

REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE
MINISTERE DE L'ENSEIGNEMENT SUPERIEUR ET DE LA RECHERCHE
SCIENTIFIQUE
UNIVERSITE A.MIRA-BEJAIA
FACULTE DE TECHNOLOGIE
DEPARTEMENT GENIE ELECTRIQUE



MEMOIRE
EN VUE DE L'OBTENTION DU DIPLOME DE
MASTER

Domaine : Sciences et Technologies
Filière : Télécommunications
Spécialité : Réseaux et Télécommunications

Présenté par

Haouchine Imene
Ait Mouhoub Sabrina

Thème

**Etude de l'IoT et réalisation d'une maison
intelligente**

Soutenu le 07 juillet 2019

Devant le Jury

Président	Mr. ALLICHE.A
Examineur	Mr. HADJI. S
Encadrant	Mr. BARRAH. S

Année Universitaire : 2018/2019

★Remerciements★

Nous remercions tout d'abord ALLAH qui nous a armés de courage, Pouvoir et la patience pour finaliser ce travail.

Nous remercions, notre encadreur Mr. BARRAH.S. Pour son encadrement de grande qualité scientifique, ses précieux conseils, et ses encouragements qui ont contribué efficacement à l'avancement de ce travail.

Nous remercions Mr. SADI. Pour son aide, ses conseils et ses encouragements.

Nous adressons de même nos remerciements à Mr HADJIS et Mr. ALLICHE.A, pour l'honneur qu'ils nous 'ont fait en acceptant d'encadré ce thème.

En fin, nous adressons nos plus sincères remerciements à nos familles : nos parents et tous nos proches et amis, qui nous ont accompagnées, aidé, soutenu et encouragé tout au long de la réalisation de ce mémoire.

Dédicaces

Je dédie ce modeste travail :

*Aux êtres qui me sont les plus chers ma mère
et mon père. Que Dieu préserve bonne santé et
longue vie. Qui ont tous fait pour m'encourager
durant les années de mes études*

A mon cher frère Fares

A mes chères sœurs :

Fatima –Wissame– Sabrina

A mes chers amis :

Sabrina – Kenza – Doucha – Dalila

Micipsa – Redouan

Mes enseignants durant tout mon cursus

Merci à tous.

HAOUCHINE IMENE

★ Dédicaces★

Je dédie ce modeste travail :

Aux êtres qui me sont les plus chers ma mère Ait mouhoub Hadia et mon père Ait mouhoub Slimane. Que Dieu préserve bonne santé et longue vie. Qui ont tous fait pour m'encourager durant les années de mes études.

A mes chers frères (FARID, MOUNIR).

A ma chère sœur (NAWEL).

A mon cher oncle MOURAD.

A ma chère Tante FARIDA, et DJAMILA.

A mes Grands Parents que Dieu les protège et à toute ma famille.

Sommaire

Remerciement	
Dédicace	
Sommaire	
Liste des Figures	
Liste des Tableaux	
Liste des abréviations	
Introduction générale	1

Chapitre I : Étude des réseaux à base de l'internet des Objets

Introduction à l'internet des objets.....	2
I.1 Définition d'internet des objets.....	2
I.2 Domaines d'application de l'IoT	2
I.2.1 Santé	2
I.2.2 Transport	3
I.2.3 L'agriculture.....	3
I.2.4 L'environnement intelligent.....	3
I.2.5 Les Villes intelligentes	3
I.2.6 La Domotique.....	3
I.3 L'évolution du Web	4
I.3.1 Web 1.0	4
I.3.2 Web 2.0	4
I.3.3 Web 3.0	5
I.3.4 Web 4.0	5
I.3.5 Web 5.0	5
I.4 Architecture et Ecosystème d'un réseau IoT	6
I.4.1 l'écosystème de l'IoT	6
I.4.2 Architecture d'un Réseau IoT	7
I.5 Les technologies de communication des objets connectés	10
I.5.1 Les technologies de courte portée	10

I.5.1.1 La RFID	10
I.5.1.2 Bluetooth	10
I.5.1.3 Zigbee	10
I.5.2 Les technologies de moyenne portée	11
I.5.2.1 Z-wave	11
I.5.2.2 Wi-Fi.....	12
I.5.2.3 Bluetooth Low Energy.....	12
I.5.3 Les technologies de longue portée	13
I.5.3.1 Réseau Cellulaires mobiles.....	13
I.5.3.2 Sigfox.....	13
I.5.3.3 LoRa	13
I.5.3.4 LTE-M	14
I.5.3.5 NB_IOT	14
I.5.3.6 Réseaux Propriétaires	14
I.6 La sécurité en internet des objets	15
I.6.1 La protection des systèmes et des appareils IOT	15
I.7 Comparaison entre M2M et l'IoT	16
I.8 Avantages et Inconvénients du réseau IoT	17
I.8.1 Les Avantages	17
I.8.2 Les inconvénients.....	17
Conclusion.....	18

Chapitre II : Présentation de matériels et de logiciel

Introduction	19
II.1 La description de matériel	19
II.1.1 La carte Arduino	19
II.1.2 La carte Arduino Mega2560	20
II.1.2.1 Les caractéristiques techniques de la carte Méga.....	20
II.1.3 Accessoires de communications avec Arduino Méga 2560	21

II.1.3.1 Module Wi-Fi ESP8266	21
II.1.3.2 Les caractéristiques du module ESP8266	22
II.1.3.3 Description de module ESP8266-01	22
II.1.4 les Capteurs	23
II.1.4.1 Les différents types des capteurs.....	24
II.1.5 les moteurs	24
II.1.6 Relais.....	24
II.1.8 buzzer.....	25
II.1.9 Les Leds	26
II.2 La description de logiciel	26
II.2.1 description de différents menus du logiciel	27
II.2.1.1 L’interface de commande de logiciel	28
II.2.1.2 Approche et utilisation du logiciel	29
II.2.2 Installer les cartes et les librairies de module ESP8266 sur l’IDE Arduino	30
II.2.3 La syntaxe du langage de programmation	34
Conclusion.....	35

Chapitre III : Programmation des capteurs et des actionneurs sur Arduino

Introduction.....	36
III.1 Capteur de distance à Ultrasons HC-SR04.....	36
III.1.1 Fonctionnement de Capteur Ultrasons HC-SR04.....	36
III.1.2 Schéma du montage sur arduino.....	37
III.1.3 Programmation du capteur Ultrasons HC-SR04	38
III.1.4 Résultats.....	39
III.2 Capteur infrarouge Passif PIR.....	39
III.2.1 Les spécifications techniques de capteur infrarouge passif.....	39
III.2.2 Fonctionnement	40
III.2.3 schéma de montage sur Arduino	40
III.2.4 Programmation du capteur infrarouge Passif PIR	42

III.2.5 Résultats.....	43
III.3 Capteur d'humidité et de température DHT11	43
III.3.1 les caractéristiques du capteur DHT11	44
III.3.2 principe de Fonctionnement	44
III.3.3 Schéma de montage sur Arduino.....	46
III.3.4 Programmation du capteur Ultrasons DHT11	46
III.3.5 Résultats.....	47
III. 4 Capteur de flamme.....	47
III.4.1 Les Caractéristiques	48
III.4.2 Principe de fonctionnement	48
III.4.3 Schéma de montage sur Arduino.....	49
III.4.4 Code de programmation de capteur de flamme.....	49
III.5 Servomoteur.....	50
III.6 Le moteur Pas à Pas.....	51
III.6.1 Caractéristiques	51
Conclusion.....	52

Chapitre IV : Réalisation d'une maison intelligente

Introduction	53
IV.1 Fabrication de la maison	53
IV.1.1 Structure de la maison	53
IV.1.2 Périphériques utilisés	54
IV.2 Automatisation de la maison	54
IV.3 Centralisation des commandes : Interface RemoteXY	65
IV.4 La maquette finale :.....	67
Conclusion.....	69
Conclusion générale	70

Annexe

Références WEBOGRAPHIE

Résumé

Chapitre I

Figure I.1: Le monde avant et après l’apparition de l’IoT2

Figure I.2 : Domaines d’utilisation de l’IOT4

Figure I.3 : Le web1.0 une conversation à sens unique4

Figure I.4 : La création des connexions bidirectionnelles dans web 2.05

Figure I.5 : Le Cloud computing faire connecte tout.....5

Figure I.6 : Principe e web 4.06

Figure I.7: Le contenu de la version de web 5.06

Figure I.8 : Les composants de l’écosystème IoT.....7

Figure I.9 : L’architectue de l’IoT7

Figure I.10 : les utilisations de la technologie Bluetooth.....10

Figure I.11 : la technologie ZigBee11

Figure I.12 : Le protocole de communication Zwave.....11

Figure I.13 : les utilisations de wifi12

Figure I.14 : les applications du protocole BLE12

Figure I.15 : La technologie de communication Sigfox13

Figure I.16 : Comparaison entre M2M et IoT.....16

Chapitre II

Figure II.1 : Les entrées et les sorties de la carte Arduino19

Figure II.2 : Module Arduino Mega256021

Figure II.3 : Types de module ESP8266.....22

Figure II.4 : Les broches de connexion d’ESP8266-0123

Figure II.5 : La chaine d’action d’un capteur23

Figure II.6 : Le relais25

Figure II.7 : Buzzer passive25

Figure II.8 : Leeds / DEL.....26

Figure II.9:L’interface de l’IDE Arduino27

Figure II.10: L’interface de l’IDE Arduino en détail28

Figure II.11: Le menu Fichier.....29

Figure II.12: Le menu Croquis30

Figure II. 13:Menu outils de sélection de la carte cible.....30

Figure II.14 : La barre d'outils	31
Figure II.15 : Sélection moniteur série	31
Figure II.16 : La voie série	32
Figure II.17 : L'adresse de saisie correspond aux libraires d'Esp8266	32
Figure II.18 : Sélection de gestionnaire de carte dans menu outils	33
Figure II.19 : L'ajoute des bibliothèques ESP8266.....	33
Figure II.20 : La fonction setup	34
Figure II.21 : La fonction loop	34
Figure II.22 : La compilation et transformation du code sur l'arduino	35

Chapitre III

Figure III.1 : Le capteur Ultrasons HC-SR04	36
Figure III.2 : Diagramme temporelle.....	37
Figure III.3 : Montage de capteur Ultrasons HC-SR04 sur la carte Arduino Méga 2560.....	38
Figure III.4 : Le code de programmation du capteur ultrasons	38
Figure III.5 : Mesure de distance par capteur Ultrasons Hc-SR04.....	39
Figure III.6 : Capteur infrarouge Passif (PIR).....	39
Figure III.7 : Fonctionnement de capteur infrarouge passif.....	40
Figure III.8 : Montage du capteur Infrarouge Passif PIR sur l'arduino Méga2560	41
Figure III.9 : Organigramme de fonctionnement de capteur PIR sur l'arduino Méga2560	41
Figure III.10 : Programme Arduino de capteur Infrarouge Passif.....	42
Figure III.11 : Détection de Mouvement par le capteur PIR.....	43
Figure III.12 : Capteur d'humidité et de température DHT11	43
Figure III.13 : Chronogramme de capteur de température et d'humidité.....	44
Figure III.14 : Le signal de départ	45
Figure III.15 : Signal de réponse	45
Figure III.16 : Lecture des données valeurs données 40bits	45
Figure III.17 : Montage du capteur d'humidité et de température DHT11	46
Figure III.18 : Programme Arduino de capteur DHT11 sur l'IDE.....	47
Figure III.19 : Les valeurs de température et d'humidité détectent par DHT11	47
Figure III.20 : Capteur de flamme	48
Figure III.21 : Montage de capteur de flamme sur arduino méga2560.....	49
Figure III.22 : Exemple de code de programmation de capteur	49
Figure III.23 : Les composants de servomoteur	50

Figure III.24 : Diagramme d’impulsions de servomoteur51

Figure III.25: Le moteur Pas à Pas51

Chapitre IV

Figure IV.1 : Illustration de la maquette utilisée dans le projet.....53

Figure IV.2 : Montage réel de moteur pas à pas55

Figure IV.3 : Montage moteur pas à pas sur Arduino Méga et le module wifi esp 8266.....55

Figure IV.4 : Interface d’ouverture et de fermeture de garage via smartphone.....56

Figure IV.5 : Montage réel d’une LED.....56

Figure IV.6 : Interface d’allumer une LED via smartphone57

Figure IV.7 : Montage réel de capteur flamme.....58

Figure IV.8 : Montage de capteur de flamme avec Arduino Méga et le module wifi esp 8266.....58

Figure IV.9 : Interface de commande de capteur de flamme.....58

Figure IV.10 : Montage de DHT11 avec Arduino Méga et le module wifi esp 826659

Figure IV.11 : Montage réel de capteur DHT1159

Figure IV.12 : Interface de contrôle de température et d’humidité60

Figure IV.13 : Montage de servomoteur avec l’Arduino Méga et l’Esp 826660

Figure IV.14 : Montage réel de servomoteur61

Figure IV.15 : Interface d’ouverture et de fermeture de fenêtre.....61

Figure IV.16: Montage du système Alarme avec l'Arduino MEGA et Esp8266.....62

Figure IV.17 : Montage réel de capteur PIR et Buzzer.....62

Figure IV.18 : Interface de commande le capteur de mouvement.....63

Figure IV.19 : Montage réel de système d’éclairage64

Figure IV.20 : Montage de relais et d’une lampe et HC-SR04 Avec Arduino Méga et Esp8266....64

Figure IV.21 : Interface d’allumage d’une lampe.....65

Figure IV.22 : Les éléments de l’interface de RemoteXY.....66

Figure IV.23 : Maquette finale de la maison intelligente67

Figure IV.24 : Montage final de la maquette réalisé avec fritzing68

Chapitre III

Tableau III.1 : Paramètres électrique de Capteur à Ultrasons HC-SR04	37
Tableau III.2 : Les spécifications techniques de capteur	44

Chapitre IV

Tableau IV.1 : Les périphériques utilisés dans les parties de maquette	54
---	----

Liste des abréviations :

A

API : Application Programming Interface

B

BLE: Bluetooth Low Energy

C

CAN : Convertisseur Analogique-Numérique

COM : Commande

D

DC : Direct Current

E

EDGE: Enhanced Data Rates for GSM Evolution

E/S : Entée/Sortie

G

GND : Ground

GPIO : General Purpose Input/Output

GPRS : General Packet Radio Service

GSM : Global System for Mobile Communications

H

HSPA : High Speed Packet Access

I

IDE : Integrated Development Environment

IC : Inter-Integrated Circuit

IOT : Internet of Things

IR : Infrarouge

ISM : Industriel, Scientifique et Médical

L

LAN : Local Area Network

LCD : Liquid Crystal Display

Led : light-emitting diode

LoRa : Long Range

LTE : Long-Term Evolution.

LTE-M : Long-Term Evolution Machine Type Communication.

M

M2M : Machine To Machine

MCU : Micro-computer Unite

N

NB_IOT: Narrowband Internet of Things.

NC : Normally Closed

NO : Normally Open

P

PCB : Printed Circuit Board

PIR : Passive Infrared

PKI : Public Key Infrastructure

PWM : Pulse Width Modulation

R

RFID : radio frequency identification

RST : Reset

RX : Réception

S

SPI : Serial Peripheral Interface

T

TCP/IP: Transmission Control Protocol/Internet Protocol

TX : Transmission

U

UHF : Ultra Hautes Fréquences.

UMTS : Universal Mobile Télécommunications System.

USB : Universal Serial Bus.

U-FL : Upper Flammable Limit

UART: Universal Asynchronous Receiver Transmitter.

UV: Ultraviolet.

W

Wifi : Wireless Fidelity

WPA : Wi-Fi Protected Access

Introduction Générale

Introduction générale

Aujourd'hui, l'Internet est devenu omniprésent, présent dans tous les coins du globe et affecte la vie humaine de manière inimaginable. Cependant le voyage est loin d'être terminé. Nous entrons maintenant dans une ère de connectivité encore plus répandue, C'est l'ère de "l'Internet of Things" abrégé en (IoT).

Cela ne se limitera pas à une simple communication mais elle s'étend sur des prises de décision qui ont un impact sur notre état d'être et notre environnement.

L'objectif de notre projet est la réalisation d'un système de commande à distance à base d'une carte Arduino pour commander des installations électriques pour la domotique (une maison intelligente).

Ce travail se répartit en quatre chapitres organisés comme suit :

- **Chapitre 1** : est consacré à définir l'internet des objets, citer ses fonctionnalités, ses différents domaines d'application et détailler par la suite l'architecture d'un écosystème IoT, ainsi une petite comparaison à la technologie machine to machine (M2M). A la fin, nous nous intéressons à ses avantages et ses inconvénients.
- **Chapitre 2** : est voué à décrire les outils matériels utilisés pour réaliser le projet final, avec une description de logiciel exploité lors de la programmation.
- **Chapitre 3** : est baptisé à programmer certains capteurs de base sur la carte Arduino et quelques actionneurs bien choisis.
- **Chapitre 4** : est dédié à la réalisation d'un système domotique sur une maquette, puis définir les parties constituant cette maquette. En parallèle l'application exploitée pour faire commander à distance à base l'arduino Méga qui sera notre Gateway pour traiter, actionné et la gestion.

Chapitre I

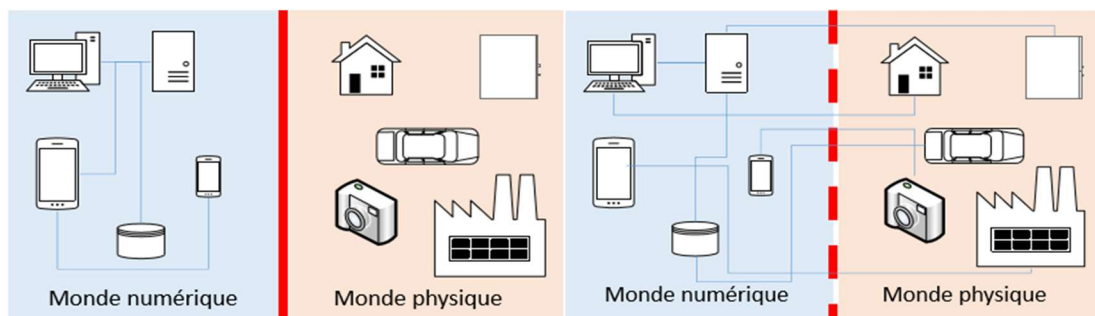
Etude des Réseaux
À base de
L'internet des objets IoT

Introduction:

Dans ce chapitre, nous allons commencer par une définition globale de l'IoT, ainsi nous présentons ses domaines d'application dans divers secteurs de la vie, ensuite nous allons décrire l'architecture de ce réseau et ses protocoles de communication, et enfin, nous citons les avantages et les inconvénients.

I.1 définition d'internet des objets :[01]

1. L'internet des objets appelé internet of things pourrait se traduire littéralement par l'internet des choses, généralement abrégé en IoT, fait référence à la connexion de périphériques à l'Internet. Les voitures, les appareils de cuisine ou tous objet peuvent collecter et échanger des données et connectés via l'IoT. Et au fur et à mesure que l'Internet des objets se développera au cours des prochaines années, d'autres appareils rejoindront cette liste (figureI.1).



FigureI.1 : Le monde avant et après l'apparition de l'IoT

I.2 Domaines d'application de l'IoT :[02]

Les champs d'applications de l'IoT sont très nombreux, elles imprègnent pratiquement tous

Les domaines de la vie quotidienne des individus, on cite quelques exemples :

I.2.1 Santé :

- ✓ Un suivi continu des patients par les professionnels ;
- ✓ Réfrigérateurs médicaux : contrôle les conditions à l'intérieur des congélateurs stockant des vaccins, médicaments et les éléments organiques ;
- ✓ Aide à la détection des chutes pour personnes âgées ou handicapées vivant indépendamment ;

Chapitre I Étude des réseaux à base de l'internet des Objets

- ✓ Surveillance de l'activité physique par la détection des petits mouvements comme la respiration et la fréquence cardiaque.

I.2.2 Transport :

- ✓ Suivi en temps réel le déplacement des populations et des moyens de transport dans le monde ;
- ✓ Le système de transport intelligent, garantissant une meilleure sécurité routière, plus de confort et de facilité de déplacement.

I.2.3 L'agriculture :

Il existe de nombreuses applications de l'IoT dans l'agriculture telles que :

- ✓ Surveiller l'humidité du sol, la vitesse du vent, le diamètre des troncs, l'infestation parasitaire et la teneur du sol ;
- ✓ Améliorer la qualité et la quantité des produits agricoles ;
- ✓ Dans le contexte de l'IoT, même les animaux seront connectés, il permet aux agriculteurs de surveiller l'état de santé des animaux et de suivre leurs déplacements.

I.2.4 L'environnement intelligent :

- ✓ Surveillance de la pollution de l'air comme le contrôle des émissions de CO₂ des usines et des gaz toxiques dans les fermes ;
- ✓ Surveillance météorologique : l'humidité, pression, vitesse de vent et pluie ;
- ✓ Étude de la qualité de l'eau dans les rivières et dans la mer pour l'admissibilité à l'usage potable.

I.2.5 Les Villes intelligentes :

- ✓ Éclairage intelligent et adapté aux conditions météorologiques ;
- ✓ L'IOT permettra une meilleure gestion de tous les réseaux qui alimentent ces villes comme l'électricité, l'eau, le gaz ;
- ✓ La gestion des parkings et diminuer les cambrements.

I.2.6 La Domotique :

- ✓ Contrôler l'électroménager, la luminance ;
- ✓ Déceler une présence par des capteurs.

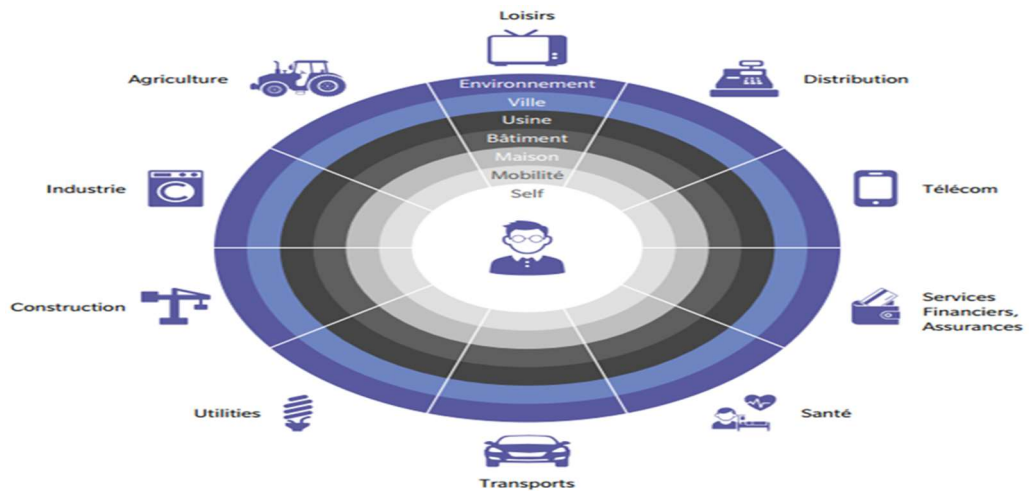


Figure I.2 : Domaines d'utilisation de l'IOT

I.3 L'évolution du Web : [03]

L'Internet et le Web sont deux choses distinctes mais liées. Comme l'Internet est un réseau porte des ordinateurs qui sont connectés, formant un réseau dans lequel ces terminaux peuvent communiquer avec n'importe quel terminal appartient à ce réseau, Le World Wide Web est l'outil d'accéder à l'information par le biais d'Internet par l'affichage des pages Web sur un navigateur, les informations pouvant être du texte, de l'audio, des vidéos...etc. on verra par la suite les changements qu'a subi le web depuis son apparition.

I.3.1 Web 1.0 :

Le Web 1.0 est la première génération du Web également appelée le Web informationnel, c'est un site Web crée par des producteurs a peu de pages Web offre seulement de la lecture des informations aux utilisateurs.



Figure I.3 : le web1.0 une conversation à sens unique

I.3.2 Web 2.0 :

Est connu sous le nom web en lecture-écriture puisqu'il permet à l'internaute de lire, écrire, collaborer et partager les informations trouvées sur le web, même modifier le contenu et interagir

Chapitre I Étude des réseaux à base de l'internet des Objets

avec d'autres utilisateurs. Egalement appelé le réseau social puisque la plupart des médias sociaux ont été créés dans le Web 2.0 tels que Facebook, YouTube.



Figure I.4 : la création des connexions bidirectionnelles dans web 2.0

I.3.3 Web 3.0 :

Le Web 3.0 est une extension combine l'intelligence humaine et artificielle des machines, il représente les connaissances de manière à permettre aux ordinateurs d'atteindre automatiquement des conclusions à l'aide de raisonnements ou de données, puis l'ordinateur peut comprendre l'historique de recherche Google d'un utilisateur, puis donner une publicité personnalisée et des suggestions.

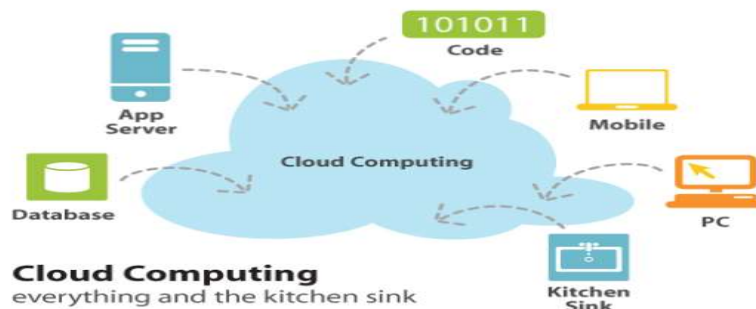


Figure I.5 : le Cloud computing faire connecte tout

I.3.4 Web 4.0 :

Le web 4.0 est le web symbiotique qui peut atteindre le niveau de l'intelligence humaine. Les ordinateurs deviendront des assistants personnels utilisant les réalités virtuelles. Il saura également que vous êtes en retard à cause du trafic et avertira vos amis ainsi que le maître de que vous serez là dans 15 min.



FigureI.6 : principe e web 4.0

I.3.5 Web 5.0 :

Aujourd'hui L'internet n'est plus une série de serveurs et de connexions, il héberge un nombre infini de systèmes sensibles. Peut-être même des souvenirs téléchargés, les émotions de personnes qui ont désormais une présence physique sur la terre. L'homme pensera à une question et la page Web s'ouvrira. Toutes sortes de paiements seront effectués en utilisant une micro puce dans le cerveau ou sur la main et tous les appareils seront connectés à Internet et seront contrôlés par les humains via des applications mobiles ou par leurs pensées.



FigureI.7 : le contenu de la version de web 5.0

I.4 Architecture et Ecosystème d'un réseau IoT :

I.4.1 l'écosystème de l'IoT : [04]

L'objet connecté est une partie essentielle de l'écosystème de l'IOT. Comme il y a d'autres parties avant l'objet et après la connectivité, il se décompose en cinq parties essentielles :

- **Chipset** : des puces et des modules électroniques qui constituent les capteurs et les transmetteurs électroniques assemblés composeront les objets connectés ;

Chapitre I Étude des réseaux à base de l'internet des Objets

- **Objet connecté** : composé de chipset, dispose d'un système d'exploitation qui permettra de gérer au mieux la collecte des données ;
- **Opérateur réseaux** : il faut relier les objets avec un opérateur réseau pour assurer la connectivité et l'échange de données de l'objet vers le destinataire ;
- **Plateforme** : stocke et analyse les données émises par les objets connectés et de les traiter afin de les rendre exploitables par les applications métiers du client final ;
- **L'Application** : l'application utilise les données d'objets connectés et les traduit en information exploitables.

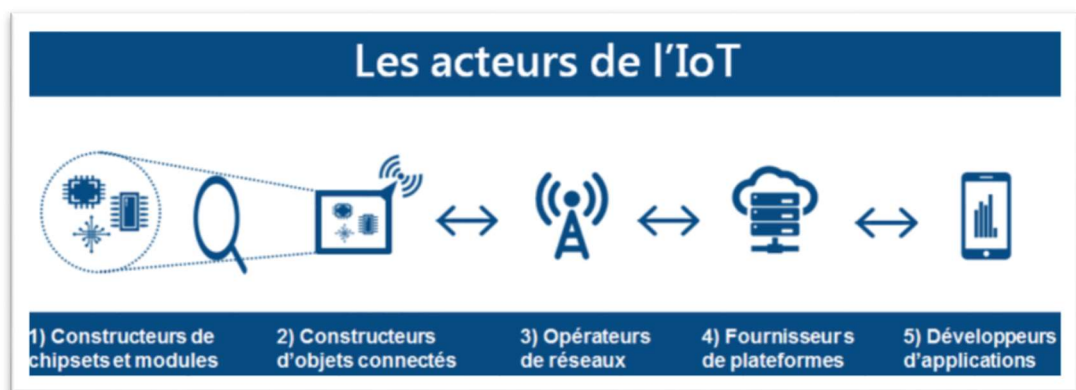


Figure I.8 : Les composants de l'écosystème IoT

I.4.2 Architecture d'un Réseau IoT :

L'architecture d'un réseau IOT est composée de plusieurs niveaux qui communiquent entre eux selon certaines réglementations pour relier les objets au monde virtuel, ces niveaux sont principalement des composants sur les quelles une architecture est encadré. Comme la montre la figure ci-dessous. [05]



Figure I.9 : L'architecture de l'IoT

1. Les Capteurs :

Sont des composants essentiels et intelligents dans un système IOT, les capteurs IoT sont généralement de petite taille, à faible coût et consomment moins d'énergie. Ils détectent les données du lieu de l'objet à mesurer et les transforment en données utiles.

2. Les actionneurs :

Sont des dispositifs qui peuvent modifier l'environnement en convertissant de l'énergie électrique en une énergie utile. On cite quelques exemples comme les éléments de chauffage ou de refroidissement, les haut-parleurs et les moteurs.

3. Passerelle :

Comme son nom indique est une passerelle vers internet pour tous les appareils auxquels nous voulons interagir. Autrement-dit jouent le rôle d'intermédiaire pour connecter l'objet à l'internet et envoyer ses données au Cloud. On prend l'exemple des routeurs domestiques, l'Arduino...etc. Ces Gateway fournissent ce qui est nécessaire en termes de connectivité et de sécurité. Ils garantissent que les périphériques et les capteurs sont correctement connectés.

Les passerelles IoT offrent des solutions locales pour le traitement et le stockage, et permettent de contrôler de manière autonome les appareils du terrain sur la base des données enregistrées par les capteurs. [06]

4. Nuage(Cloud) :

Le Cloud IoT est un réseau de haute performance qui connecte des serveurs pour optimiser les performances des processus de traitement de données traitées par plusieurs périphériques à la fois, aussi la gestion du trafic et fournir des résultats d'analyse de données précises, ce qui permet aux industries et les services d'accéder facilement à ces données à distance et prendre des décisions critiques en cas de besoin. [07]

Le Cloud associe de nombreux périphériques, passerelles, protocoles et un magasin de données pouvant être analysés efficacement. Ces systèmes sont utilisés par de nombreuses entreprises afin de disposer d'une analyse de données améliorée et efficace pouvant contribuer au développement des services et des produits.

❖ **L'architecture IOT en 4 étapes consiste à :** [05]

1^{er}Étape :

Il est important de commencer par inclure des capteurs dans les 4 étapes puisque s'est-il à détecter et collecter les données qui seront transférées sur le réseau. Encore les actionneurs permettent aux choses d'agir et intervenir dans la réalité par exemple allumée ou éteindre la lumière, ouvrir ou fermer une porte.

En fait, la phase de détection et d'activation couvre et ajuste tout ce qui est nécessaire dans le monde physique pour obtenir les informations nécessaires à une analyse approfondie.

2^{eme}Étape :

Système d'agrégation de données de capteurs et conversion de données analogique en numérique.

Les systèmes de la phase 2 sont installés à proximité de capteurs et d'actionneurs, L'importance vitale de cette étape est dans le traitement des quantités d'informations énorme collectées lors de l'étape précédente et le filtrage des données avant de les transférer dans le nuage afin de réduire le volume de données et d'avoir la taille optimale pour une analyse plus approfondie, cette étape consiste à numériser et à regrouper les données.

3^{eme} Étape :

Une fois les données sont numérisées et agrégées, elles sont préparées et prêtes à transférer au monde informatique, en particulier les systèmes informatiques périphériques effectuent ici une analyse et un prétraitement amélioré.

4^{eme}Étape :

Analyse, gestion et stockage des données :

Le processus de cette phase se passe dans un centre de données physique dans un système de nuage, précisément où des systèmes informatiques plus puissant peuvent analyser, gérer et stocker les données.

I.5 Les technologies de communication des objets connectés :

Il existe plusieurs technologies différentes qui ont chacune leurs particularités, et le choix se fait en fonction de certains nombres de critères tels que la portée, le débit et l'autonomie c'est-à-dire la consommation électrique des objets connectés.

I.5.1 Les technologies de courte portée :

I.5.1.1 La RFID : [08]

Est une technologie d'identification par radio fréquence, les objets équipés d'une puce électronique qui possèdent des étiquettes qui sont identifiés automatiquement par l'énergie radio fréquence. Le protocole RFID est un standard de communication à courte distance du contact à quelques centimètres et jusqu'à 15 mètres en haut fréquence (UHF).

I.5.1.2 Bluetooth :

Est le standard de transfert de données sans fil. Permettant les échanges à courtes distances utilisant des ondes radio UHF à courte longueur dans la bande ISM de 2.4-2.485GHz, quasiment tous les smartphones sont équipés de cette technologie. [09]

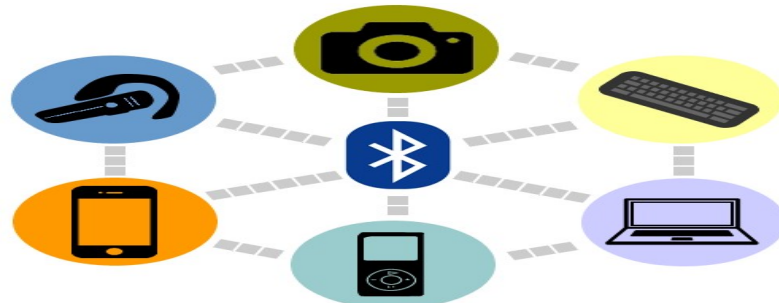


Figure I.10 : Les utilisations de la technologie Bluetooth

I.5.1.3 Zigbee :

C'est le protocole de communication radio développée pour les applications de la domotique, est un réseau d'une portée moyenne de 10-100 mètre et de faible débit de données jusqu'à 250kps .[09]

Le ZigBee Fonctionne à une bande de 915 MHz à 40 kbps aux États-Unis et à 868 MHz à 20 kbps en Europe. Il est conçu pour fonctionner en réseau maillé à topologie en étoile ou en arbrece qui offre à des nœuds de communiquer par le biais de liens vers d'autres nœuds situés dans la plage

faisant office de répéteurs, ZigBee peut prendre en charge jusqu'à 65 000 nœuds. L'un des grands avantages de ZigBee est sa flexibilité. [10]



Figure I.11 : La technologie ZigBee

I.5.2 Les technologies de moyenne portée :

I.5.2.1 Z-wave :

C'est le protocole de communication dédié à la domotique de produits tels que les contrôleurs de lampe, il permet de transmettre des données sur des distances allant de 30 mètres en intérieur à 100 mètres en plein air. Il utilise la bande ISM sans licence et fonctionne à 900 MHz aux États-Unis et au Canada. Optimiser pour la communication fiable et à faible temps de latence de petits paquets de données avec des débits pouvant atteindre 100 Kbit/ s.[09]

Il est facile à installer à la maison mais il n'est pas aussi universel que le Wifi car tous les appareils de la maison ne pourront pas forcément communiquer avec un objet connecté via cette technologie.



Figure I.12 : Le protocole de communication Zwave

I.5.2.2 Wi-Fi :

C'est l'ensemble de protocoles de communication sans fil, le wifi permet de transférer un nombre important de données de 600Mbps comme maximum mais le plus courant est de 150-200 Mbps, et des connexions à haut débit de 20 à 100 mètres. Cette technologie de réseau fonctionne à 2.4GHz - 05 GHz. Il s'agit d'un réseau local sans fil très énergivore ce qui impose un problème il ne peut pas être utilisé que pour les appareils branchés au secteur dans la maison par exemple. [09]



Figure I.13 : Les utilisations de wifi

I.5.2.3 Bluetooth LowEnergy :

Est une variante de la norme sans fil Bluetooth conçue pour une faible consommation d'énergie, qui réduit la consommation des débits de données en permettant aux connexions entrantes de contrôler le moment où le périphérique consommant de l'énergie. Il fonctionne sur l'une des fréquences de 204 GHz (ISM) d'une Portée de 50-150m (smart BLE) avec un Débit de données de 1Mbps (smart BLE).[09]

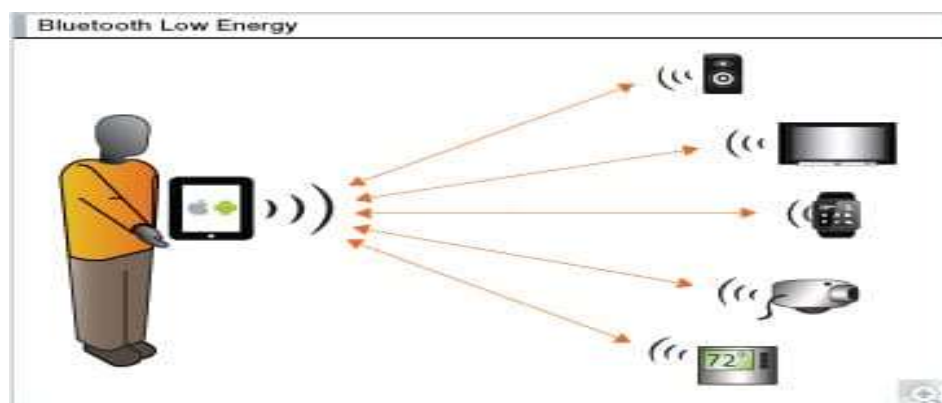


Figure I.14 : Les applications du protocole BLE

I.5.3 Les technologies de longue portée :

I.5.3.1 Réseau Cellulaires mobiles : [09]

Est la base des réseaux de téléphonie mobile, il convient également aux applications IOT qui fonctionnent sur de plus longues distances. Une portée de quelques kilomètres en ville à 30 km en zone rurale, Le cellulaire est capable d'envoyer et transporter un volume important de données avec une bonne couverture au niveau national et international. Il existe différentes normes de réseau cellulaires tels que le GSM, GPRS, EDGE, UMTS, LTE et chaque norme fonctionne sur sa propre fréquence par exemple 2100 MHz pour LTE. Pour la Portée aussi ça diffère 35 km pour GSM et 200Kmm pour HSPA.

Pour le Débit de données est varié de 35 à 170kbps pour GPRS, 120-384kbps pour EDGE, et entre 3 à 10 Mbps pour LTE.

I.5.3.2 Sigfox :

Le Sigfox est une technologie sans fil à bas débit à bande étroite d'une portée de 3- 10 Km en milieu urbain et peut atteindre les 30 aux 50 Km en milieu rural. Il permet de transmettre une faible quantité de données de 10 à 1000bps [09], comme ses appareils ne consomment que peu d'énergie. Sigfox fonctionne à 868 MHz en Europe et à 902 MHz aux Etats Unis. [10]

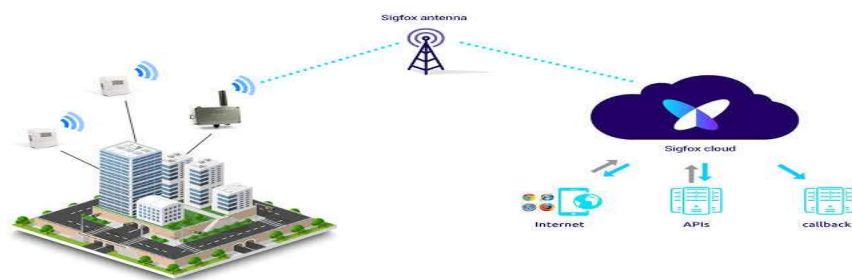


Figure I.15 : La technologie de communication Sigfox

I.5.3.3 LoRa :[09]

LoRa est une technologie de la couche physique qui module les signaux dans la bande ISM avec une bande plus large généralement de 125 kHz ou plus. Est Un Protocol de communication à basse consommation permet l'échange des données de faible taille et quantité en bas débit ce qui réduit la consommation énergétique des appareils. Et les transmettre à des distances de 2 à 5 Km

Chapitre I Étude des réseaux à base de l'internet des Objets

dans le milieu urbain et atteindre les 15 Km dans l'environnement suburbain. Débit de données 0.3-50Kbps.

LoRa est un réseau ouvert permet à n'importe quelle entreprise souhaite déployer son réseau et ses propres installations et de les exploiter et rendre les objets communicants il suffit de disposer d'une antenne reliée à l'internet wifi par exemple et une station de base émettant sur la bande de fréquence 868 MHz en Europe ou 915 MHz aux États-Unis. Par conséquence des performances excellentes et une portée importante.

I.5.3.4 LTE-M :

LTE-M est une nouvelle technologie simple à mettre en œuvre, Utilisé par des plus importants opérateurs mondiaux. Cette technologie utilise les mêmes protocoles que la 4G, il offre un débit plus soutenu allant jusqu'à 1Mbps voir 10Mbps avec une faible latence. [09]

Cette technologie permet de communiquer dans les endroits isolés d'accès difficile. Aujourd'hui via des bracelets connectés en LTE-M les personnes en difficulté pourront lancer une alerte et mises en relation immédiate avec un service d'assistance, Comme elle offre une autonomie importante pour les objets connectés.

I.5.3.5 NB_IOT :

NB_IOT est le protocole de communication sans fil, appliqué principalement au secteur de l'industrie utilise la bande de fréquence 200 KHz utilisée précédemment pour le GSM, il Est adapté pour des parcs tel que la télémétrie.

NB-IOT est l'une des technologies les plus efficaces en termes de capacité et de couverture qui est étendue avec une faible latence. Pour le débit de transmission de données est limitée à 150 Kbps.

I.5.3.6 Réseaux Propriétaires :

Certains groupes industriels veulent déployer des appareils communicantes à grande échelle pour cela préfèrent d'installer leur propre réseau de communication qui dit privé ou propriétaires.

I.6 La sécurité en internet des objets :

Les applications IoT affectant nos vies quotidiennes, dans les divers secteurs industriels, transports ou même la santé, il devient impératif de garantir la sécurité du système IoT.

Cependant, avec l'adoption continue des réseaux IP, les applications IoT sont devenues la cible d'attaques qui continueront de croître en ampleur et en sophistication. C'est pourquoi il est important de se concentrer sur l'analyse de la menace et l'évaluation des risques afin de mesurer les conséquences en cas d'incident de sécurité ou de violation puisque les tentatives d'intrusion peuvent survenir au niveau des différentes couches logicielles où les objets connectés sont régis par des systèmes d'exploitation et des logiciels PC ou des applications smartphones qui offrent de nombreux points d'accès aux hackers.

I.6.1 La protection des systèmes et des appareils IOT : [11]

Intégration de la sécurité lors de la conception :

Les développeurs IoT devraient inclure la sécurité au début de tout développement de périphérique grand public, professionnel ou industriel et Activer la sécurité par défaut est essentiel, tout en fournissant les systèmes d'exploitation les plus récents et en utilisant un matériel sécurisé.

Une mesure supplémentaire prend en charge exigée pour que les informations d'identité soient mises à jour par un utilisateur avant que le périphérique fonctionne.

➤ **PKI et certificats numériques**

Le PKI et les certificats numériques jouent un rôle essentiel dans le développement de dispositifs IoT, ils fournissent la confiance et le contrôle nécessaires à la distribution et à l'identification de clés de cryptage publiques, aussi à la sécurisation des échanges de données sur les réseaux et même la vérification de l'identité.

Sécurité de l'API :

La sécurité des API est importante pour protéger l'intégrité des données envoyées par les appareils IoT aux systèmes dorsaux et afin de garantir que seuls les appareils, et les applications autorisées communiquent avec ces API.

Gestion de l'identité :

Pour chaque périphérique il faut fournir un identifiant unique pour comprendre en quoi consiste le périphérique, son comportement avec d'autres périphériques avec lesquels il interagit et les mesures de sécurité appropriées à prendre pour ce périphérique.

Sécurité matérielle :

Le durcissement des points d'extrémité consiste à rendre les dispositifs inviolables. Ceci important lorsque les périphériques seront utilisés lorsqu'ils ne seront pas surveillés physiquement.

Sécurité Internet :

Protéger un réseau IoT consiste notamment à assurer la sécurité des ports, utilisation de logiciel anti-programme malveillant, de pare-feu et de système de détection d'intrusion ou de prévention d'intrusion, bloquer les adresses IP non autorisées et en veillant à ce que les systèmes soient corrigés et à jour.

Passerelles de sécurité :

Les passerelles de sécurité disposent de l'avantage de puissance de traitement et de capacités, ce qui leur donne la possibilité d'implémenter des fonctionnalités telles que des pare-feu pour empêcher les pirates d'accéder aux périphériques IoT qui connectent.

I.7 Comparaison entre M2M et l'IoT : [12]

Le Machine-to-Machine est défini comme un ensemble de technologies réseaux sans fil ou filaires rendant des systèmes ou deux machines communiquent et leur permettant de s'échanger automatiquement des informations sans interaction humaine. La communication est établie point à point entre la machine et un serveur, via différentes connexions utilisant des réseaux et plus particulièrement les réseaux sans fil publics comme GPRS. En fait, M2M et IoT sont presque des synonymes puisque les deux fournissent un accès à distance aux données entre les machines, à l'exception l'IoT étant fait généralement référence aux communications sans fil, alors que un réseau M2M peut faire référence à deux machines câblées ou sans fil comme il fonctionne sur un espace plus réduit.

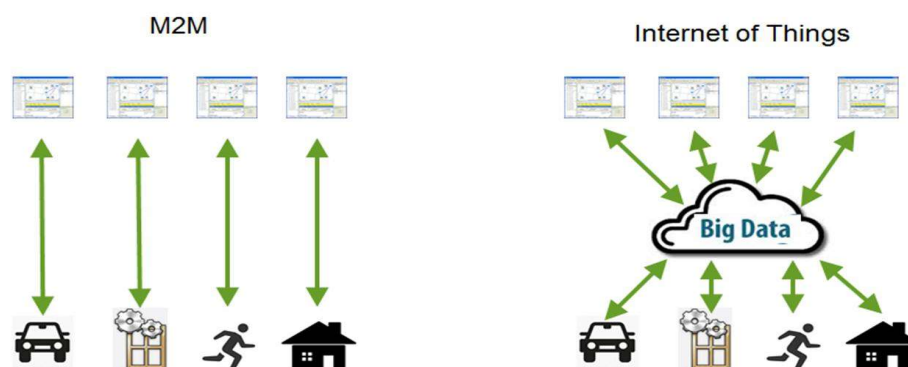


Figure I.16 : Comparaison entre M2M et IoT

I.8 Avantages et Inconvénients du réseau IoT :

L'IOT est une nouvelle infrastructure qui possède des avantages et des inconvénients, voici quelques avantages et désavantages.

I.8.1 Les Avantages : [13]

- L'IoT encourage la communication entre les périphériques ce qui rendre l'environnement connecté et pouvoir communiquer avec lui. De ce fait, les périphériques physiques peuvent rester connectés, ce qui permet une transparence totale avec moins d'inefficacité et une qualité supérieure.
- Améliore la sécurité et la qualité de vie, Connaitre la date des choses à la maison, la date de péremption des produits avant les consommer, et savoir ce qu'il faut obtenir de l'épicerie pendant que vous êtes dehors sans avoir à vérifier par vous-même.
- Permet de gagner un temps précieux. Au lieu de répéter les mêmes tâches tous les jours, cela permet aux utilisateurs de réaliser d'autres tâches créatives.
- L'aspect financier est le plus grand avantage de l'IoT puisque cette technologie pourrait remplacer les humains responsables de la surveillance et de la maintenance.

I.8.2 Les inconvénients : [14]

- Le problème majeur de l'IOT est la sécurité et la confidentialité de nos données personnelles, Avec toutes ces données IoT en cours de transmission, le risque de perte de confidentialité augmente, le fait qu'une ordonnance soit modifiée ou que les détails de votre compte soient piratés peut veut mettre en danger.
- La Complexité : l'IoT est un réseau diversifié et complexe. Toute défaillance ou tout bogue dans le logiciel ou le matériel aura des conséquences graves.
- Les fréquences exploitées sont également utilisées pour d'autres usages wifi, micro-ondes..., Et peuvent perturber le fonctionnement dans l'IoT.
- le développement et l'utilisation de l'IOT dans tous les domaines de la vie, nécessite une augmentation de nombre des ondes électromagnétiques et ceci est nocif pour notre santé.

Conclusion

Dans ce chapitre, nous avons commencé par une introduction sur l'internet des objets dans le but de comprendre certaines généralités sur le réseau "IOT" et avoir une définition globale sur lui, comme nous avons présenté différents concepts et les composants d'un système IoT dans le cadre de son architecture.

En fait, une meilleure vision sur le réseau IoT nous permet de visionner leurs domaines d'utilisation, puis les différentes étapes d'évolution de ce réseau et enfin les avantages et les inconvénients que peut posséder un tel réseau.

Tout cela nous donne un aperçu général sur les réseaux IOT, c'est qui nous permet d'avancer au chapitre suivant pour présenter tout ce qui nécessaire comme Matériel et logiciel afin de réaliser le projet final.

Chapitre II

Présentation de matériels et de
logiciel

Introduction

Ce chapitre demeure à mettre l'accent sur la description des accessoires utilisés dans notre montage de la maquette, en termes de composants physiques (tels que les capteurs, moteur pas à pas, servomoteur) et en terme de moyen de communication module Wifi dans une première partie, puis une description sur le logiciel de programmation dans la seconde partie.

II.1 La description de matériel :

II.1.1 La carte Arduino :

La carte Arduino est un circuit imprimé programmable comportant tous les composants électroniques pour faire fonctionner un microcontrôleur associé à une interface USB lui permettant de communiquer avec un ordinateur. Un microcontrôleur est l'équivalent d'un petit ordinateur, il contient un microprocesseur associé à différentes mémoires et des interfaces lui permettant de communiquer avec l'extérieur.[15]

La carte associée des entrées sorties permet à l'utilisateur de brancher différents types d'éléments externes, figure (II.1).

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

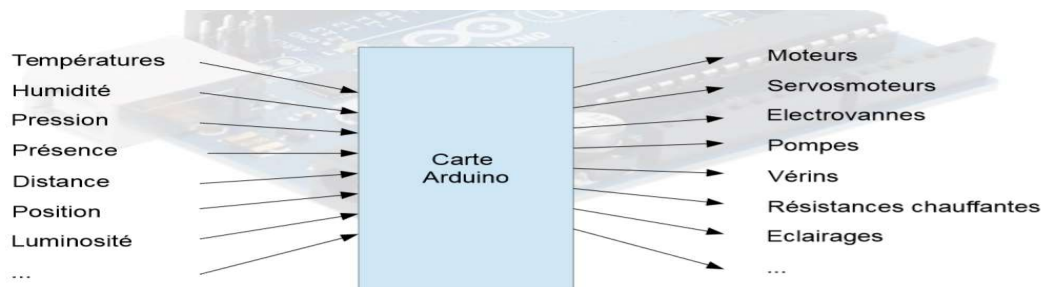


Figure II.1 : Les entrées et les sorties de la carte Arduino

Il existe Plusieurs modèles de cartes Arduino (Uno, Micro, NANO, DUE, Mega2560...), plus ou moins puissantes. Dans notre cas, nous allons utiliser le modèle : Arduino Mega2560.

II.1.2 La carte Arduino Mega2560 :^[16]

La carte Arduino Mega2560 est une carte microcontrôleur basée sur la technologie ATmega2560, Comme son nom le suggère, Méga2560 est une carte plus performante que quelques d'autres types, elle offre la base Uno, mais avec des fonctionnalités supplémentaires. Elle dispose de 54 broches numériques entrées/ sorties, dont 14 peuvent être utilisées comme sorties PWM, elle dispose 16 entrées analogiques, ainsi que plusieurs liaisons séries. Elle peut être alimentée via la connexion USB type B ou avec une source d'alimentation externe.

II.1.2.1 Les caractéristiques techniques de la carte Méga :

- Microcontrôleur ATmega2560 ;
- Fonctionnement Tension 5V ;
- Tension d'entrée (recommandé) 7-12V ;
- Tension d'entrée (limites) 6-20V ;
- Courant pour Pin 50 mA ;
- Vitesse d'horloge 16 MHz ;
- Dimensions : 107*53*15mm.

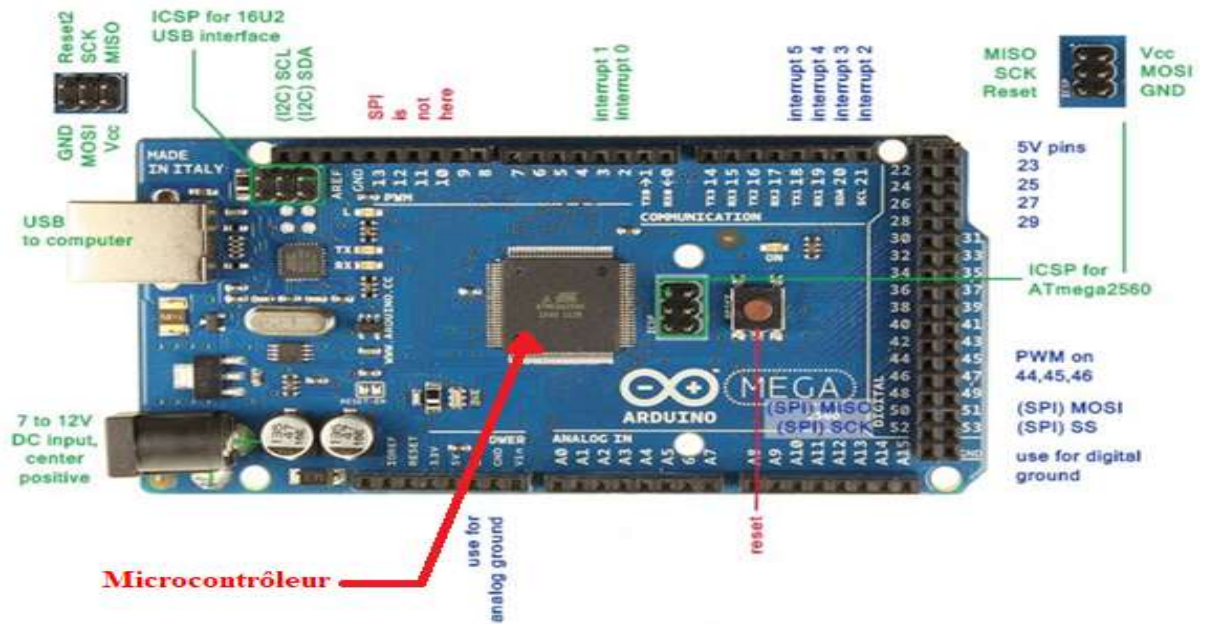


Figure II.2 : Carte Arduino Mega2560

II.1.3 Accessoires de communications avec Arduino Méga2560 :

La carte ArduinoMéga2560est dotée de plusieurs ports de communications, on peut éclaircir actuellement le type utilisé dans notre projet est le module Wifi ESP 8266 :

II.1.3.1 Module Wi-Fi ESP8266:[17]

L'ESP8266 est un composant électronique qui permet de monter une connexion Wifi sur le box de notre réseau local, par conséquent d'accéder à tous les équipements connectés sur le LAN Le module ESP8266 est un émetteur-récepteur sans fil autonome peu coûteux qui peut être utilisé pour les développements IoT de bout en bout.

De nombreux fabricants produisent différents modules basés sur cette puce. Ainsi, le module est livré avec différentes options de disponibilité des broches telles que :

- L'ESP-01 est livré avec 8 broches (2 broches GPIO) antenne trace PCB.
- L'ESP-02 est livré avec 8 broches (3 broches GPIO) connecteur d'antenne U-FL.
- L'ESP-03 est livré avec 14 broches (7 broches GPIO) Antenne en céramique.
- L'ESP-04 est livré avec 14 broches (7 broches GPIO) pas d'antenne.



Figure II.3: Types de module ESP8266

II.1.3.2 Les caractéristiques du module ESP8266 :[18]

- Wi-Fi 2,4 GHz (802.11 b / g / n, prenant en charge WPA / WPA2),
- Microcontrôleur 32bits intégré basse consommation.
- CAN 10-bits intégré.
- Dotée d'une Pile TCP/TP.
- Consommation en veille <1.0 mW.
- entrée / sortie polyvalente (16 GPIO),
- Protocole de communication série du circuit inter-intégré (I²C).
- Protocole de communication série SPI.
- UART (sur des broches dédiées, plus un UART de transmission uniquement peut être activé sur GPIO2).
- modulation de largeur d'impulsion (PWM).

II.1.3.3 Description de module ESP8266-01 :[17]

ESP8266-01 est un petit module à base de microcontrôleur ESP8266 qui embarque une puce Wifi de petit taille, Le module dispose d'une mémoire flash plus petite (de 512Ko à 1MB), c'est la version la plus limitée de l'ESP8266, il dispose seulement de 2 GPIO. Comme est montré dans la figure (II.4).

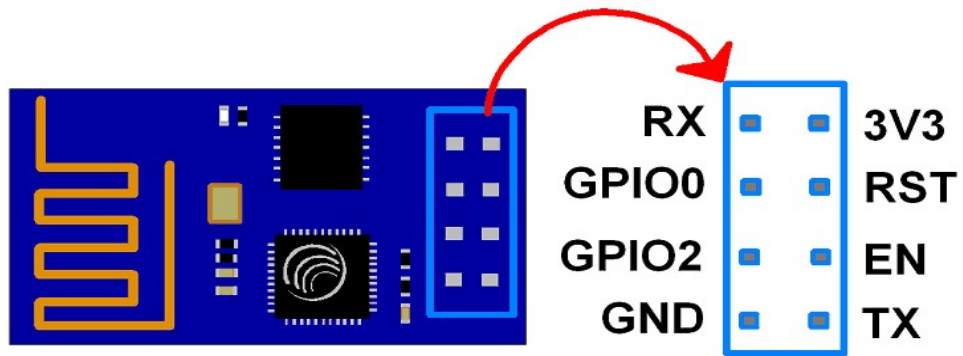


Figure II.4 : Les broches de connexion d'ESP8266-01

- **3V3:** Broche alimentation;
- **GND:** Broche de masse;
- **RST:** Broche de réinitialisation active à l'état bas;
- **EN:** Broche active De l'état haut;
- **TX:** Broche de transmission série de l'UART ;
- **RX:** Broche de réception en série d'UART ;
- **GPIO0 & GPIO2:** Broches d'E/S.

II.1.4 les Capteurs : [19]

Un capteur est un dispositif sophistiqué utilisé pour détecter les signaux avec ses diverses natures électriques, ou optiques et transforme la grandeur physique mesurée qui peut être une température, Une pression, l'humidité ou une vitesse en une grandeur pouvant être mesurée électriquement (intensité électrique) (figure II.5).

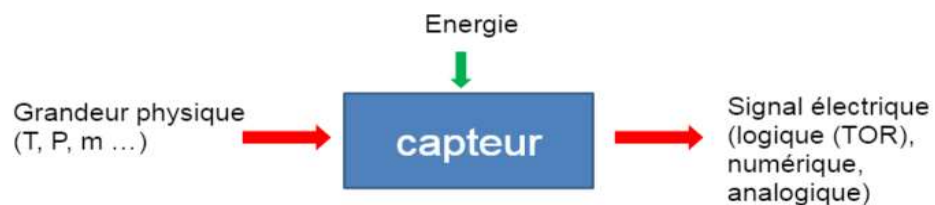


Figure II.5: La chaîne d'action d'un capteur

II.1.4.1 Les différents types de capteurs :[20]

Il existe une large gamme de capteurs que nous pouvons exploiter pour mesurer beaucoup de propriétés physiques qui nous entourent. Les capteurs peuvent être regroupés selon plusieurs critères, on distingue :

1. Capteurs actifs :

Sont des capteurs qui ont leur propre source d'énergie et qui ne nécessitent pas de source d'alimentation externe pour leur fonctionnement. Ils génèrent de l'énergie en eux-mêmes pour fonctionner.

2. Capteur passifs :

Les capteurs passifs sont des capteurs qui nécessitent une source d'alimentation externe pour leur fonctionnement. La plupart des capteurs résistifs, inductifs et capacitifs sont passifs tout comme, les transformateurs, les antennes, les potentiomètres.

II.1.5 les moteurs :

Sont des systèmes qui a pour but de produire un mouvement précis en réponse à une commande externe, C'est un actionneur produisant une action, dans notre cas nous avons exploité le moteur Pas à Pas et le servomoteur.

II.1.6 Relais :[21]

Un relais est un commutateur électrique qui permet de commander plusieurs organes simultanément grâce à ses multiples contacts synchronisés. Utilisant généralement une tension et un courant bien que supérieur à ce que l'arduino pourrait accepter.

Globalement, il permet une transition entre deux sources différentes en isolant ces dernières. Le relais est un mécanisme simple d'interrupteur on/off ; il se ferme lorsque l'entrée à 5v et s'ouvre lorsque elle est à 0v.



Figure II.6 : Le relais

Ce module se compose d'une partie commande à droite qui est composée d'une broche d'alimentation de 5 v pour le relais, une masse et une broche de commande pour faire basculer l'état. De l'autre côté, il fournit 3 connections COM, NC, NO. Celles qui seront connectées à notre dispositif.

II.1.8 buzzer :

Le buzzer est un dispositif de signalisation audio qui peut être mécanique, électromécanique ou piézoélectrique largement utilisé dans les ordinateurs, les alarmes, jouets électroniques. On distingue deux types le buzzer passif et actif.



Figure II.7:Buzzer passif

II.1.9 Les LEDs :

La diode électroluminescente est un composant optoélectronique, capable d'émettre de la lumière lorsqu'il est parcouru par un courant électrique.



Figure II.8: LEDs / DEL

II.2 La description de logiciel :

Après le choix de la carte il va falloir installer l'outil nécessaire à sa programmation, il existe plus d'un logiciel de programmation de l'arduino parmi ces logiciels :

- Arduino IDE (logiciel arduino officiel) ;
- Arduino Web Editor ;
- Eclipse Arduino IDE.

Pour notre projet, nous avons choisi arduino IED. Le logiciel Arduino est un environnement de développement open source et gratuit, téléchargeable sur site officiel d'Arduino(www.arduino.cc). Il permet de :

- Editer un programme : des croquis (sketch en anglais) ;
- Compiler ce programme dans le langage machine de l'Arduino ;
- Téléverser le programme dans la mémoire de l'Arduino ;
- Communiquer avec la carte Arduino grâce au terminal.

II.2.1 description de différents menus du logiciel :

Après avoir installé l'IDE arduino, nous ouvrons le logiciel une fenêtre comme ci-dessous ;

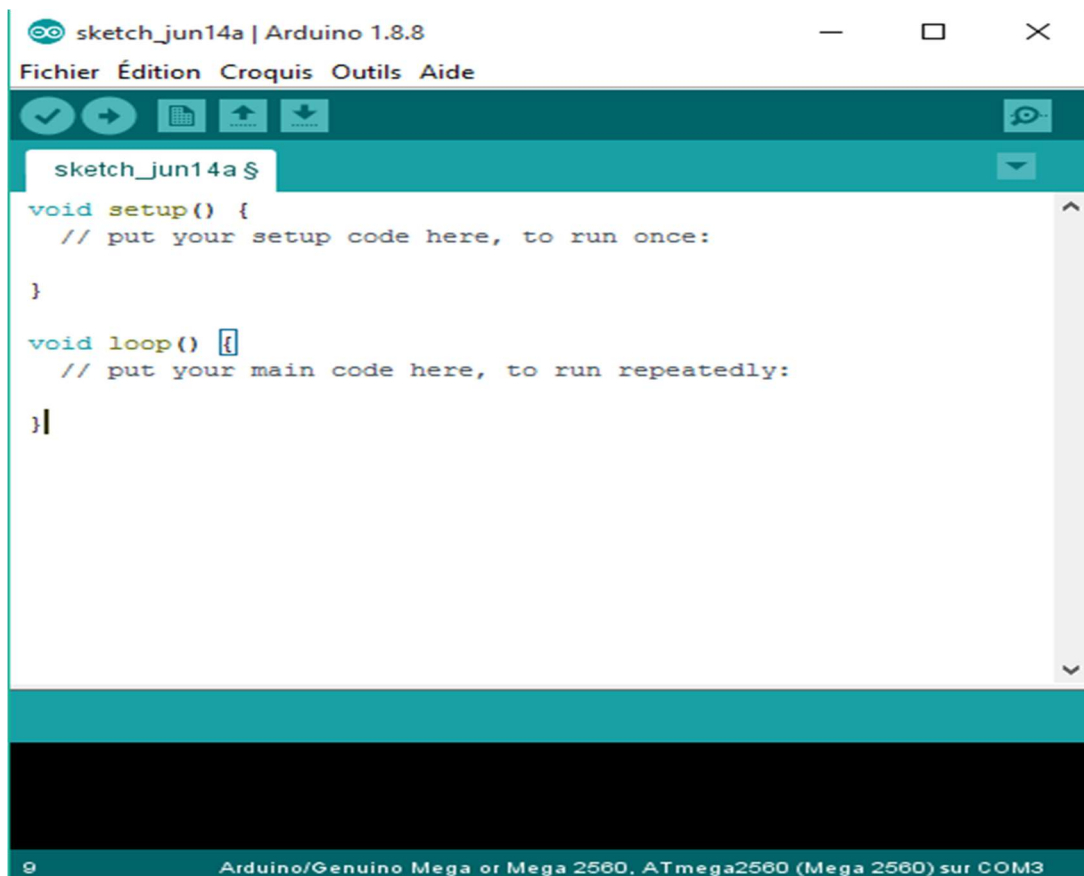


Figure II.9:L'interface de l'IDE Arduino

II.2.1.1 L'interface de commande de logiciel :[22]

l'image précédente se découpe en plusieurs parties :

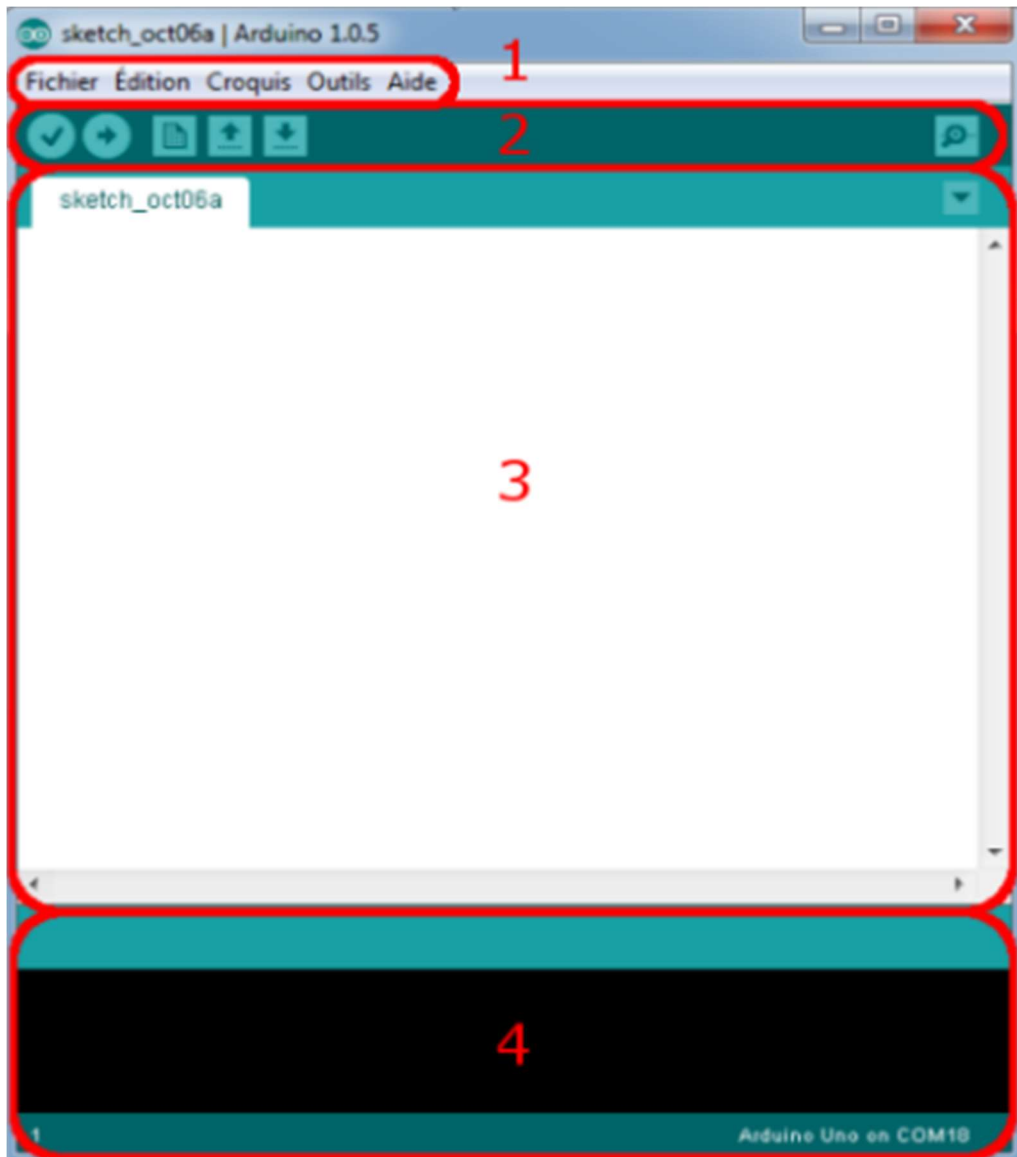


Figure II.10: L'interface de l'IDE Arduino en détail

- Le cadre numéro 1 : ce sont les options de configuration du logiciel ;
- le cadre numéro 2 :il contient les boutons qui vont nous servir lors de la programmation ;
- Le cadre numéro 3 : ce bloc va contenir le programme que nous allons créer ;
- Le cadre numéro 4 : C'est le débogueur, il va nous aider à corriger les fautes dans notre programme.

II.2.1.2 Approche et utilisation du logiciel :

Nous Allons décrire l'utilisation du logiciel, plus particulièrement la barre des menus entourée en rouge portant le numéro 1 dans la figure (II.10) :

Le menu Fichier :

Principalement ce menu Il dispose d'un certain nombre de choses qui vont être très utiles.



Figure II.11: Le menu Fichier

- **Carnet de croquis :** Ce menu regroupe les fichiers fait et s'ils sont enregistrés dans le dossier par défaut du logiciel.
- **Exemples :** toute une liste se déroule pour afficher les noms d'exemples de programmes existants qui aident à la création des programmes ou tester de nouveaux composants.
- **Préférences :** permet de régler quelques paramètres du logiciel.

Le menu Croquis :

Cemenu est utile lorsqu'on veut ajouter des bibliothèques qui ne sont pas disponibles dans le logiciel sous un fichier zip.

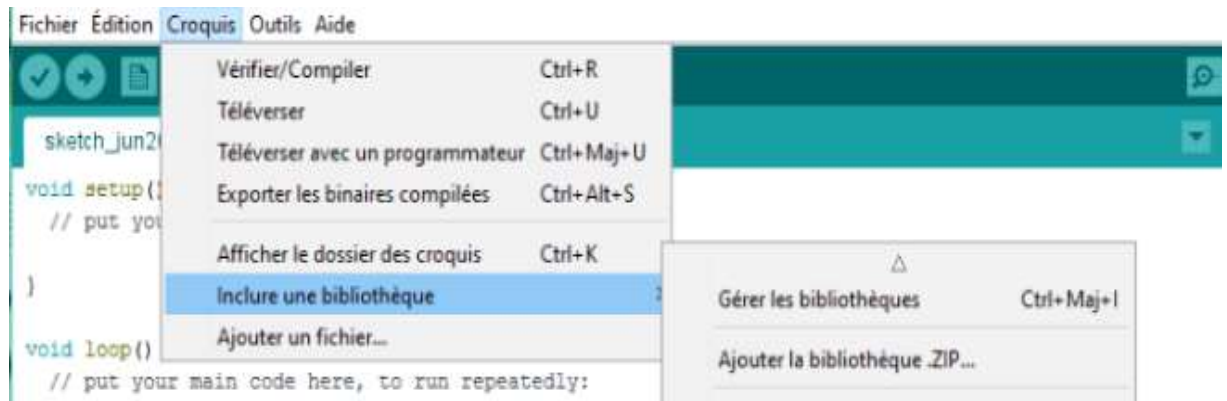


Figure II.12: Le menu Croquis

- Téléverser : Permet d'envoyer le programme sur la carte Arduino.
- Téléverser avec un programmeur : Identique que ci-dessus, mais avec l'utilisation d'un programmeur.



Le menu outils :

Dans le menu des outils, nous pouvons sélectionner le type de carte Arduino utilisée et même la sélection de port.

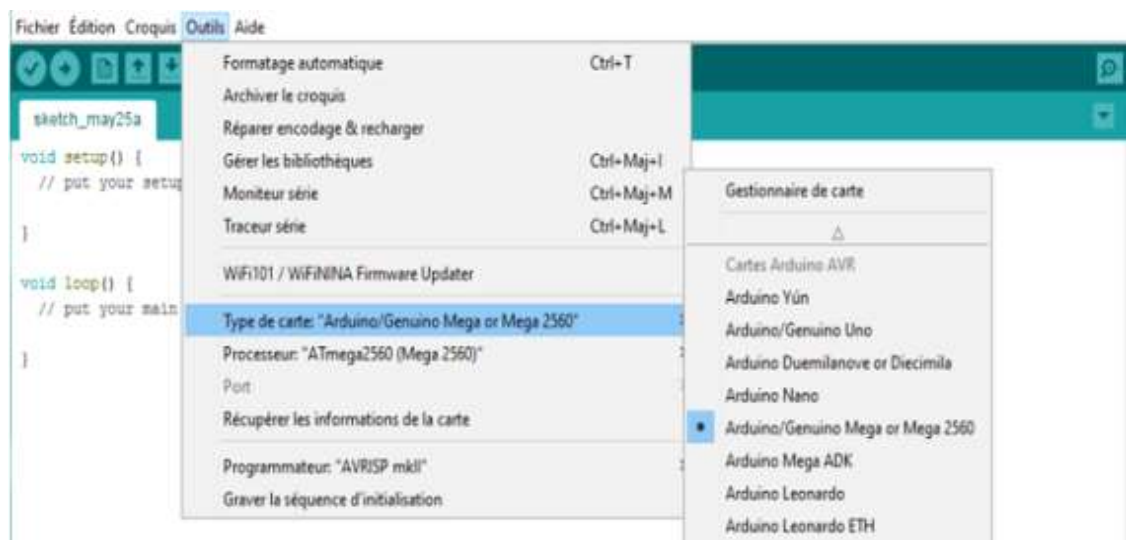


Figure II.13: Menu outils de sélection de la carte cible

Les boutons :[22]

Nous allons voir à présent, à quoi servir les boutons encadrés en rouge et numérotés par le chiffre 2 dans la figure (II.10).

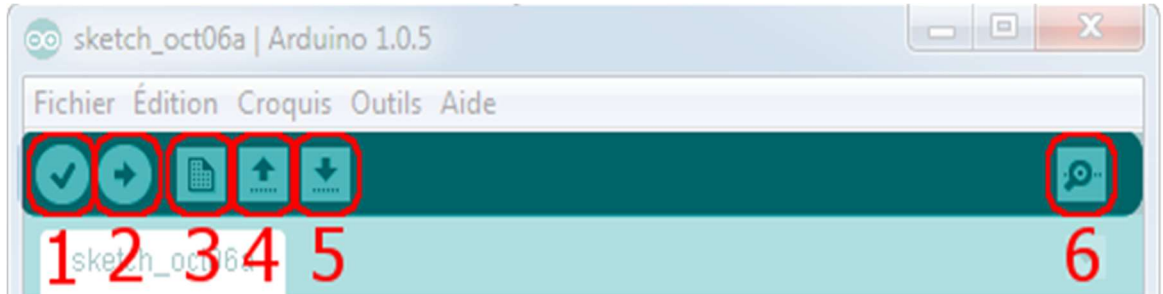


Figure II.14 : La barre d’outils

- Bouton 1 : permet de vérifier le programme, il actionne un module qui cherche les erreurs dans le programme.
- Bouton 2: Charge (télé verser) le programme dans la carte Arduino.
- Bouton 3 : Crée un nouveau fichier.
- Bouton 4 : Ouvre un fichier.
- Bouton 5 : Enregistre le fichier.
- Bouton 6 : Ouvre le moniteur série.

Le moniteur série

Il suffit de cliquer sur le bouton dans la barre de menu entourée en rouge pour démarrer l’outil



Figure II.15 :Sélection moniteur série

Une nouvelle fenêtre s'ouvre c'est le moniteur série (figure II.17).



Figure II.16 : la voie série

Dans cette fenêtre, nous allons pouvoir envoyer des messages sur la voie série de notre ordinateur, recevoir les messages que l'Arduino nous envoie, et régler deux à trois paramètres tels que la vitesse de communication avec l'Arduino et l'auto scroll qui fait défiler le texte automatiquement.

II.2.2 Installation des cartes et des bibliothèques de module ESP8266 sur l'IDE Arduino :[23]

Après lancement de l'IDE Arduino depuis le menu fichier puis références, dans le champ de saisie indiquée en rouge dans la figure (II.17), coller l'adresse suivante puis valider par [ok]. (https://arduino.esp8266.com/stable/package_esp8266com_index.json).

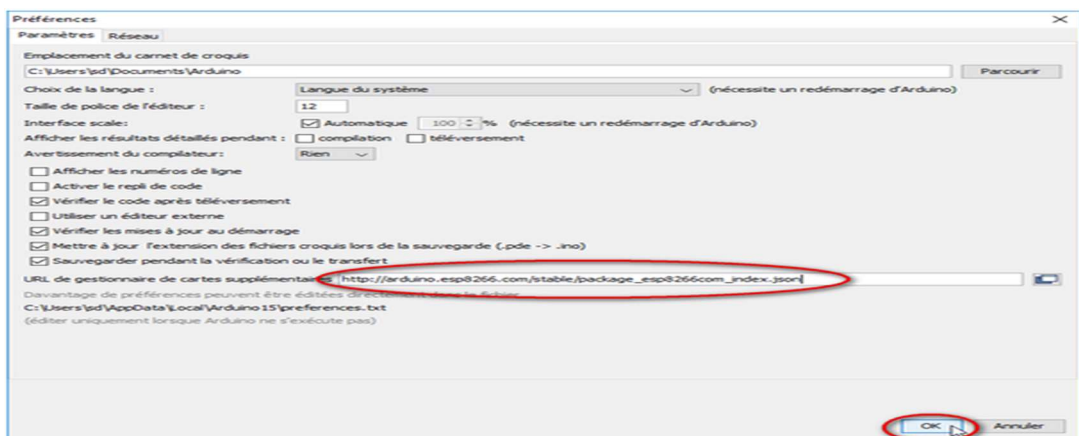


Figure II.17 : L'adresse de saisie correspond aux bibliothèques d'Esp8266

Dans le menu **Outils** puis **Type de Cartes** et enfin **Gestionnaire de carte** comme est montré sur la figure(II.18).

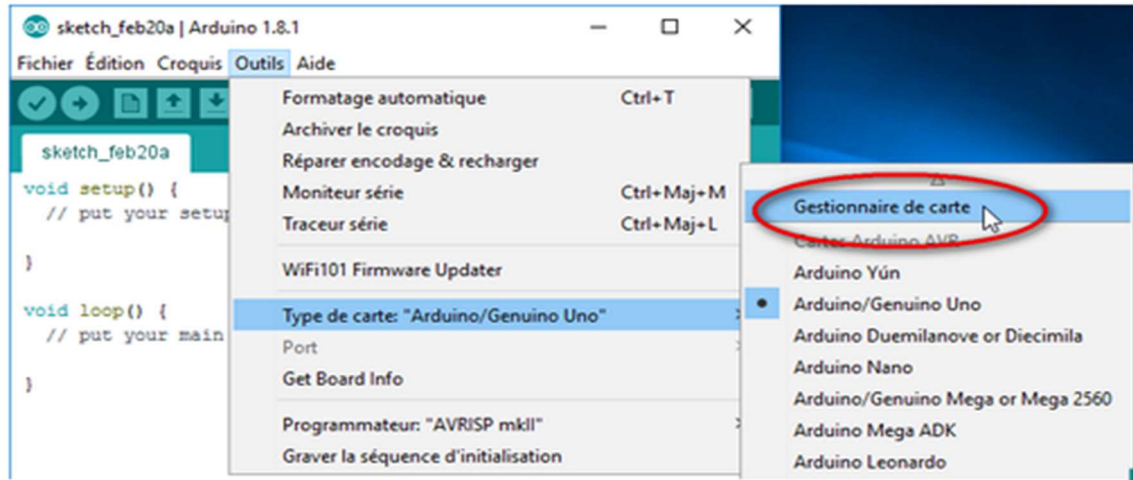


Figure II.18 :Sélection de gestionnaire de carte dans menu outils

Dans le champ de recherche, saisir esp8266 pour trouver les nouvelles cartes à installer

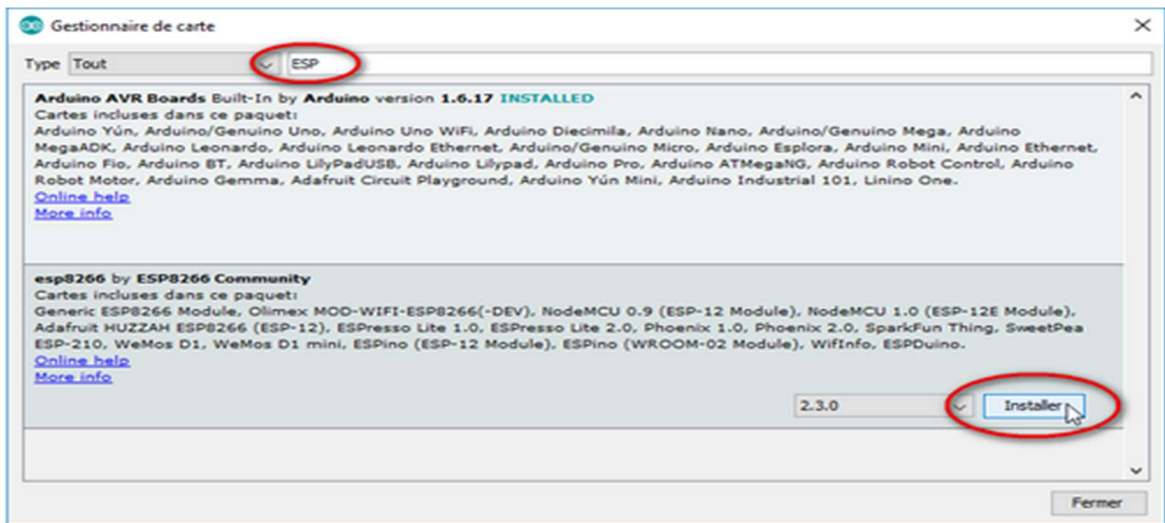


Figure II.19:L'ajoute des bibliothèques ESP8266

Dans le menu Outils, dans le sous menu **Type de carte**, une nouvelle section appelée **ESP8266 Modules** est ajoutée.

II.2.3 La syntaxe du langage de programmation :[22]

La syntaxe d'un langage de programmation est l'ensemble des règles d'écriture liées au langage de programmation utilisé qui est le C++ dans l'IDE.

Avec Arduino, Dans le code utilisé, on trouve deux fonctions essentielles qui sont en fait des portions de code :

- La fonction **setup** : la fonction d'initialisation est appelée une seule fois lorsque le programme commence.

```
// fonction d'initialisation de la carte
void setup() {
  // contenu de l'initialisation
  // on écrit le code à l'intérieur
}
```

Figure II.20 : La fonction setup

- La fonction **voidloop** : dans la fonction loop où l'on va écrire le contenu du programme. Cette fonction est exécutée une fois, puis lorsque son exécution est terminée, on la ré-exécute et encore.

```
// fonction principale, elle se répète (s'exécute) à l'infini
void loop() {
  // contenu de votre programme
}
```

Figure II.21 : La fonction loop

Une fois le programme est rédigé et envoyé dans la carte, on clique sur le bouton 2 dans la figure (II.15). Le logiciel Arduino va vérifier si le programme ne contient pas d'erreur puis le transférer dans la carte.

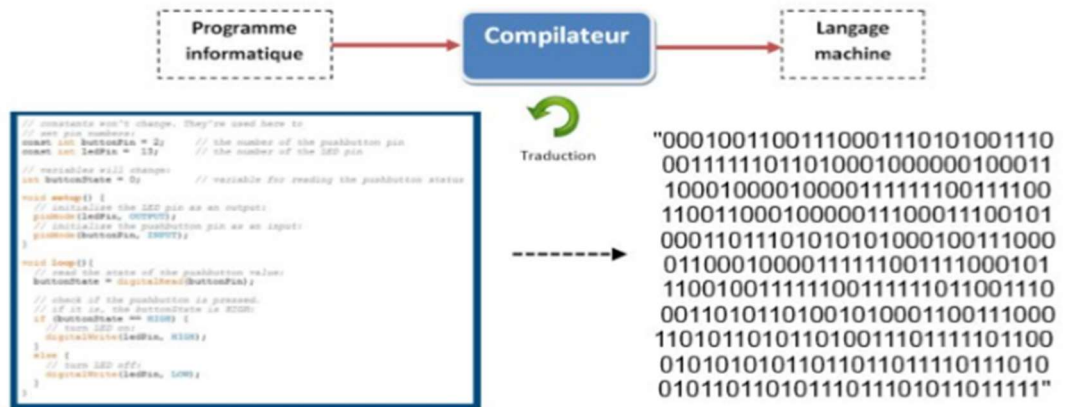


Figure II. 22 : La compilation et transformation du code sur l'arduino

Conclusion

Dans ce chapitre, nous avons présenté :

La carte Arduino Méga 2560 et le module wifi ESP8266 ainsi tous les composants électroniques et outils logiciels utilisés dans notre projet, on finit par une syntaxe sur langage de programmation.

Ce chapitre nous a permis de donner tous les concepts nécessaires qui vont être utilisés dans les chapitres suivants.

Chapitre III

Conception des capteurs et des
actionneurs sur Arduino

Introduction

Les capteurs sont les yeux et les oreilles de l'internet des objets, il n'y a pas d'IoT sans capteurs, ils sont des facteurs clé de succès de l'IoT.

Dans ce chapitre, nous allons présenter les différents capteurs et actionneurs utilisés dans la réalisation de notre projet. Ensuite, nous allons les programmer sur la carte Arduino.

III.1 Capteur de distance à Ultrasons HC-SR04 :

Le capteur HC-SR04 utilise les ultrasons pour déterminer la distance d'un objet, peu importe l'intensité de la lumière, la température ou le type de matière. Il mesure la distance des objets situés de 2cm à 400cm du capteur avec une précision de 3mm, son champ de vision est de 90°selon l'environnement.[24]

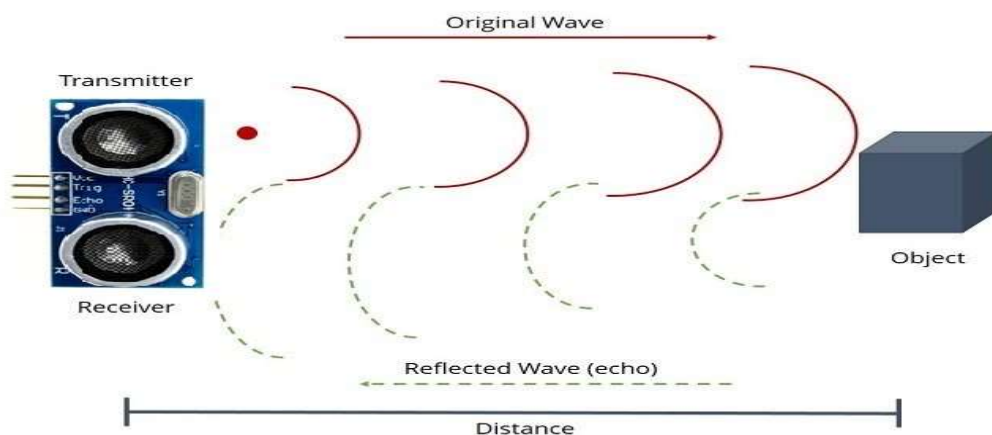


Figure III.1: Le capteur Ultrasons HC-SR04

III.1.1 Fonctionnement de Capteur Ultrasons HC-SR04 :[25]

Le déclenchement d'une mesure se fait par l'envoi d'une impulsion à l'état haut (5v) pendant 10 μ s sur l'entrée Trig. Le capteur émet une série de 8 impulsions ultrasoniques à 40 kHz, Puis il attend le signal réfléchi, lorsque celui-ci est détecté, une impulsion de durée proportionnelle à la distance mesurée est envoyée sur la sortie "Echo". Le calcul de la distance peut se faire grâce au temps mesuré entre le signal émis et le signal reçu : $D = (T * V_s) / 2$

D = distance entre l'objet et le capteur.

T= temps mesuré entre le signal émis et le signal reçu.

Vs= vitesse du son dans l'air : 340 m/s.

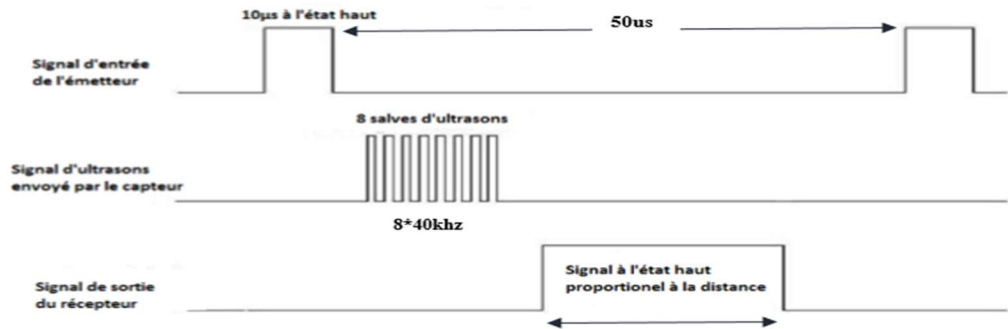


Figure III.2:Diagramme temporelle

Paramètres électriques :

Tension d'alimentation	5V DC
Courant d'alimentation	15mA
Fréquence de travail	40khz
Distance maximale de détection	4m
Distance minimale de détection	2cm
Angle de détection	15 degrés
Signal d'entrée de l'émetteur	Impulsion à l'état haut de 10µs
Signal de sortie du récepteur	Signal numérique à l'état haut et la distance proportionnellement
Dimension	45*20*15mm

Tableau III.1 : Paramètres électrique de Capteur à Ultrasons HC-SR04

III.1.2 Schéma du montage sur Arduino :

Broches de connexion :

- Vcc = Alimentation +5V DC ;
- Trig = Entrée de déclenchement de la mesure (Trigger input) ;
- Echo = Sortie de mesure donnée en écho (Echo output) ;
- GND = Masse de l'alimentation (0v).

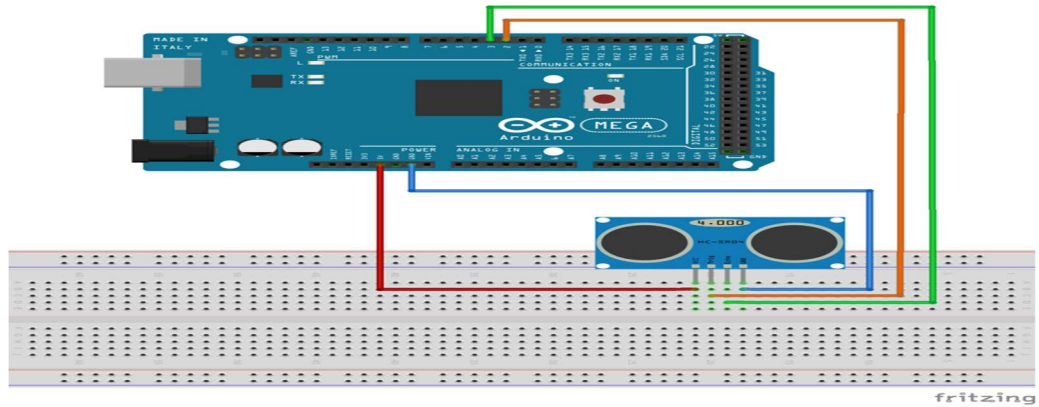


Figure III.3: Montage de capteur Ultrasons HC-SR04 sur la carte Arduino Méga 2560

III.1.3 Programmation du capteur Ultrasons HC-SR04 :

```
Fichier Édition Croquis Outils Aide
sketchhhh
/* Constantes pour les broches */
const byte TRIGGER_PIN = 2; // Broche TRIGGER
const byte ECHO_PIN = 3; // Broche ECHO
/* Constantes pour le timeout */
const unsigned long MEASURE_TIMEOUT = 25000UL; // 25ms = ~8m à 340m/s
/* Vitesse du son dans l'air en mm/us */
const float SOUND_SPEED = 340.0 / 1000;
void setup() {
  /* Initialise le port série */
  Serial.begin(9600);
  /* Initialise les broches */
  pinMode(TRIGGER_PIN, OUTPUT);
  digitalWrite(TRIGGER_PIN, LOW); // La broche TRIGGER doit être à LOW au repos
  pinMode(ECHO_PIN, INPUT);
  /* Initialise le terminal*/
  Serial.println("Demarrage des mesures");
}
void loop() {
  /* 1. Lance une mesure de distance en envoyant une impulsion HIGH de 10µs sur la broche TRIGGER */
  digitalWrite(TRIGGER_PIN, HIGH);
  delayMicroseconds(10);
  digitalWrite(TRIGGER_PIN, LOW);
  /* 2. Mesure le temps entre l'envoi de l'impulsion ultrasonique et son écho (si il existe) */
  long measure = pulseIn(ECHO_PIN, HIGH, MEASURE_TIMEOUT);
  /* 3. Calcul la distance à partir du temps mesuré */
  float distance_mm = measure / 2.0 * SOUND_SPEED;
  /* Affiche les résultats en mm, cm et m */
  Serial.print(F("Distance: "));
  Serial.print(distance mm);

  Serial.print(distance_mm / 10.0, 2);
  Serial.print(F("cm, "));
  Serial.print(distance_mm / 1000.0, 2);
  Serial.println(F("m"));
  /* Délai d'attente pour éviter d'afficher trop de résultats à la seconde */
  delay(500)
}
```

Figure III.4: Le code de programmation du capteur ultrasons

III.1.4 Résultats :

Les résultats seront affichés dans le moniteur série Arduino, la distance mesurée par le capteur ultrasons HC-SR04 en divers unités millimètre, centimètre et mètre comme est montrée sur la figure ci-dessous.

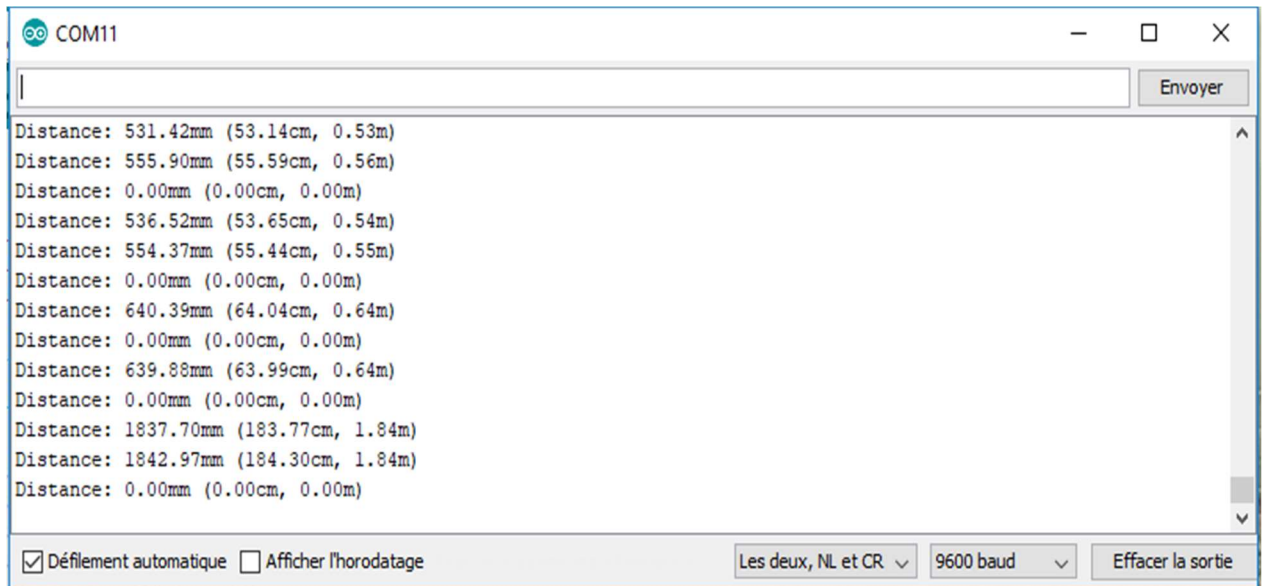


Figure III.5: Mesure de distance par capteur Ultrasons Hc-SR04

III.2 Capteur infrarouge Passif PIR :

Le terme passif signifie que le capteur n'utilise aucune énergie pour la détection, il fonctionne en détectant l'énergie dégagée par les autres objets.

Le PIR est un capteur capable de détecter les mouvements. Il capte les rayonnements infrarouges dans son champ de vision et en déduit une présence ou un mouvement. Ce capteur peut servir dans de nombreuses applications notamment dans un système d'alarme.[26]



Figure III.6: Capteur infrarouge Passif (PIR)

III.2.1 Les spécifications techniques de capteur infrarouge passif :

- Entrée : 4.5v à 20v
- Sortie : high 3.3v / low 0v

- Angle : $< 100^\circ$.
- Dimension : 32 mm*24 mm
- Délai : de 5 à 200 secondes (ajustable).
- Porté : de 3 à 7 m.
- Au repos : 50 μ A.

III.2.2 Fonctionnement :[27]

Le module se compose de capteur pyroélectrique doté de deux cellules au minimum, qui sont sensibles à la chaleur et qui mesure les radiations IR émises par les objets se trouvant dans son champ de vision. Cela signifie que lorsqu'un corps humain ou animal va entrer dans la portée du capteur, il détectera un mouvement.

Il contient également une enveloppe appelée lentille de Fresnel, qui focalise les signaux infrarouges sur le capteur pyroélectrique.

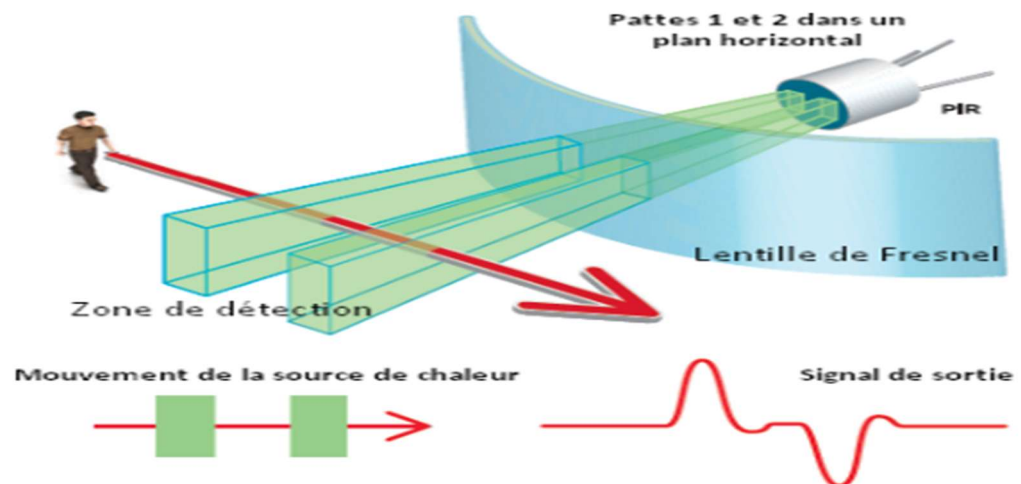


Figure III.7: Fonctionnement de capteur infrarouge passif

Le capteur possède deux modes de fonctionnement :

- Un mode de détection unique qui permet de détecter un seul mouvement à la fois.
- Un mode de détection multiple qui permet de détecter plusieurs mouvements et de déclencher le compteur du temps à l'état haut à partir du dernier mouvement détecté.

III.2.3 schéma de montage sur Arduino :

Le module possède seulement trois broches de connexion :

- VCC : pour alimenter le module

Chapitre III Programmation des capteurs et des actionneurs sur Arduino

- OUT : une broche de sortie qui donne un niveau logique élevé si un objet est détecté.
- GND : la masse d'alimentation.

On branche donc :

- Le VCC du PIR sur 5v de l'Arduino.
- Le GND de PIR sur le GND de l'Arduino.
- La dernière branche sur le Pin 2de l'Arduino.
- On ajoute LED de contrôle entre le Pin 13et un GND de l'Arduino.

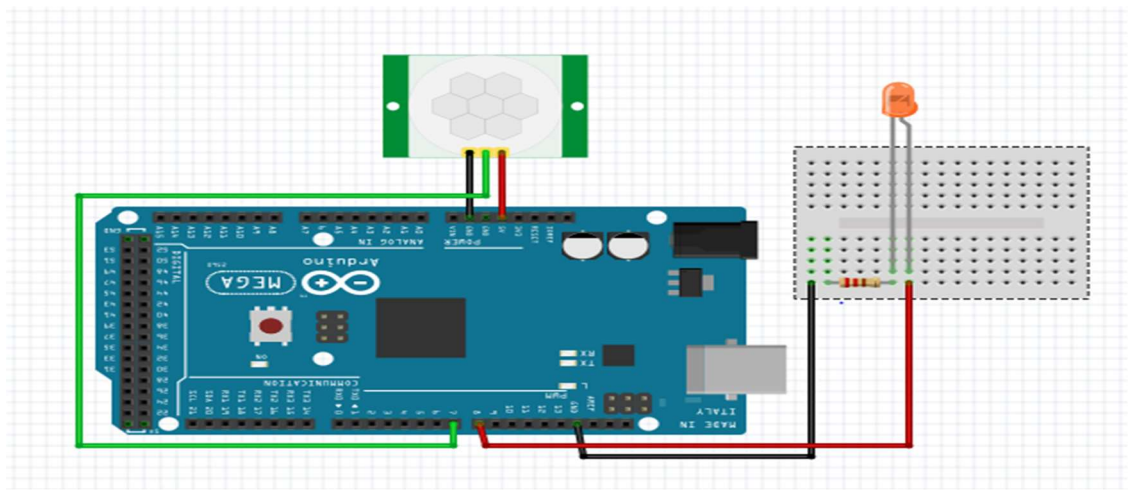


Figure III.8:Montage du capteur Infrarouge Passif PIR sur l'arduino Méga2560

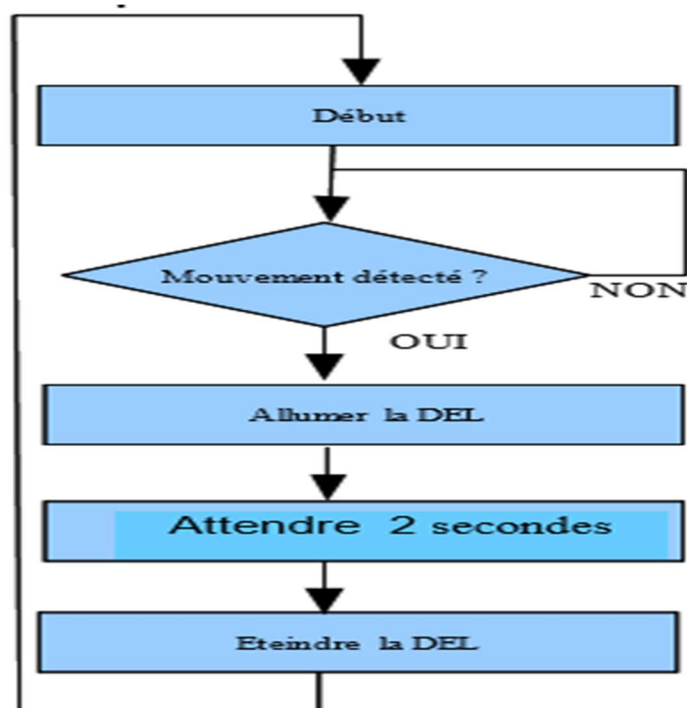


Figure III.9: Organigramme de fonctionnement de capteur PIR sur l'Arduino Méga2560

III.2.4 Programmation du capteur infrarouge Passif PIR :

```
piir$
// le temps que nous donnons au capteur de calibrer (10 à 60 secondes selon la fiche technique)
int calibrationTime = 30;
int ledPin = 8;           // choisissez la broche pour la LED
int inputPin = 7;        // choisissez la broche d'entrée (pour le capteur PIR)
int pirState = LOW;     // nous commençons, en supposant qu'aucun mouvement n'a détecté
int val = 0;            // variable pour lire l'état de la broche

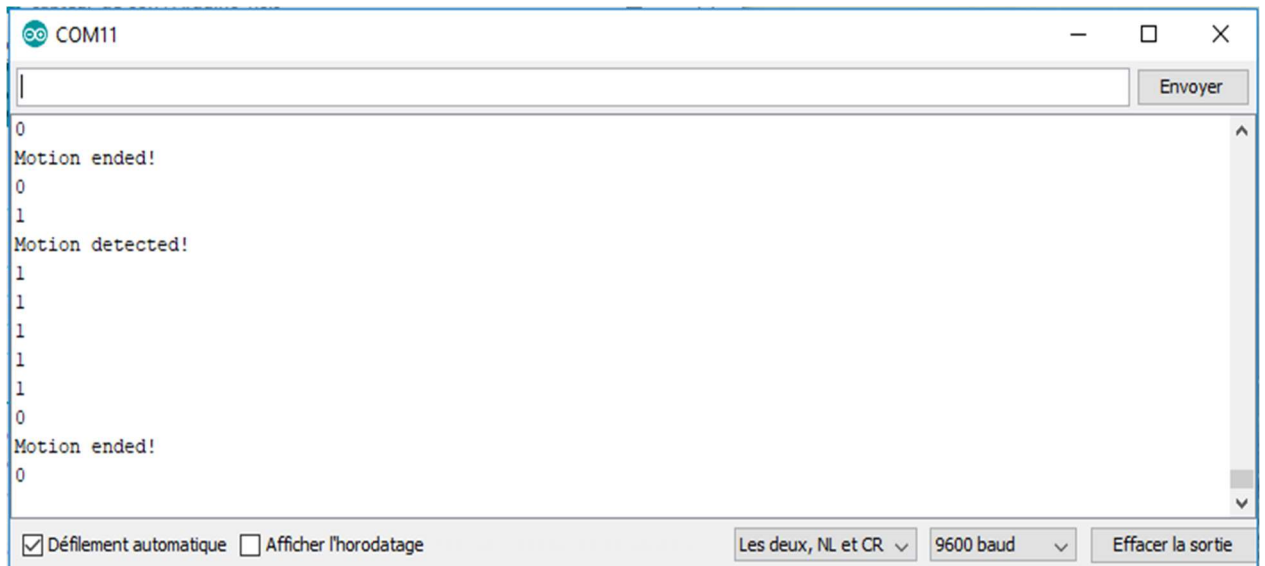
void setup() {
  pinMode(ledPin, OUTPUT); // déclare la LED comme sortie
  pinMode(inputPin, INPUT); // déclare le capteur comme entrée
  Serial.begin(9600);
  Serial.print("calibrating sensor ");
  for(int i = 0; i<calibrationTime; i++){
    Serial.print(".");
    delay(1000);
  }
}

void loop(){
  val = digitalRead(inputPin); // lecture de la valeur d'entrée
  Serial.println(val);
  if (val == HIGH) { // check sil'entréest HIGH
    digitalWrite(ledPin, HIGH); // allume la LED
    delay(150);
    if (pirState == LOW) {
      // nous venonsd'allumer
      Serial.println("Motion detected!");
    }
  }
  // Nous voulons seulement imprimer sur le changement de sortie, pas l'état
  pirState = HIGH;
  }
  } else {
    digitalWrite(ledPin, LOW); // éteignez la LED
    delay(300);
    if (pirState == HIGH){
      // nous venons de tourner off
      Serial.println("Motion ended!");
      // Nous voulons seulement imprimer sur le changement de sortie, pas l'état
    }
  }
  pirState = LOW;
  }
}
```

Figure III.10: Programme Arduino de capteur Infrarouge Passif

III.2.5 Résultats :

La valeur qui nous renvoi le capteur infrarouge passif c'est 0 ou 1, l'envoi de « 0 » lorsque y'a pas de signal et le « 1 » lorsqu'il détecte une variation infrarouge en même temps la LED s'allume.



```
COM11
|
|
0
Motion ended!
0
1
Motion detected!
1
1
1
1
1
1
0
Motion ended!
0
```

Figure III.11: Détection de Mouvement par le capteur PIR

III.3 Capteur d'humidité et de température DHT11 :

Un Capteur d'humidité et de température est un capteur numérique de température et d'humidité de base à très faible cout. Il est constitué d'un capteur d'humidité capacitif et d'une thermistance pour mesurer l'air ambiant et générer un signal numérique sur la broche de donnés (aucune broche analogique n'est requise). Il est relativement simple à utiliser, mais nécessite un minutage minutieux pour récupérer les données. [28]

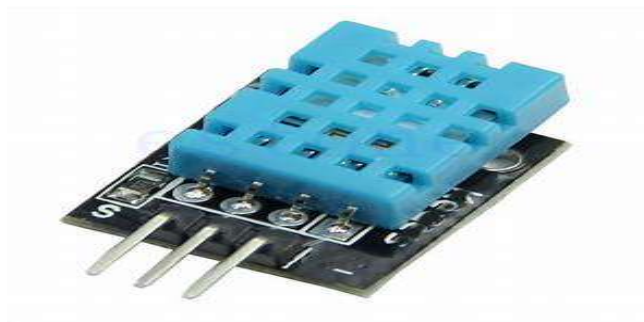


Figure III.12: Capteur d'humidité et de température DHT11

III.3.1 les caractéristiques du capteur DHT11 :

Humidité (relative %)	20 ~ 80%
Précision (humidité)	+/- 5%
Température	0 ~ +50°C
Précision (température)	+/- 2°C
Fréquence mesure max	1Hz (1 mesure par seconde)
Tension d'alimentation	3 ~ 5 volts
Stabilité à long terme	+/- 1% par an

Tableau III.2 Les spécifications techniques de capteur

III.3.2 principe de Fonctionnement : [29]

Une fois le capteur est mis sous tension, il faut l'équivalent de 2 secondes pour devenir stable, pendant cette période le capteur teste la température et l'humidité de l'environnement et les enregistre. Une fois terminé, il entre automatiquement dans le mode veille.

La communication commence lorsque le maître MCU envoie un signal de départ où le capteur sort du mode veille.

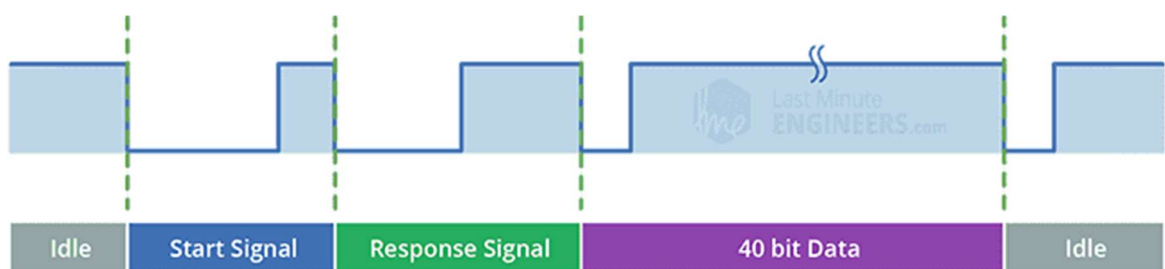


Figure III.13: Chronogramme de capteur de température et d'humidité

Signal de départ: pour demander une lecture, le microcontrôleur tire la ligne de données LOW pendant les 800µs, après ce temps la ligne est nouveau élevée, mais pour un temps beaucoup plus court de 20 us, cette séquence agit comme un signal de démarrage et réveille le capteur de son état.

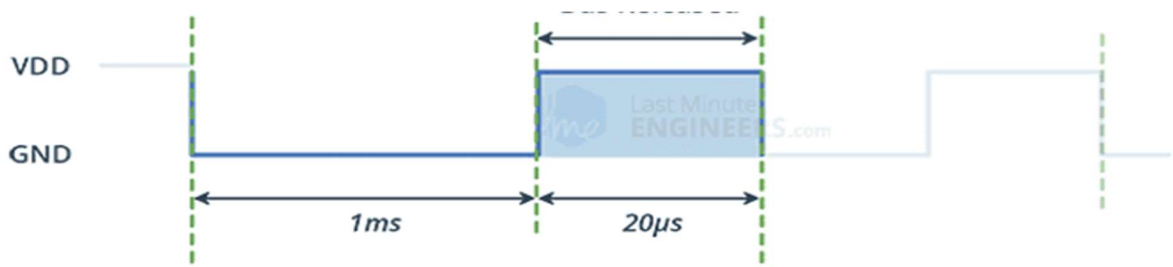


Figure III.14: Le signal de départ

Signal de réponse : une fois le signal de départ terminé, le capteur envoie un signal de réponse, et teste à nouveau la température et l'humidité de l'environnement, puis enregistre les données relatives et se prépare à la transmission des données.

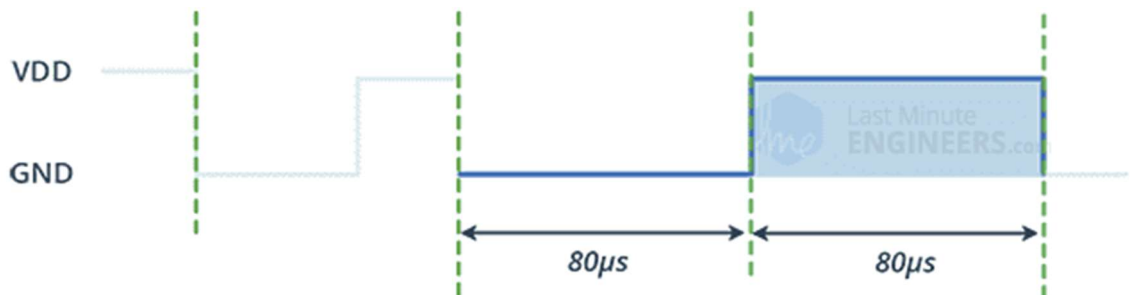


Figure III.15: Signal de réponse

Données 40bits : lorsque le signal de réponse est terminé, le capteur tire la ligne de données LOW pendant 50µs. Ensuite la durée de l'impulsion HIGH détermine l'état de bit. Si la ligne est HIGH pendant 25-28 µs, elle indique le bit « 0 », si l'impulsion est de 70 µs elle indique le bit « 1 ».

Une fois que tous les 40 bits sont transmis, le capteur tire la ligne de données LOW pendant 50 ms et entre en mode veille automatiquement.

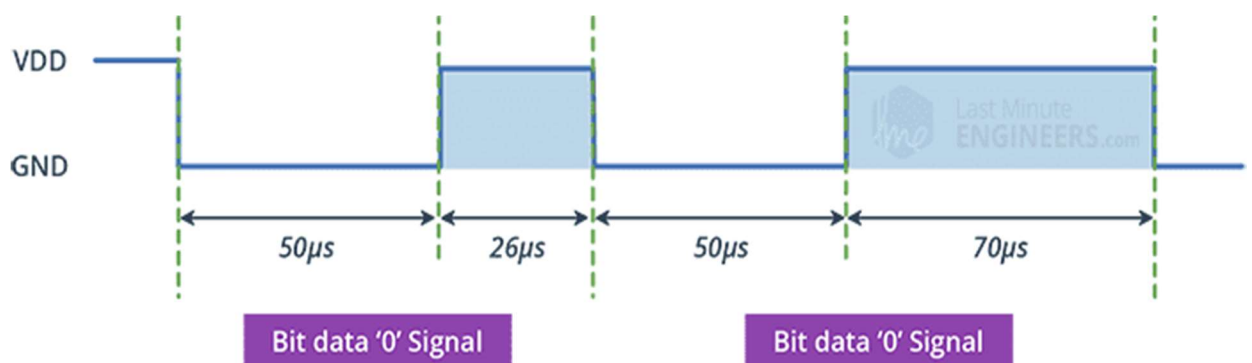


Figure III.16: Lecture des données valeurs données 40bits

III.3.3 Schéma de montage sur Arduino :

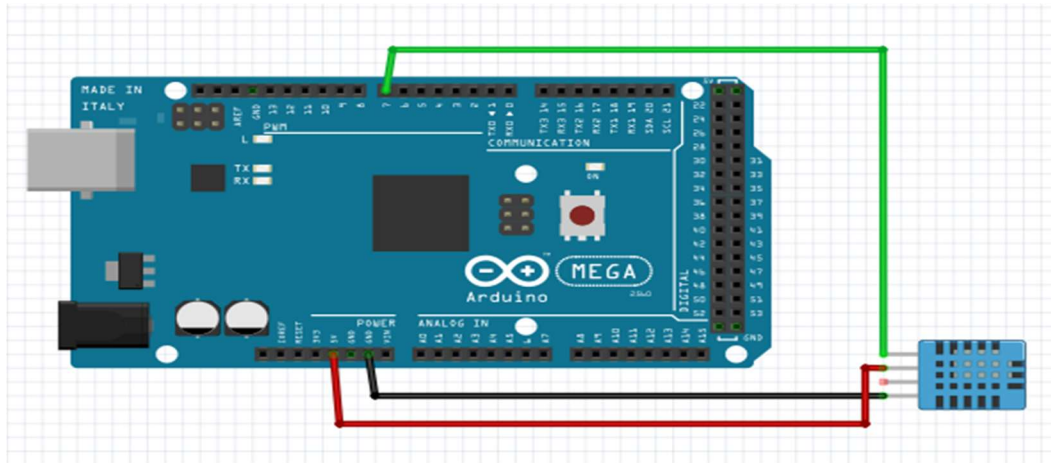
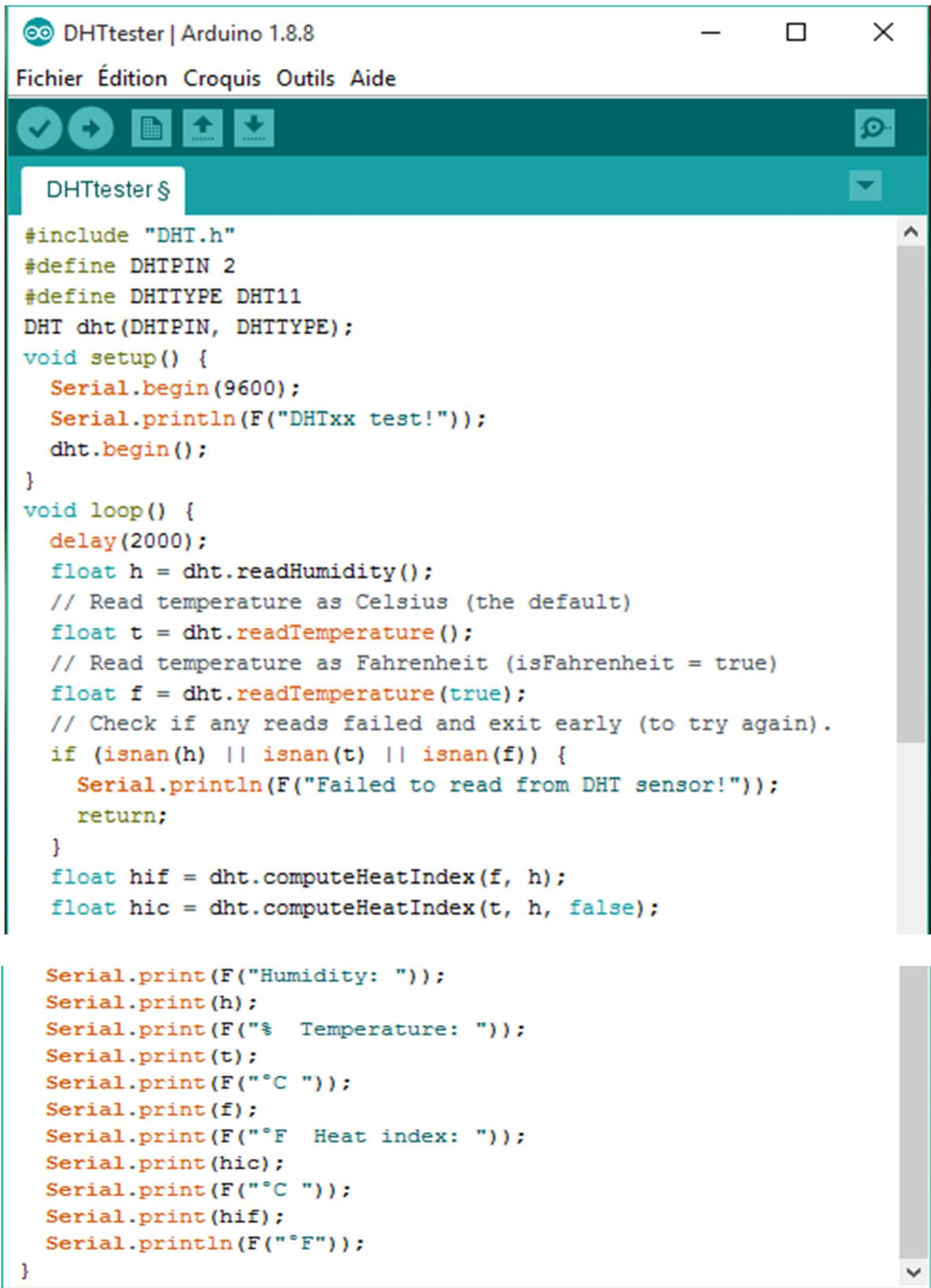


Figure III.17: Montage du capteur d'humidité et de température DHT11

III.3.4 Programmation du capteur Ultrasons DHT11 :



```
#include "DHT.h"
#define DHTPIN 2
#define DHTTYPE DHT11
DHT dht(DHTPIN, DHTTYPE);
void setup() {
  Serial.begin(9600);
  Serial.println(F("DHTxx test!"));
  dht.begin();
}
void loop() {
  delay(2000);
  float h = dht.readHumidity();
  // Read temperature as Celsius (the default)
  float t = dht.readTemperature();
  // Read temperature as Fahrenheit (isFahrenheit = true)
  float f = dht.readTemperature(true);
  // Check if any reads failed and exit early (to try again).
  if (isnan(h) || isnan(t) || isnan(f)) {
    Serial.println(F("Failed to read from DHT sensor!"));
    return;
  }
  float hif = dht.computeHeatIndex(f, h);
  float hic = dht.computeHeatIndex(t, h, false);

  Serial.print(F("Humidity: "));
  Serial.print(h);
  Serial.print(F("%  Temperature: "));
  Serial.print(t);
  Serial.print(F("°C "));
  Serial.print(f);
  Serial.print(F("°F  Heat index: "));
  Serial.print(hic);
  Serial.print(F("°C "));
  Serial.print(hif);
  Serial.println(F("°F"));
}
```

Figure III.18: Programme Arduino de capteur DHT11 sur l'IDE

III.3.5 Résultats :

Les Résultats Obtenus sont montrées via le moniteur série, les valeurs de température et d'humidité mesurées par le capteur DHT11 sont illustrées dans la figure (III.19) comme suit :

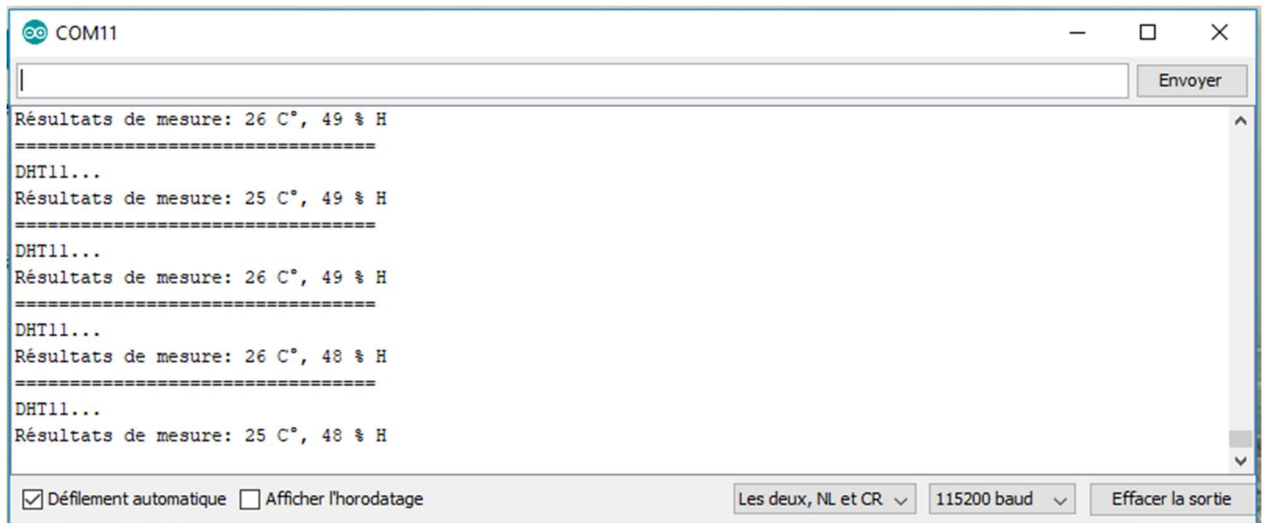


Figure III.19: Les valeurs de température et d'humidité détectent par DHT11

III. 4 Capteur de flamme :

Un détecteur de flamme est un capteur conçu pour détecter la présence d'une flamme ou d'un incendie, dans les environnements extrêmement dangereux. Il détecte le rayonnement infrarouge émis par le feu ou par toute source lumineuse et le convertit en signaux analogiques et numériques que le microcontrôleur peut traiter.

Il existe plusieurs types de capteurs de flamme tels que les détecteurs composés de capteurs Ultra-Violet, détecteurs dotés de capteurs Infra-Rouge et les détecteurs combinant IR et UV.

Certains d'entre eux déclenchent une alarme, et d'autres peuvent activer un système d'extinction d'incendie ou désactiver une conduite de combustible. [30]

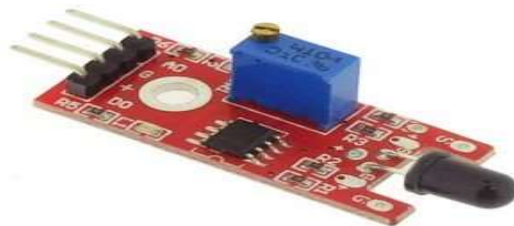


Figure III.20: Capteur de flamme

III.4.1 Les Caractéristiques :[31]

- Alimentation :5 Vcc
- Plage de mesure 760 à 1100 nm
- Sorties : A0, GND, Vcc, et D0
- Température de service : - 40 °c à + 85°c
- Dimensions : 42*16*15 mm
- Sortie numérique : un signal est émis si une flamme est détectée
- Sortie analogique : mesure directe du capteur

III.4.2Principe de fonctionnement :

Une fois Les flammes émettent des rayonnements électromagnétiques dans les longueurs d'ondes IR et les longueurs d'ondes UV, selon la source de combustible. Le détecteur de flamme répond aux rayonnements électromagnétiques émis par une flamme, en les distinguant des rayonnements interférents présents dans l'environnement d'utilisation. [32]

III.4.3 Schéma de montage sur Arduino :

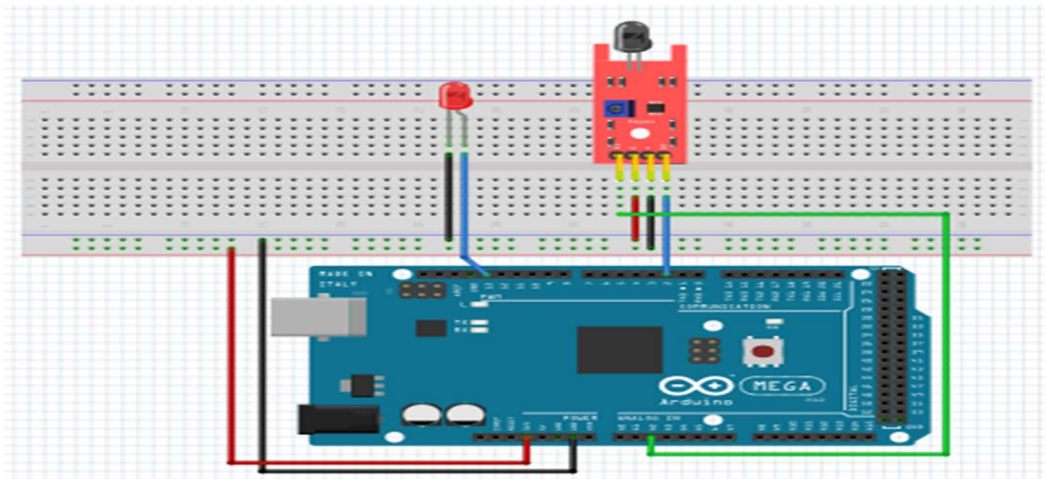


Figure III.21: Montage de capteur de flamme sur arduino méga2560

III.4.4 code de programmation de capteur de flamme :

```
flamme §
// Définition des broches utilisées
int Led = 13 ;// LED
int capt_num = 2; // Interface Capteur Flamme
int capt_ana = A3; // Interface Capteur Flamme
int val ;
float capteur_flamme;
void setup ()
{
  pinMode(Led, OUTPUT); // Définir l'interface de la LED en sortie
  pinMode(capt_num, INPUT); // Définir l'interface du capteur de fl
  pinMode(capt_ana, INPUT); // Définir l'interface du capteur de fl
}
void loop ()
{
  capteur_flamme = analogRead(capt_ana);

  val = digitalRead (capt_num) ;
  if (val == HIGH) // Quand le capteur détecte une flamme, la le
  {
    digitalWrite (Led, HIGH);
  }
  else
  {
    digitalWrite (Led, LOW);
  }
  delay(2000);
}
```

Figure III.22: Exemple de code de programmation de capteur

III.5 Servomoteur :

Un servomoteur est un système qui a pour but de produire un mouvement précis en réponse à une commande externe, c'est un actionneur qui mélange l'électronique, la mécanique et l'automatique. [33]

En réalité un servomoteur est un moteur continu équipé d'un réducteur (des engrenages), dont l'objectif est de réduire la vitesse et d'augmenter le couple (la puissance) ; et d'un potentiomètre qui permet au servomoteur de garder l'angle d'inclinaison choisit comme c'est indiqué dans la figure (III.24). Le couple indique la relation entre la longueur du palonnier qui est le bras de servomoteur et la force exercée sur le bras par le poids de l'objet que l'on souhaite mettre en mouvement. [34]



Figure III.23: Les composants de servomoteur

✚ Connectique :

Le servomoteur a besoin de trois fils de connexion pour fonctionner, deux fils servent à son alimentation, le dernier étant celui qui reçoit le signal de commande :

- Rouge : pour alimentation positive (4.5 v à 6 v en général)
- Noir ou marron : pour la masse 0 V
- Orange, jaune, blanc... : entrée de signal de commande.

✚ Le signal :

Le servomoteur a un angle de rotation limité de 0° à 180° , Il faudra un signal carré d'une période de 20 ms, généralement la commande se fera de 1 ms à 2 ms. 1 ms pour 0° et 2 ms pour 180° entre les deux 1.5 ms pour 90° . [35]

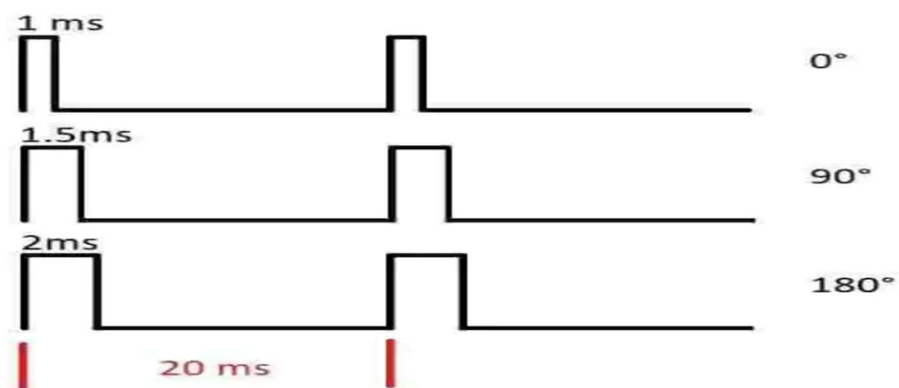


Figure III.24: Diagramme d'impulsions de servomoteur

III.6 Le moteur Pas à Pas :

Le moteur Pas à Pas est un moteur électrique d'une conception particulière. Sa structure permet par une commande électronique appropriée d'obtenir une rotation du rotor d'un angle égal à n fois un angle élémentaire appelé « pas ». [36]

Est un moteur à courant continu qui avance d'un seul pas lorsque le sens de courant dans une des bobines change.

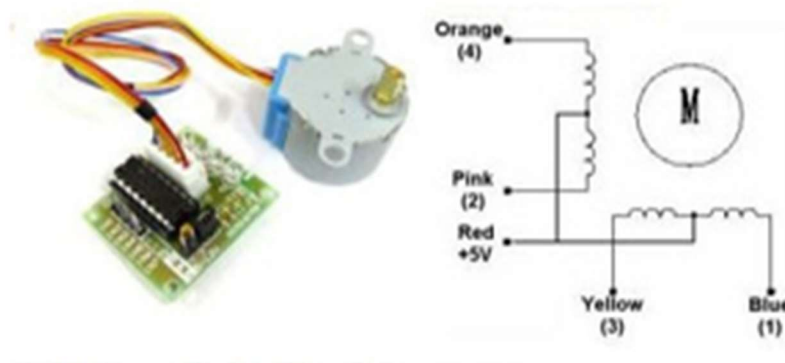


Figure III.25: Le moteur Pas à Pas

III.6.1 Caractéristiques : [37]

- Alimentation : 5 Vcc
- Résistance : 50 ohms
- Intensité : 25 mA
- Réduction : 1/64
- Nombre de Pas : 64 (4096 en sortie d'axe après la réduction)

Chapitre III Programmation des capteurs et des actionneurs sur Arduino

- Niveau sonore : 40 dB.
- Entraxe de fixation : 35 mm
- Axe : Ø5 mm avec double méplat (épaisseur 3 mm)
- Longueur de l'axe : 8 mm

Conclusion

Dans ce chapitre, nous avons simulé et implémenté quelques capteurs de base sur Arduino Méga2560 comme le capteur de distance Ultrasons HCs-SR04, le capteur infrarouge Passif, le capteur d'humidité et température DHT11, et capteur de flamme. Suivi des codes de programmation choisis pour ces derniers. Ensuite, nous avons défini quelques actionneurs utiles tels que le servomoteur et moteur pas à pas. Finalement Ces tous ces éléments basiques nous permettront de réaliser notre maquette dans le chapitre suivant.

Chapitre IV

Réalisation d'une maison
intelligente

Introduction

Le projet « smart house » est réalisé dans le but de répondre à un ensemble de besoins qui spécifient précisément les services demandés et attendus par l'utilisateur. Concernent principalement le confort (commande à distance d'appareils), la sécurité (protection contre les intrusions, détection d'incendie).

Dans ce chapitre nous allons mettre notre travail dans son contexte général. Tout d'abord, nous commençons par faire une présentation de la structure de la maison. Et nous présentons le cahier de charge du système et tout périphérique utilisé. Et aussi l'application exploitée comme outil de commande à distance.

IV.1 Fabrication de la maison

IV.1.1 Structure de la maison :

Nous allons construire une maquette de taille de 56 cm sur 55 cm constituée d'une chambre, un salon, une cuisine, une salle de bain, un couloir et un garage externe (figure IV.1).

<p>Chambre : LED DHT11</p>	<p>Salle de bain : LED</p>	<p>Salon : LED Servomoteur</p>
<p>Cuisine : Capteur de Flamme+ LED</p>	<p>Couloir : LED PIR+ Buzzer</p>	
<p>Garage</p>	<p>Jardin : Relais +une lampe</p>	
<p>La porte : moteur pas à pas Ultra-sons HC-SR04</p>		

Figure IV.1 : Illustration de la maquette utilisée dans le projet

IV.1.2 Périphériques utilisés :

Dans chaque partie dans la maquette de la maison, nous allons présenter les périphériques exploités comme est montré dans le tableau (IV.1).

Localisation	Périphériques
Couloir	LED PIR+ Buzzer
Salon	Servomoteur et LED
La chambre	LED DHT11
La cuisine	Flamme + LED
La salle de bain	LED
Garage	Moteur pas à pas
Jardin	Relais + lampe Ultra-son HC-SR04

Tableau IV.1 : Les périphériques utilisés dans les parties de maquette

IV.2 Automatisation de la maison :

Les maisons modernes sont le plus souvent équipées d'un système domotique. Ce système comprend non seulement la gestion centralisée des différents systèmes d'éclairage et de chauffage, mais aussi de l'ouverture et de la fermeture des portes et des fenêtres, mais aussi de la sécurité.

1. Portail automatique :

Les portails automatiques font partie des équipements indispensables pour les maisons modernes. L'automatisation d'un portail permet d'améliorer grandement son confort d'utilisation et sa sécurité.

En effet, l'intervention d'une personne ne sera plus nécessaire pour ouvrir et fermer le portail ou un garage dans notre cas. Cela évite notamment de sortir de son véhicule pour ouvrir le garage afin d'accéder à la cour de la maison.

Le matériel que nous avons utilisé est le suivant :

- ✓ Arduino Méga 2560 ;
- ✓ moteur pas à pas ;
- ✓ ESP8266-01 ;

- ✓ l'interface de commande à distance RemoteXY.



Figure IV.2: Montage réel de moteur pas à pas

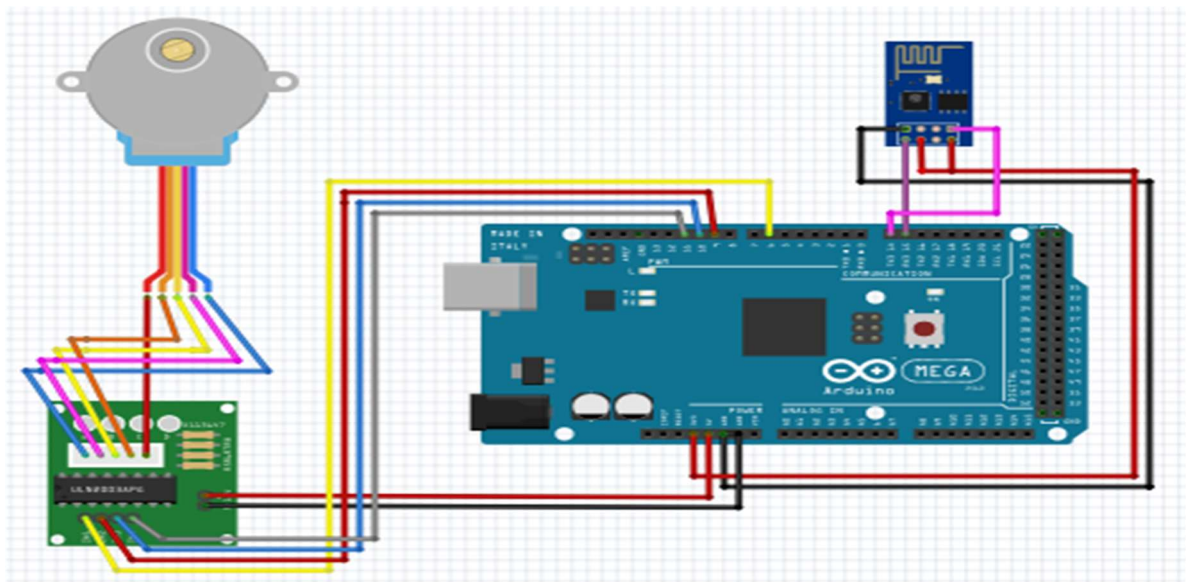


Figure IV.3: Montage moteur pas à pas sur Arduino Méga et le module wifi esp 8266

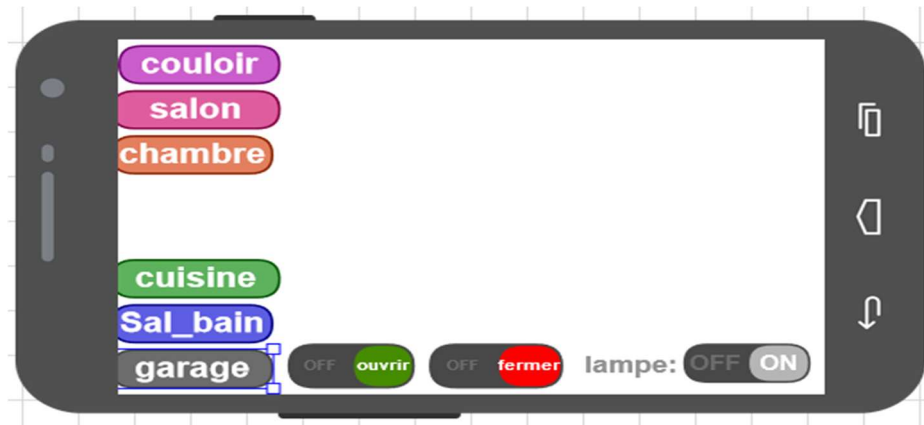


Figure IV.4: Interface d'ouverture et de fermeture de garage via smartphone

2. Système d'allumage :

Dans chaque coin de la maquette que nous avons fabriquée, nous avons installé un système d'allumage, c'est le cas dans la salle de bain, on allume la LED à travers l'interface RemoteXY implémenté dans le smartphone comme est indiqué sur la figure (IV.5).

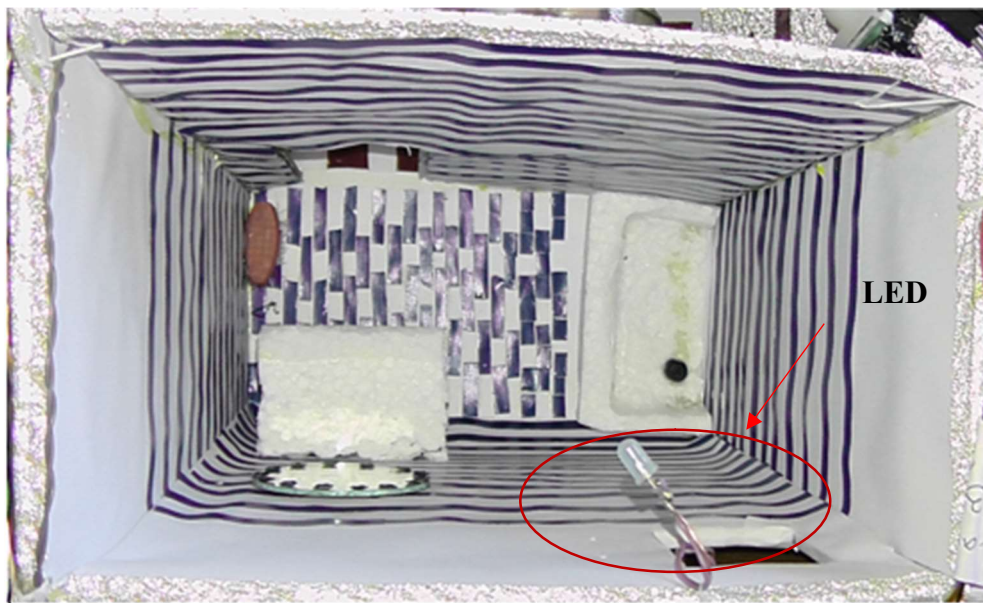


Figure IV.5: Montage réel d'une LED

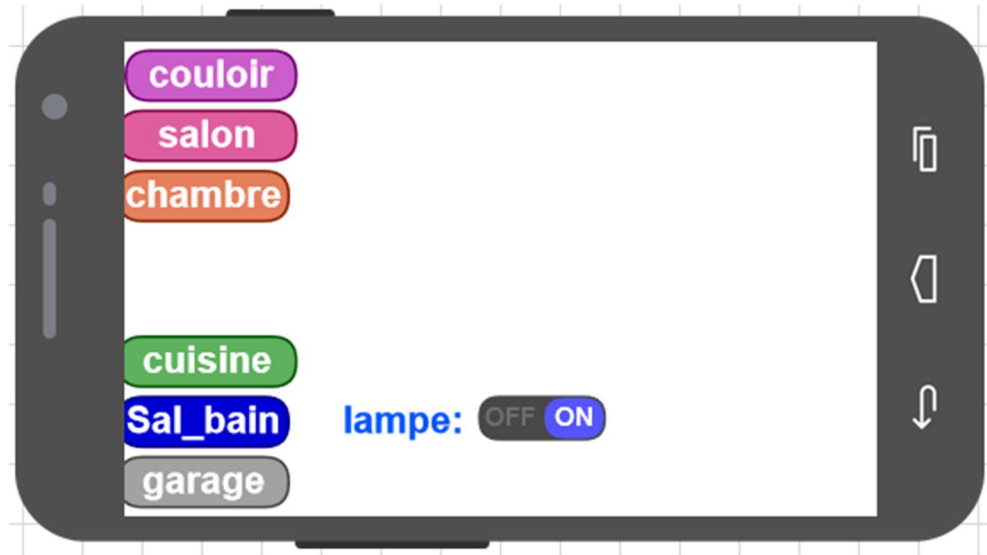


Figure IV.6:Interface d'allumer une LED via smartphone

3. Système de détection de feu :

La sécurité est devenue un élément fondamental dans le choix d'une maison. Et l'une des plus grande crainte d'accident reste l'incendie.

Pour une bonne sécurité, nous avons installé un système de détection de feu dans la cuisine. Ce système signale la présence d'un incendie dans une maison à travers l'application sur le smartphone. Le matériel utilisé est le suivant :

- ✓ Une carte Arduino Méga 2560 ;
- ✓ détecteur de flamme ;
- ✓ module wifi ESP8266-01 ;
- ✓ interface de contrôle RemoteXY.

ainsi une LED qui va s'allumer et s'éteindre à distance par cette interface (figure IV.7).

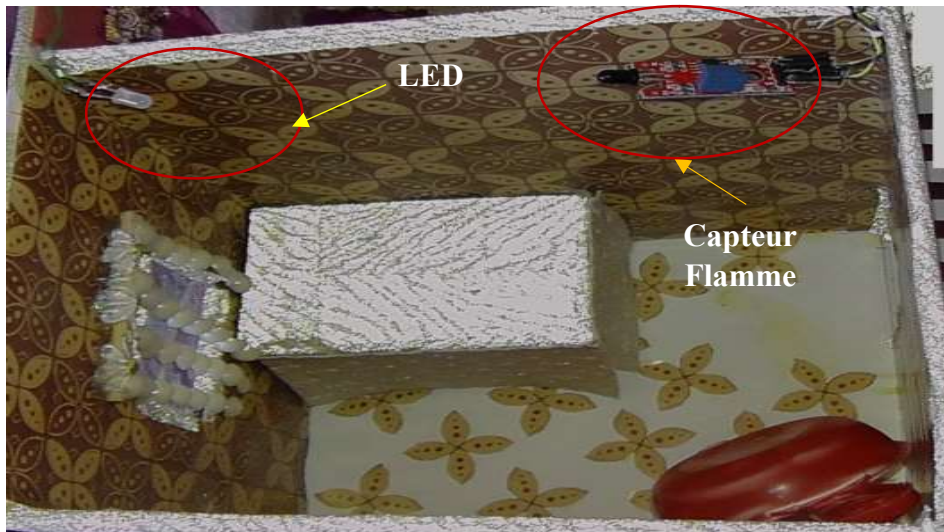


Figure IV.7: Montage réel de capteur flamme

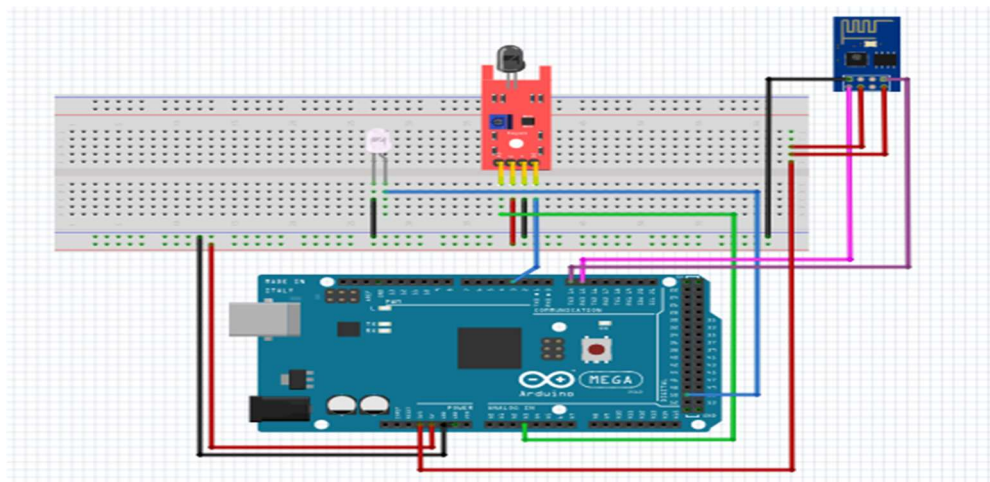


Figure IV.8:Montage de capteur de flamme avec Arduino Méga et le module wifi esp 8266

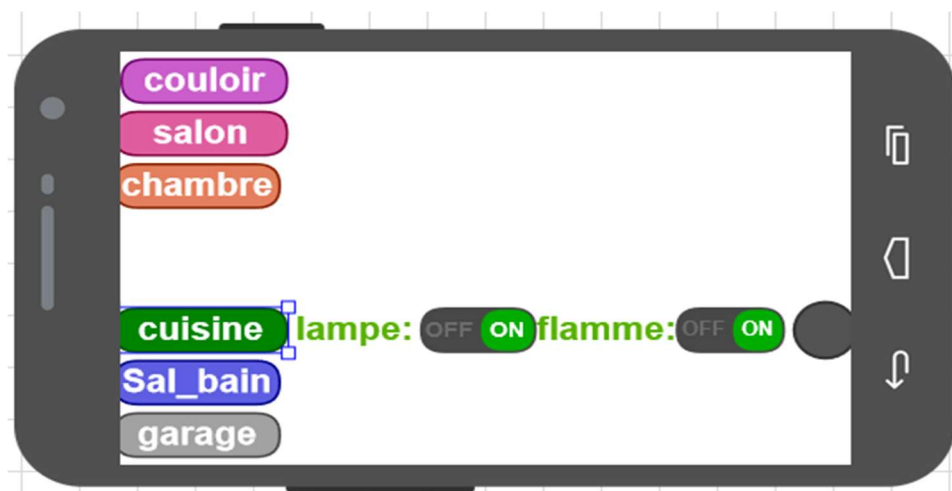


Figure IV.9:Interface de commande de capteur de flamme

4. Gestion de l'énergie et de l'alimentation :

Ce dispositif permet d'améliorer la gestion d'énergie et d'alimentation, il prend en charge la détection de la température et de l'humidité dans une chambre de la maison, dans la maquette, ce qui permet de rationaliser des charges électriques et de déconnecter l'équipement non utilisé prioritairement sur la base de la consommation d'énergie dans un temps donné. Nous avons utilisé comme matériel :

- ✓ Une carte Arduino Méga 2560 ;
- ✓ Un capteur DHT 11 ;
- ✓ Esp8266-01 ;
- ✓ interface d'affichage RemoteXY.

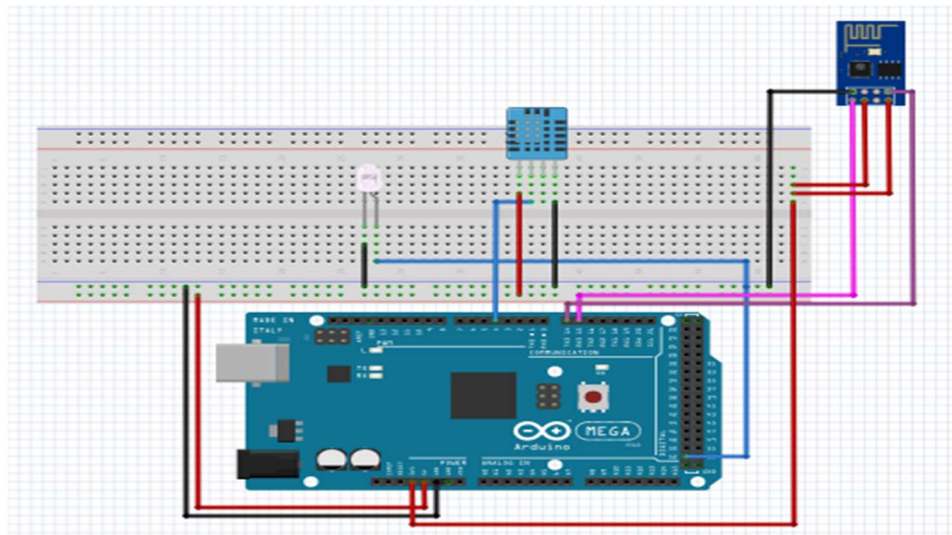


Figure IV.10:Montage de DHT11 avec Arduino Méga et le module wifi esp 8266



Figure IV.11: Montage réel de capteur DHT11

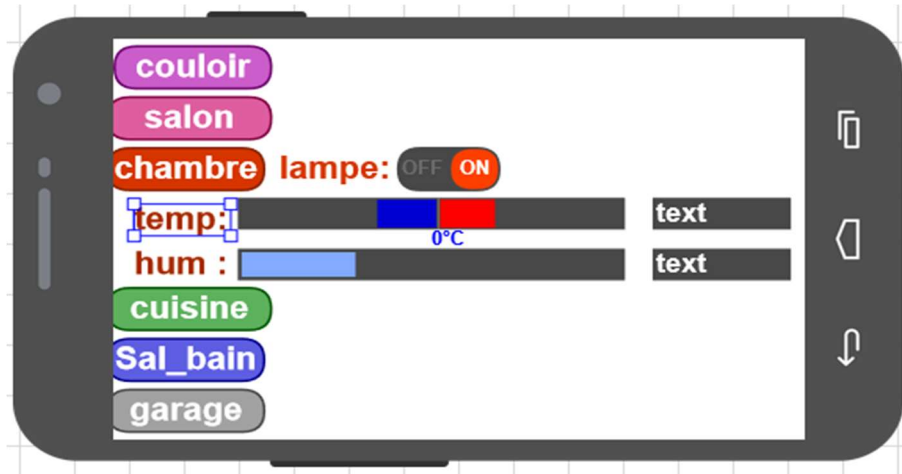


Figure IV.12: Interface de contrôle de température et d'humidité

5. L'automatisation de rideau fenêtre :

Dans une maison, le confort est placé au premier plan, et l'utilisation d'un système d'ouverture et de fermeture automatique d'une fenêtre est une nouvelle manière de gérer un logement et d'augmenter ce confort, alors même de faciliter la vie au quotidien. Pour cela nous avons pensé à ce dispositif qui est le servomoteur qui va nous permet de commander avec une simple télécommande l'ouverture et la fermeture les rideaux de salon, même en étant allongé dans notre lit. Le matériel que nous avons utilisé la carte arduino Méga et :

- ✓ servomoteur 9g ;
- ✓ ESP 8266-01 ;
- ✓ interface RemoteXY.

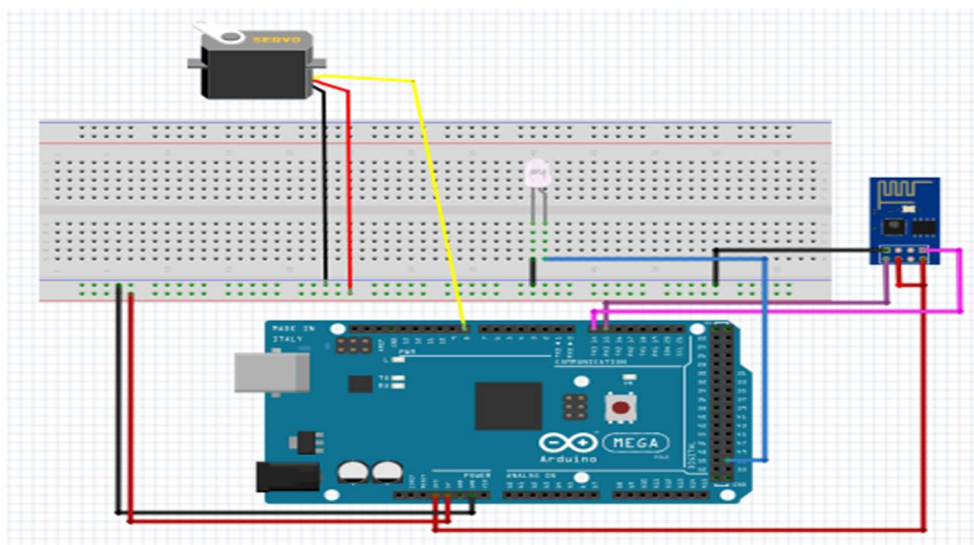


Figure IV.13:Montage de servomoteur avec l'Arduino Méga et l'Esp 8266



Figure IV.14: Montage réel de servomoteur

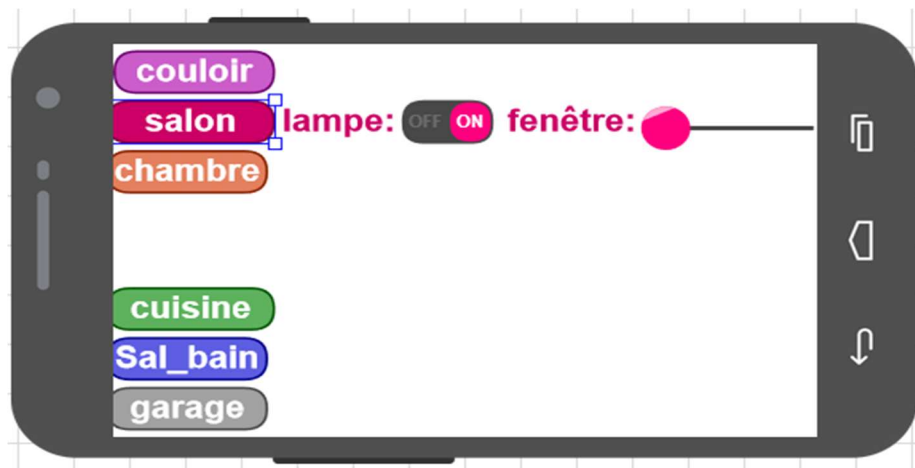


Figure IV.15: Interface d'ouverture et de fermeture de fenêtre

6. Système d'alarme :

Le domicile d'une personne est pour elle l'endroit qui lui est le plus cher, c'est à cet endroit que se trouvent la plupart de ses biens, et c'est dans le but de protéger ces richesses que les alarmes maison et les systèmes de sécurité existent.

Les alarmes permettent d'alerter le propriétaire du domicile dans le cas où ses biens seraient mis en danger.

Pour cela nous avons choisi les outils ci-dessous pour réaliser tel système dans notre maquette dans le couloir, grâce à la carte arduino Méga 2560 avec les accessoires suivants :

- ✓ capteur PIR ;
- ✓ Buzzer ;
- ✓ Esp8266-01 ;

- ✓ interface d'affichage RemoteXY.

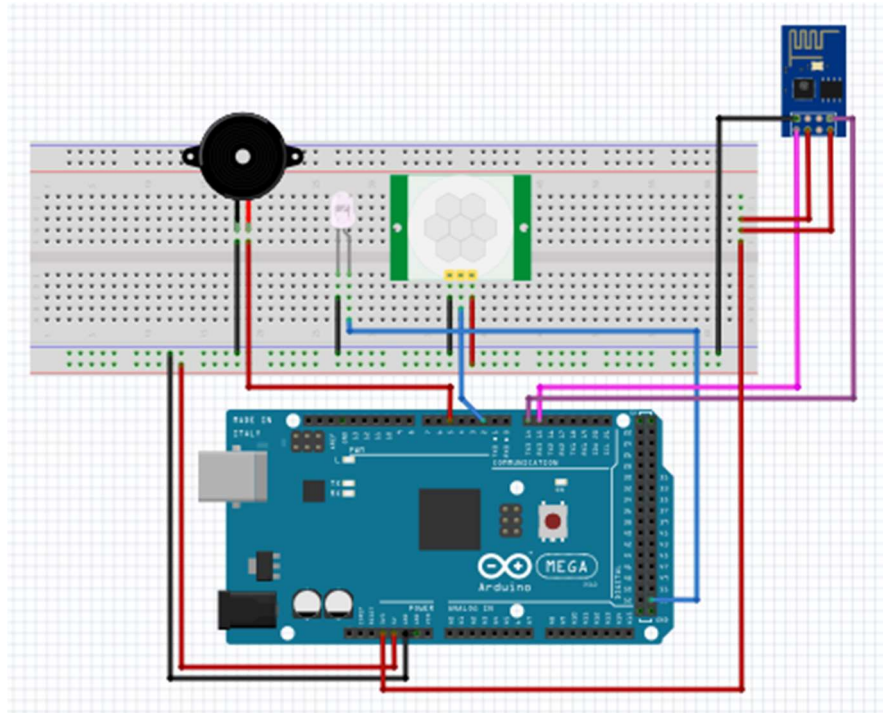


Figure IV.16:Montage du système Alarme avec l'Arduino MEGA et Esp8266

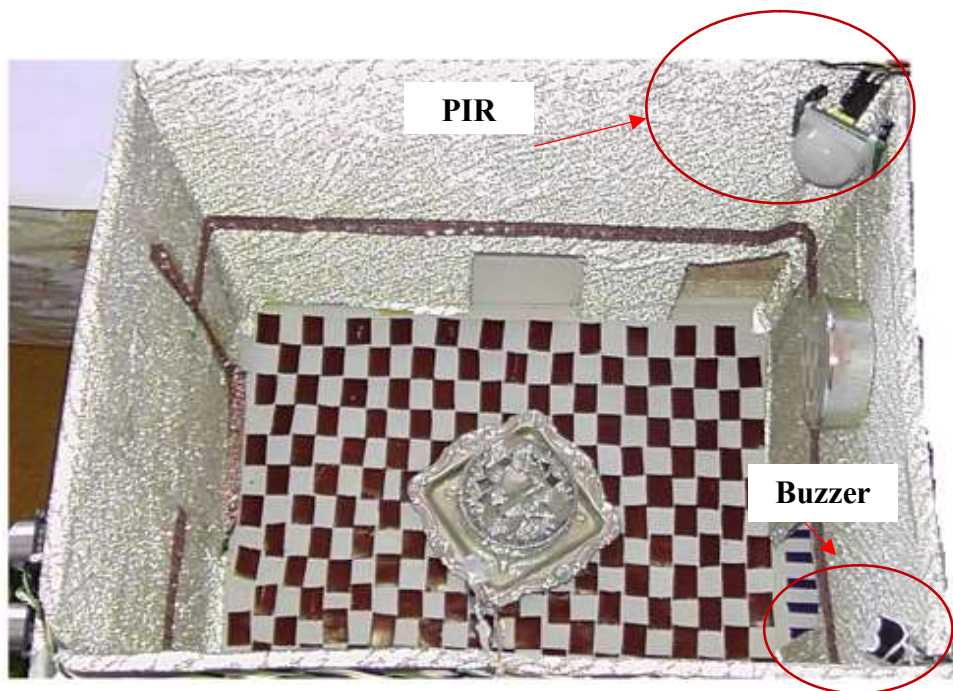


Figure IV.17: Montage réel de capteur PIR et Buzzer



Figure IV.18: Interface de commande le capteur de mouvement

7. Système d'éclairage externe :

Eclairer correctement un jardin ou tout autre espace extérieur d'un habitat est essentiel, non seulement d'un point de vue esthétique, mais aussi, et surtout, de celui de la sécurité à la maison.

Ce système de commande d'éclairage à distant est particulièrement pratique pour réduire le coût de fonctionnement d'un système d'éclairage extérieur.

Le matériel que nous avons utilisé est le suivant :

- ✓ carte Arduino Méga ;
- ✓ HC-SR04 ;
- ✓ Relais 5V/ 220 V ;
- ✓ une lampe ;
- ✓ Esp8266-01.



Figure IV.19: Montage réel de système d'éclairage

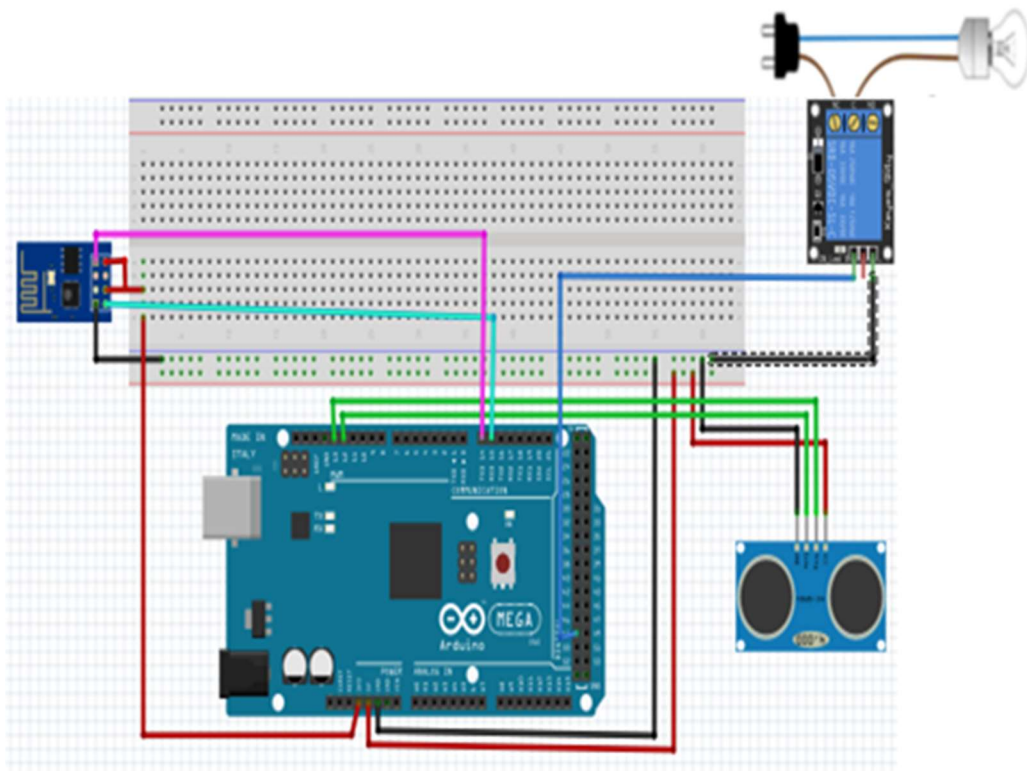


Figure IV.20: Montage de relais et d'une lampe et HC-SR04 Avec Arduino Méga et Esp8266

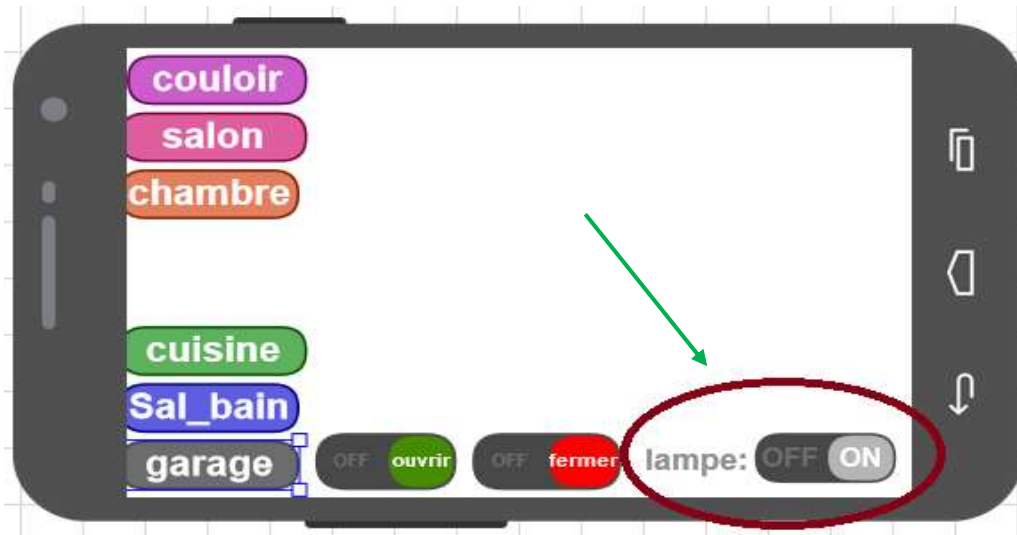


Figure IV.21:Interface d'allumage d'une lampe

Le programme final du schéma global de la figure (IV.24) situé en Annexe.

IV.3 Centralisation des commandes : Interface RemoteXY

Pour rendre la maison connectée et commandable à distance, nous avons exploité une carte wifi Esp 8266-01 pour Arduino Méga 2560, et comme interface de commande nous avons choisi RemoteXY.

RemoteXY est un moyen simple, il permet de créer et d'utiliser une interface graphique mobile pour les cartes Arduino à contrôler via un smartphone ou une tablette des objets connectés. En utilisant l'éditeur d'interfaces graphiques mobiles. Le système comprend :

- Editeur d'interfaces graphiques mobiles pour cartes contrôleur.
- Application mobile RemoteXY qui permet de se connecter au contrôleur et de le contrôler via une interface graphique.

L'interface graphique prend en charge les éléments suivants :



Figure IV.22: Les éléments de l'interface de RemoteXY

Le générateur de code source supporte les contrôleurs suivants :

- Arduino UNO, Arduino MEGA, Arduino Leonardo, Arduino Nano, Arduino MICRO.

Il prend en charge les modules de communication suivants:

- Wifi ESP8266 ;
- Bluetooth : HC-05, HC-06, HM-10 ou compatible ; BT Classique et BLE sur ESP32 ;
- Bouclier Ethernet W5100 ;
- Wifi ESP32.

IV.4 La maquette finale :



Figure IV.23: Maquette finale de la maison intelligente

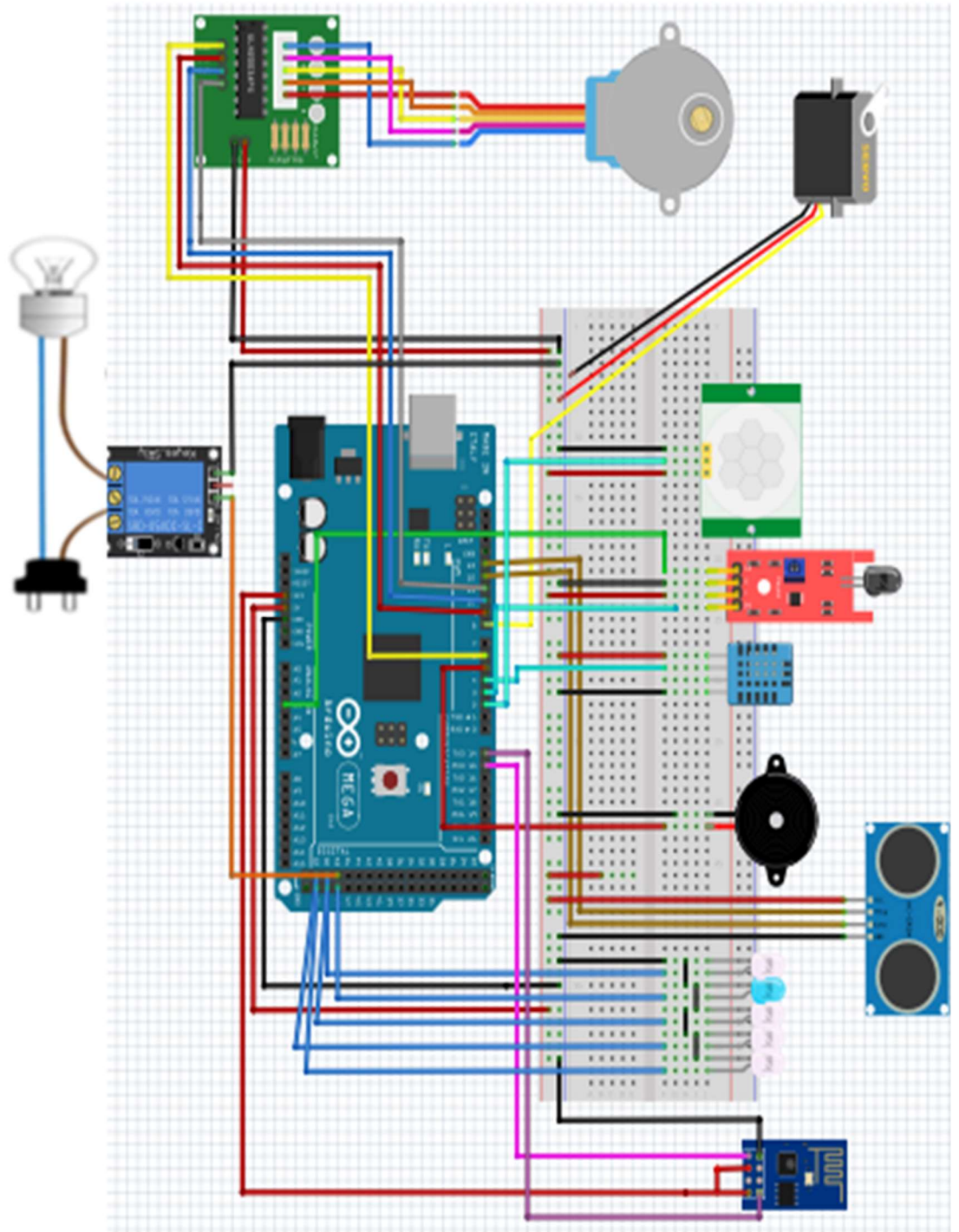


Figure IV.24: Montage final de la maquette réalisé avec fritzing

Conclusion

Nous avons consacré ce chapitre pour décrire la partie pratique et la réalisation de la maquette finale. Tout d'abord, Nous avons décrit la structure de la maison fabriquée, avec le montage réel de chaque périphérique utilisé dans chaque coin de la maquette. Ensuite une description sur l'application qui est l'interface de commande de notre maison à distance.

On peut dire que les systèmes de la maison intelligente sont des systèmes révolutionnaires voués à évoluer encore plus dans le futur. Ces systèmes permettent de contrôler l'entièreté d'une maison par un Smartphone.

Conclusion Générale

Conclusion générale

La domotique regroupe les technologies de l'électronique, de l'automatique, de l'informatique et des télécommunications permettant d'améliorer le confort, la sécurité, la communication et la gestion d'énergie d'une maison ou d'un lieu public. Dans ce projet nous nous sommes concentrés à la conception et à la réalisation d'un système d'IoT à distance utilisant un protocole de communication sans fil (wifi) avec un Smartphone.

Pour mieux expliquer cette technologie, on a défini tout d'abord dans le premier chapitre ses domaines d'application, son architecture et ses composants, Les technologies utilisées pour la communication sans fils, ainsi comment sécuriser et protéger notre système (matériel, logiciel et données).

Avant d'entrer dans la pratique on a consacré le deuxième chapitre pour présenter le matériel de base dont nous avons besoin pour réaliser notre système à savoir : la carte Arduino méga2560 (Gateway), ESP8266-01 pour la communication sans fils, les capteurs..., l'Arduino IDE qui joue un rôle très important dans la programmation de notre système.

Le troisième chapitre a deux buts essentiels, le premier est de savoir le principe de fonctionnement et les caractéristiques de chacun des capteurs ou des actionneurs que nous allons utiliser dans notre maquette (maison intelligente).Le deuxième intérêt est de réaliser plusieurs expérimentations avec la carte Arduino Méga2560 et l'Arduino IDE pour mieux se familiariser avec, ainsi pour assurer le bon fonctionnement du matériel choisi.

Le quatrième chapitre consiste à faire la réalisation pratique de notre système, nous avons développé une interface de commande sous Androïde grâce à l'application RemoteXY. Nous avons préparé les icônes et les labelles des commandes pour superviser les résultats.

Dans notre maison on a réalisé plusieurs systèmes : système d'alarme en cas de mouvement, la détection d'une flamme si elle existe, afficher la température et l'humidité et système d'éclairage et l'automatisation des portails et des fenêtres.

Comme perspectives, il est souhaitable dans la venir d'ajouté le réseau internet pour augmenter la distance de communication sans fils. Ou de faire des simulations sur des projets réelles (entreprise, maison, usine, jardin...) dans le but d'atteindre une nouvelle innovation.

Annexe

Fritzing :

Nous avons exploité ce logiciel durant la réalisation des montages, en fait fritzing est une initiative open source visant à aider à travailler de manière créative avec l'électronique interactive, il est possible de compléter la bibliothèque de composants et Chaque composant est défini à l'aide de 3 éléments : l'image du composant, le symbole du composant et la représentation du composant sur le circuit imprimé.

Programme :

```
////////////////////////////////////
// RemoteXY includelibrary //
////////////////////////////////////
// RemoteXY select connection mode and includelibrary
#define REMOTEXY_MODE __ESP8266_HARDSERIAL_POINT
#include<RemoteXY.h>
// RemoteXY connection settings
#define REMOTEXY_SERIAL Serial3
#define REMOTEXY_SERIAL_SPEED 115200
#define REMOTEXY_WIFI_SSID "RemoteXY"
#define REMOTEXY_WIFI_PASSWORD "12345678"
#define REMOTEXY_SERVER_PORT 6377
// RemoteXY configurate
#pragma pack(push, 1)
uint8_t RemoteXY_CONF[] =
{255, 11, 0, 27, 0, 37, 2, 8, 31, 4,
 2, 1, 24, 54, 18, 8, 1, 118, 26, 31,
 31, 111, 117, 118, 114, 105, 114, 0, 79, 70,
 70, 0, 2, 1, 44, 54, 19, 8, 1, 36,
```

26, 31, 31, 102, 101, 114, 109, 101, 114, 0,
79, 70, 70, 0, 131, 0, -1, 55, 23, 7,
1, 27, 31, 103, 97, 114, 97, 103, 101, 0,
131, 0, -1, 9, 24, 7, 2, 245, 31, 115,
97, 108, 111, 110, 0, 4, 128, 75, 10, 28,
7, 2, 246, 26, 131, 0, -1, 39, 24, 7,
3, 132, 31, 99, 117, 105, 115, 105, 110, 101,
0, 65, 10, 92, 1, 8, 7, 6, 2, 1,
74, 1, 17, 7, 6, 231, 26, 31, 31, 79,
78, 0, 79, 70, 70, 0, 131, 0, -1, 47,
23, 7, 4, 203, 31, 83, 97, 108, 95, 98,
97, 105, 110, 0, 66, 129, 47, 25, 27, 5,
5, 36, 26, 66, 129, 18, 33, 56, 5, 5,
193, 26, 66, 193, 18, 25, 29, 5, 5, 203,
26, 67, 4, 78, 25, 20, 5, 5, 31, 26,
11, 67, 4, 78, 33, 20, 5, 5, 31, 26,
11, 129, 0, 3, 26, 14, 5, 5, 204, 116,
101, 109, 112, 58, 0, 129, 0, 3, 33, 12,
5, 5, 204, 104, 117, 109, 32, 58, 0, 2,
1, 41, 9, 13, 7, 2, 246, 26, 31, 31,
79, 78, 0, 79, 70, 70, 0, 129, 0, 24,
10, 16, 5, 2, 245, 108, 97, 109, 112, 101,
58, 0, 129, 0, 56, 10, 18, 5, 2, 245,
102, 101, 110, 195, 170, 116, 114, 101, 58, 0,
129, 0, 46, 30, 5, 3, 5, 204, 48, 194,

176, 67, 0, 2, 1, 41, 39, 16, 7, 3,
133, 26, 31, 31, 79, 78, 0, 79, 70, 70,
0, 2, 1, 42, 46, 14, 6, 4, 206, 26,
31, 31, 79, 78, 0, 79, 70, 70, 0, 129,
0, 24, 40, 16, 5, 3, 119, 108, 97, 109,
112, 101, 58, 0, 129, 0, 24, 47, 16, 5,
4, 190, 108, 97, 109, 112, 101, 58, 0, 131,
0, -1, 17, 23, 7, 5, 49, 31, 99, 104,
97, 109, 98, 114, 101, 0, 129, 0, 24, 18,
16, 5, 5, 49, 108, 97, 109, 112, 101, 58,
0, 2, 1, 41, 17, 15, 7, 5, 50, 26,
31, 31, 79, 78, 0, 79, 70, 70, 0, 129,
0, 66, 56, 13, 4, 1, 28, 108, 97, 109,
112, 101, 58, 0, 2, 1, 80, 54, 18, 7,
1, 29, 26, 31, 31, 79, 78, 0, 79, 70,
70, 0, 131, 1, 0, 1, 23, 7, 6, 231,
31, 99, 111, 117, 108, 111, 105, 114, 0, 65,
4, 92, 38, 9, 9, 3, 129, 0, 55, 34,
18, 6, 5, 119, 0, 2, 1, 76, 39, 15,
7, 3, 133, 26, 31, 31, 79, 78, 0, 79,
70, 70, 0, 129, 0, 57, 40, 19, 5, 3,
119, 102, 108, 97, 109, 109, 101, 58, 0, 129,
0, 59, 2, 15, 5, 6, 231, 109, 111, 117,
118, 58, 0, 129, 0, 24, 2, 16, 5, 6,
231, 108, 97, 109, 112, 101, 58, 0, 2, 1,

41, 1, 17, 7, 6, 231, 26, 31, 31, 79,

78, 0, 79, 70, 70, 0

};

// this structure defines all the variables of your control interface

struct {

// input variable

uint8_t ouvrir; // =1 if switch ON and =0 if OFF

uint8_t fermer; // =1 if switch ON and =0 if OFF

int8_t salon; // =0..100 slider position

uint8_t couloir; // =1 if switch ON and =0 if OFF

uint8_t L2; // =1 if switch ON and =0 if OFF

uint8_t L4; // =1 if switch ON and =0 if OFF

uint8_t L5; // =1 if switch ON and =0 if OFF

uint8_t L3; // =1 if switch ON and =0 if OFF

uint8_t jardin; // =1 if switch ON and =0 if OFF

uint8_t flamme; // =1 if switch ON and =0 if OFF

uint8_t L1; // =1 if switch ON and =0 if OFF

// output variable

uint8_t led_1_g; // =0..255 LED Green brightness

int8_t level_temp_up; // =0..100 level position

int8_t level_hum; // =0..100 level position

int8_t level_temp_down; // =0..100 level position

char text_temp[11]; // string UTF8 end zero

char text_hum[11]; // string UTF8 end zero

uint8_t led_2_r; // =0..255 LED Redbrightness

```
// other variable

uint8_t connect_flag; // =1 if wireconnected, else =0

} RemoteXY;

#pragma pack(pop)

////////////////////////////////////

//      END RemoteXY include      //

////////////////////////////////////

#include<Servo.h>          //la bibliothèque de servomoteur
#include<Stepper.h>        //la bibliothèque de moteur pas à pas
#include<DHT.h>           //la bibliothèque de dht11

#define PIN_OUVRIR 6  ///
#define PIN_FERMER 6  ////
#define PIN_COULOIR 2  //////////
#define PIN_JARDIN 36  //////////
#define PIN_FLAMME 3  ////////// les pins de chaque
#define PIN_L1 53      ////////// commande
#define PIN_L2 51      //////////de notre interface
#define PIN_L3 52      //////////
#define PIN_L4 50      ////
#define PIN_L5 49     ///

#define DHTPIN 4      //pin d'entrée de dht11

#define eclirage 48 //pin de sortie d'éclairage

#define trigPin 36 //pin choisie pour Trig

#define echoPin 26 // pin choisie pour Echo

int inputPin = 2 // Pin d'entrée choisie pour surveiller le capteur PIR
```

```
intsonPin = 5; // pin de buzzer

intval_mouv = 0; // Variable utilisée pour lire l'entrée Pin 2, Pour commencer, nous

                //considérons qu'il n'y a pas de mouvement

intcapt_num = 3; // Interface Capteur Flamme port numérique

intcapt_ana = A3; // Interface Capteur Flamme port analogique

intval_flame ; // Variable utilisée pour lire l'entrée numérique de capteur de flamme

floatcapteur_flamme; //Variable utilisée pour lire l'entrée analogique de capteur de //flamme

intmonbreDePas = 48 * 64; // nombre de pas de moteur

Stepper monMoteur(monbreDePas, 9, 11, 10, 6); //déclaration des pins et le nom de //moteur

Servomyservo; //déclaration le nom de servomoteur

DHT dht(DHTPIN, DHT11); //déclaration le thype de capteur

void setup()

{

RemoteXY_Init ();

    /*******couloir*****/

pinMode (PIN_COULOIR, OUTPUT);

pinMode (PIN_L1, OUTPUT);

pinMode(sonPin, OUTPUT); // déclarer le son comme sortie

pinMode(inputPin, INPUT); // déclarer le senseur comme entrée

    /*******salon*****/

pinMode (PIN_L2, OUTPUT);

myservo.attach(8); // Pin d'entrée choisie pour SERVOMOTEUR

RemoteXY.salon = 50; //nous établissons la position initiale du curseur sur 50
```

```

/*****chambre*****/
pinMode (PIN_L3, OUTPUT);

dht.begin();          //initialisation de DHT11

/*****cuisine*****/

pinMode (PIN_L4, OUTPUT);

pinMode (PIN_FLAMME, OUTPUT);

pinMode(capt_num, INPUT); // Définir interface numérique du capteur de flamme en entrée
pinMode(capt_ana, INPUT); // Définir interface analogique du capteur de flamme en entrée

/*****salle de bain*****/

pinMode (PIN_L5, OUTPUT);

/*****garage et jardin*****/

pinMode (PIN_JARDIN, OUTPUT);

pinMode (PIN_OUVRIER, OUTPUT);

pinMode (PIN_FERMER, OUTPUT);

pinMode(trigPin, OUTPUT); //On définit Trig comme une sortie
pinMode(echoPin, INPUT); //On définit Echo comme une entrée
pinMode(eclairage, OUTPUT); //On définit la led comme une sortie
monMoteur.setSpeed(9); //La vitesse max de moteur (9 tour par minute)
}

voidloop()
{
RemoteXY_Handler ();

/*****couloir*****/

digitalWrite(PIN_L1, (RemoteXY.L1 == 0) ? LOW : HIGH);

```

```
if (RemoteXY.L1 == 1) {
digitalWrite(PIN_L1, HIGH); // Allumer la led
} else {
digitalWrite(PIN_L1, LOW); // Eteindre la led
}
digitalWrite(PIN_COULOIR, (RemoteXY.couloir == 0) ? LOW : HIGH);
if (RemoteXY.couloir == 1) {
val_mouv = digitalRead(inputPin); // Lire la valeur d'entrée
if (val_mouv == HIGH) { // Vérifier si l'entrée est HIGH
tone(5, 400); // Allumer le son (activité PIR)
RemoteXY.led_1_g = 255; //une led verte s'allume sur le téléphone
} else {
noTone(5); // Eteindre le son (plus d'activité PIR)
RemoteXY.led_1_g = 0; //sinon une led verte s'éteint sur le téléphone
}
} else {
noTone(5); // Eteindre le son (plus d'activité PIR)
RemoteXY.led_1_g = 0; //sinon une led verte s'éteint sur le téléphone
}
/*****salon*****/
digitalWrite(PIN_L2, (RemoteXY.L2 == 0) ? LOW : HIGH);
if (RemoteXY.L2 == 1) {
digitalWrite(PIN_L2, HIGH);
} else {
digitalWrite(PIN_L2, LOW);
```



```
} nous établissons la position initiale du curseur sur 50

int ms = RemoteXY.salon * 20 + 500; // convertit la valeur de la position du curseur, // allant de
0 à 100 en durée d'impulsion, pour contrôler le servomoteur

myservo.writeMicroseconds(ms);

/*****chambre*****/

digitalWrite(PIN_L3, (RemoteXY.L3 == 0) ? LOW : HIGH);

if (RemoteXY.L3 == 1) {
digitalWrite(PIN_L3, HIGH);
} else {
digitalWrite(PIN_L3, LOW);
}

float temp = dht.readTemperature(); // lire la valeur de la température
float hum = dht.readHumidity(); // lire la valeur de la humidité
dtostrf(temp, 0, 1, RemoteXY.text_temp);
dtostrf(hum, 0, 1, RemoteXY.text_hum);

if (temp < 0) {
RemoteXY.level_temp_up = 0;
RemoteXY.level_temp_down = min(-temp * 2, 100);
}

else if (temp > 0) {
RemoteXY.level_temp_up = min(temp * 2, 100);
RemoteXY.level_temp_down = 0;
}

else {
RemoteXY.level_temp_up = 0;
RemoteXY.level_temp_down = 0;
}
```

```
}  
  
RemoteXY.level_hum = hum;  
  
/*****cuisine*****/  
digitalWrite(PIN_L4, (RemoteXY.L4 == 0) ? LOW : HIGH);  
  
if (RemoteXY.L4 == 1) {  
digitalWrite(PIN_L4, HIGH);  
  
} else {  
digitalWrite(PIN_L4, LOW);  
  
}  
  
digitalWrite(PIN_FLAMME, (RemoteXY.flamme == 0) ? LOW : HIGH);  
  
if (RemoteXY.flamme == 1) {  
capteur_flamme = analogRead(capt_ana); //lire la valeur de pin analogique  
val_flame = digitalRead (capt_num) ; //lire la valeur de pin numérique  
  
if (val_flame == HIGH) { // Quand le capteur détecte une flamme  
RemoteXY.led_2_r = 255; // une led rouge s'allume sur le téléphone  
}  
else {  
RemoteXY.led_2_r = 0; //sinon une led rouge s'éteint sur le téléphone  
}  
  
}  
else {  
RemoteXY.led_2_r = 0; //sinon une led rouge s'éteint sur le téléphone  
}  
  
/*****salle de bain*****/  
  
digitalWrite(PIN_L5, (RemoteXY.L5 == 0) ? LOW : HIGH);  
  
if (RemoteXY.L5 == 1) {  
digitalWrite(PIN_L5, HIGH); //la led s'allume
```

```
} else {  
digitalWrite(PIN_L5, LOW); //sinon la led s'éteint  
}  
  
/*****garage et jardin*****/  
digitalWrite(PIN_JARDIN, (RemoteXY.jardin == 0) ? LOW : HIGH);  
if (RemoteXY.jardin == 1) {  
digitalWrite(PIN_JARDIN, HIGH);  
long duration, distance; // Déclare les deux variables distance et duration  
digitalWrite(trigPin, LOW);  
delayMicroseconds(2);  
digitalWrite(trigPin, HIGH);  
delayMicroseconds(10); //Trig envois pendant 10ms  
digitalWrite(trigPin, LOW);  
duration = pulseIn(echoPin, HIGH); // On calcule le temps pour l'aller-retour du signal  
distance =(duration / 2) / 29.1; // Distance proportionnelle à la durée de sortie  
if (distance < 30) //si la distance est <30cm  
{  
digitalWrite(eclairage, HIGH); // On allume l'éclairage  
delay(5000);  
} else {  
digitalWrite(eclairage, LOW); //sinon on éteint l'éclairage  
}  
} else {  
digitalWrite(PIN_JARDIN, LOW);  
} digitalWrite(PIN_OUVRIER, (RemoteXY.ouvrir == 0) ? LOW : HIGH);
```

```
if (RemoteXY.ouvrir == 1) {  
monMoteur.step(100); // 100 pas dans le sens positif  
  
}  
digitalWrite(PIN_FERMER, (RemoteXY.fermer == 0) ? LOW : HIGH);  
if (RemoteXY.fermer == 1) {  
monMoteur.step(-100); // 100 pas dans le sens negatif  
  
}}
```

Références WEBOGRAPHIE :

- [01] <https://www.sergentweb.com/2016/02/25/internet-of-things-comment-ca-marche/>
- [02] <https://blog.lesjeudis.com/03/2019/10-applications-de-l-internet-des-objets-qui-revolutionnent-la-societe>
- [03] <https://the.businessofsocial.wordpress.com/04/2019/tag/web-5-0/>
- [04] <https://www.digitalcomer-wavestone.com/2016/04/salon-iot-world-decryptage-de-lecosysteme-de-liot-2/>
- [05] <https://www.quora.com/05/2019/What-is-IOT-architecture>
- [06] <https://translate.google.com/2018/translate?hl=fr&sl=en&u=https://www.rfpage.com/what-are-the-major-components-of-internet-of-things/&prev=search>
- [07] <https://www.newgenapps.com/2017/iot-ecosystem-components-the-complete-connectivity-layer>
- [08] <https://www.rs-online.com/designspark/eleven-internet-of-things-iot-protocols-you-need-to-know-about>
- [09] <https://www.digikey.fr/fr/articles/techzone/2017/oct/comparing-low-power-wireless-technologies>
- [10] <http://internetofthingsagenda.techtagret.com/2019/definition/IOT-security-internet-of-Things-security>
- [11] <https://www.peerbits.com/28/06/2019/difference-between-m2m-and-iot.html>
- [12] https://fr.internetofthing.org/Machine_to_machine
- [13] <https://linkedin.com/pulse/advantages-disavantages-internet-things-iot-tommy-quek>
- [14] <https://www.quora.com/what-are-the-advantages-and-disadvantages-of-iot>
- [15] <https://docplayer.fr/3571175-La-carte-arduino-1-qu-est-ce-que-la-carte-arduino.html>
- [16] https://www.gotronic.fr/art-carte-arduino-mega-2560-12421.htm#complete_desc
- [17] <http://www.electronicwings.com/13/06/2019/sensors-modules/esp8266-wifi-module>

- [18] <http://esp8266.net/>
- [19] <https://www.technologuepro.com/cours-capteurs-actionneurs-instrumentation-industrielle/ch12-les-differents-types-de-capteurs.pdf>
- [20] <https://www.engineersgarage.com/articles/sensors>
- [21] <http://eclats-antivols.fr/fr/alimentation-et-relais/19535-module-relais-arduino-5v-1-canal-220v-10a-domotique-pic-dsp-bras-avr-arm-mcu-sg092-sz-5410329400415.html>
- [22] <http://eskmmon.fr/arduino-202-int>
- [23] <https://projetsdiy.fr/programmer-esp8266-ide-arduino-librairies-gpio-web-serveur-client/>
- [24] <http://www.ihm3d-fr.com/capteur-ultrason-hc-Sr04.html>
- [25] http://microsann.com/images/Atelier_Robotique/Documentation/Fiche_capteur_HC.pdf
- [26] <http://blogtechniciens.com/comment-un-capteur-pir-fonctionne-comment-utiliser-avec-carte-arduino-filiere-informatique-systemes-information/>
- [27] <http://nagashur.com/blog/2015/08/capteur-pir-et-arduino-tutorial/>
- [28] <https://components101.com/08/2014/dht11-temperature-sensor>
- [29] <https://lastminuteengineers.com/dht11-dht22-arduino-tutorial/amp/>
- [30] http://veerobot.com/2018/Flame_sensor
- [31] <http://www.gotronic.fr/art-detecteur-de-flamme-st060-26123.html>
- [32] http://sensorkit.fr/joy-it/index.php?title=KY-026%D%A9tecteur_de_flamme
- [33] <http://www.supinfo.com/articles/single/296-qu-est-ce-qu-servomoteur>
- [34] <https://ardwinner.jimdo.com/arduino/v-les-servomoteurs/>
- [35] <https://lacavernedelucan.com/06/05/2019/comment-fonctionne-un-servo-moteur/>
- [36] <http://www.mdp.fr/2017/05/documentation/lexique/pas-a-pas/definition.html>
- [37] <http://a2itronic.ma/fr/home/441-moteur-28byj-48-5-pas-a-pas-unipolaire-module-uln2003.html>

Résumé

Depuis la fin des années 1980, Internet a évolué de manière spectaculaire. La dernière étape est l'utilisation de ce réseau mondial pour la communication avec des objets ou entre objets, cette évolution nommée Internet des Objets (IoT pour Internet of Things).

La domotique parmi les divers champs d'application de l'IoT, c'est un ensemble des techniques et des technologies permettant de superviser, d'automatiser, de programmer et de coordonner les tâches de confort, de sécurité, de maintenance et plus généralement de services dans l'habitat individuel ou collectif. Ce travail est consacré à la réalisation d'une maison intelligente à l'aide du module Arduino, qui est un microcontrôleur programmé peut analyser et produire des signaux électriques de manière à effectuer des tâches très diverses. En fait, nous avons réussi de réalisé plusieurs systèmes comme système d'éclairage, système d'alarme, automatisation des portails...ect.

Mots Clés : Maison intelligente, Arduino, Microcontrôleur, Domotique, Internet.

Abstract

Since the 1980s, the Internet has evolved dramatically. The last step is the use of this global network for communication with objects or between objects, this evolution called the Internet of Things (IoT).

Home automation among the various fields of application of the IoT is a set of techniques and technologies for supervising, automating, programming and coordinating comfort, safety, maintenance and more generally services in individual or collective housing. This work is devoted to the realization of a smart home using the Arduino module, which is a microcontroller programmed can analyze and produce electrical signals so as to perform very diverse tasks. we managed to realize several systems like lighting system, alarm system, automation of gates ... ect.

Keywords : Smart home, Arduino, Microcontroller, Home automation, Internet..