chambre1_LED, HIGH);   // turn the LED on (HIGH is the voltage level)
}
else if (buf.endsWith("GET /B")) {

digitalWrite(Chambre1_LED, LOW);    // turn the LED off by making the voltage LOW
}
}
//-----------------------------------------------------
void chamb2()
{
// Check to see if the client request was "GET /H" or "GET /L":
if (buf.endsWith("GET /C")) {
digitalWrite(Chambre2_LED, HIGH);   // turn the LED on (HIGH is the voltage level)
}
else if (buf.endsWith("GET /D")) {
digitalWrite(Chambre2_LED, LOW);    // turn the LED off by making the voltage LOW
}
}
//-------------------------------------------------
void salon()
{
if (buf.endsWith("GET /E")) {
```

```
digitalWrite(Salon_LED, HIGH);   // turn the LED on (HIGH is the voltage level)
}
else if (buf.endsWith("GET /F")) {
digitalWrite(Salon_LED, LOW);    // turn the LED off by making the voltage LOW
}
}
//------------------------------------------------------------
void cuis()
{
WiFiEspClient client = server.available();  // listen for incoming clients
if (client) {                               // if you get a client,
Serial.println("New client");             // print a message out the serial port
buf.init();                               // initialize the circular buffer
while (client.connected()) {              // loop while the client's connected
if (client.available()) {                 // if there's bytes to read from the client,
char c = client.read();                   // read a byte, then
buf.push(c);                              // push it to the ring buffer
// printing the stream to the serial monitor will slow down
// the receiving of data from the ESP filling the serial buffer
// Serial.write(c);
// you got two newline characters in a row
// that's the end of the HTTP request, so send a response
if (buf.endsWith("\r\n\r\n")) {
sendHttpResponse1(client);
break;
}
// Check to see if the client request was "GET /H" or "GET /L":
if (buf.endsWith("GET /G")) {

digitalWrite(Cuisine, HIGH);   // turn the LED on (HIGH is the voltage level)
```

```
}
else if (buf.endsWith("GET /H")) {
digitalWrite(Cuisine, LOW);    // turn the LED off by making the voltage LOW
}
}
}
// close the connection
client.stop();
Serial.println("Client disconnected");
}
}
//-----------------------------------------------------------------------------------------------
//-----------------------------------------------------------------------------------------------
void sendHttpResponse(WiFiEspClient client)
{
// HTTP headers always start with a response code (e.g. HTTP/1.1 200 OK)
// and a content-type so the client knows what's coming, then a blank line:
client.println("HTTP/1.1 200 OK");
client.println("Content-type:text/html");
client.println();
}
void sendHttpResponse1(WiFiEspClient client)
{
// HTTP headers always start with a response code (e.g. HTTP/1.1 200 OK)
// and a content-type so the client knows what's coming, then a blank line:
client.println("HTTP/1.1 200 OK");
client.println("Content-type:text/html");
client.println();
}
//-----------------------------------------------------------------------------------------------
```

```
void printWifiStatus()
{
// print the SSID of the network you're attached to
Serial.print("SSID: ");
Serial.println(WiFi.SSID());
// print your WiFi shield's IP address
IPAddress ip = WiFi.localIP();
Serial.print("IP Address: ");
Serial.println(ip);
// print where to go in the browser
Serial.println();
Serial.print("To see this page in action, open a browser to http://");
Serial.println(ip);
Serial.println();
}
```