

**REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE**  
**MINISTERE DE L'ENSEIGNEMENT SUPERIEUR ET DE LA RECHERCHE SCIENTIFIQUE**  
**UNIVERSITE MOHAMED BOUDIAF - M'SILA**

**FACULTE : TECHNOLOGIE**

**DEPARTEMENT : ELECTRONIQUE**

**N° : 2018/ESEM/03/87**



**DOMAINE : SCIENCES ET TECHNOLOGIE**

**FILIERE : ELECTRONIQUE**

**OPTION : ELECTRONIQUE DES SYSTEMES  
EMBARQUES**

**Mémoire présenté pour l'obtention  
Du diplôme de Master Académique**

**Par: MEKHALFIA Toufik et GHADBANE Toufik**

**Intitulé**

**Etude et réalisation d'un système de commande  
à distance des installations électriques pour la  
domotique**

**Soutenu devant le jury composé de:**

Dr LAALAOUI Lahouaoui	Université : de M'sila	Président
Dr BOUCHAMA Idris	Université : de M'sila	Rapporteur
Mr KHENNOUF Salah	Université : de M'sila	Examineur

**Année universitaire : 2017/2018**

## *Remerciement*

*Nous remerciant ALLAH qui nous aide et nous donne  
la patience et le courage durant ces  
longues années d'étude.*

*Nous tiens à remercier grandement nos Encadreur*

***Dr: BOUCHAMA IDRIS***

*pour sa grande disponibilité et ses précieux conseils.*

*Nous remerciant également tous les enseignants du  
département d'électronique d'université de M'sila*

*plus spécialement les membres de jury de notre travail.*

*Enfin, nous adressons nos plus sincères remerciements à tous*

*nos amis et collègues qui nous ont toujours*

*soutenue et encouragée au cours de*

*la réalisation de ce mémoire.*

*Merci à tous.*

## *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 mes Grands Parents que Dieu les protège et à toute  
ma famille*

*A mes chers frères et mes chères sœurs*

*A mes chers amis*

*Ainsi que la promotion Master ESEM sans oublier tous  
mes enseignants durant tout mon cursus*

*Merci à tous.*

*MEKHALFIA Toufik*

## *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 mes Grands Parents que Dieu les protège et à toute  
ma famille*

*A mes chers frères et mes chères sœurs*

*A mes chers amis*

*Ainsi que la promotion Master ESEM sans oublier tous  
mes enseignants durant tout mon cursus*

*Merci à tous.*

*GHADBANE Toufik*

---

## Sommaire

Introduction générale	1
<b>Chapitre I: Domotique et maison intelligente</b>	
I.1. Introduction	3
I.2. Définitions de la domotique	3
I.3. Historique	4
I.4. fonctionnement de la domotique	5
I.5. Différents domaines d'application de la domotique	6
I.5.1. Domotique pour le confort	6
I.5.2. Domotique pour l'énergie	7
I.5.3. Domotique pour la sécurité	7
I.5.4. Domotique pour la santé	9
I.6. Recherche de la domotique	9
I.7. Maison intelligente	10
I.8. L'internet des objets (IOT)	11
I.8.1. Objet connecté	11
I.9. Maison communicante	12
I.9.1. Domotique sans fil	13
I.9.2. Domotique à courant porteur CPL	14
I.9.3. Domotique câblée	15
I.10. Maison intelligente et la domotique	15
I.11. Avantages et inconvénients d'une maison intelligente	17
I.11. 1. Avantages	17
I.11. 2. Inconvénients	18
I.12. Conclusion	18

---

## Chapitre II: Arduino et Shields

II.1. Introduction	19
II.2. Domaine d'utilisation et ces applications	19
II.3. Avantages d'Arduino	20
II.4. Présentation de la carte	20
II.4.1. Qu'est-ce qu'un microcontrôleur	20
II.4.2. Microcontrôleur ATMEL ATmega328	21
II.4.3. Caractéristiques techniques de l'Arduino UNO	22
II.4.4. Les différents éléments de la carte de commande	24
II.5. Capteurs	24
II.5.1. Présentation des capteurs	25
II.5.2. Communication	28
II.5.2.1. Module Arduino Bluetooth	28
II.5.2.2. Module shield Arduino Wifi	28
II.6. Présentation du logiciel	29
II.6.1. IDE Arduino	29
II.6.1. Les boutons	30
II.6.2. Langage Arduino	30
II.6.3. Fonctionnalité de base	33
II.6.3.1. Les entrées/sorties	33
II.6.3.2. Gestion du temps	35
II.6.3.3. Les interruptions	35
II.6.3.4. Quelques bibliothèques	38
II.7. Conclusion	39

---

## Chapitre III: Conception et réalisation du système

III.1. Introduction	4
III.2. Schéma synoptique du système à réaliser	40
III.3. Schéma électrique du montage sur Fritzing	41
III.4. Description des shields utilisés	42
III.4. 1.Capteur de Température et d'Humidité (DHT22)	43
III.4.1.1. Description du capteur	43
III.4.1.2. Principales caractéristiques de DHT22	43
III.4.2. Capteur de gaz (MQ2)	44
III.4.2.1. Description du capteur	44
III.4.2.2. Principales caractéristiques de gaz/fumée MQ-2	44
III.4.3. Capteur de mouvement (PIR)	45
III.4.3.1. Description du capteur	45
III.4.3.2. Principales caractéristiques de mouvement PIR	45
III.4.4. Module GSM SIM800L	46
III.4.4. Principales caractéristiques module GSM SIM800L	46
III.4.5. Module de Relais à 4 canaux	46
III.4.6. Servomoteurs	47
III.4.6. Fonctionnement de servomoteurs	48
III.4.7. Ventilateur	49
III.4.8. Description du module de connexion Wifi ESP8266	49
III.4.8.1. Mise sous tension	50
III.4.8.2. Principales caractéristiques d'ESP8266	50
III.5. Réalisation de la maison intelligente	51
III.5.1. Développement d'une application Androïde	52
III.5.1.1. le système Androïde	52
III.5.1.2. Application Android dans les Smartphones	53

III.5.2. Structure d'une application App Inventor	53
III.6.Explication et démarche	57
III.6.1.Présentation des fonctions de système	58
III.6.1.1.Fonction de point d'accès (Passerelle)	58
III.6.1.2.Fonction de la sécurité	58
III.6.1.3.Fonction d'ouverture du garage et la porte principale	58
III.6.1.4.Fonction d'éclairage	58
III.6.1.5.Fonction de l'acquisition de la température et d'humidité	58
III.6.1.6.Fonction de ventilation	58
III.6.1.7.Fonction de détection de gaz/fumée	59
III.6.2.Présentation de l'interfaçage Arduino-PC	59
III.7.Conclusion	60
Conclusion générale	61
<b>Bibliographie</b>	63

## Liste des figures

Figure I.1 : Illustre du confort	7
Figure I.2 : Détecteur de fumée	8
Figure I.3: Alarme piscine	8
Figure I.4 : Alarmes anti-intrusion	8
Figure I.5 : Schéma représentatif de fonctionnement général des équipements d'un Maison intelligente	10
Figure I.6 : La lampe DAL (premier objet connecté)	11
Figure I.7 : Thermostat Qivivo	12
Figure I.8 : Le protocole radio Zwave	13
Figure I.9 : Le HomeEasy	13
Figure I.10: Le X2D (courant porteur)	13
Figure I.11: L'io-Home	14
Figure I.12: Le réseau Zigbee	14
Figure I.13: courant porteur CPL	14
Figure I.14 : Kit Alarme sans fil Vidéo Domonial Standard HONEYWELL	16
Figure I.15 : Aspirateur robot Wi-Fi Roomba 695 d'iRobot	16
Figure I.16 : Led ampoule sans fil	17
Figure I.17 : Moniteur d'énergie connecté	17
Figure II.1: Datasheet ATMega328	21
Figure II.2: La carte Arduino UNO	23
Figure II.3: Schéma simplifié de la carte Arduino UNO	23
Figure II.4 : Type de modules Bluetooth	28
Figure II.5: Module shield wifi	28
Figure II.6 : IDE Arduino	29
Figure II.7 : Présentation des boutons	30
Figure II.8 : Un code minimal	30
Figure II.9.: Signal PWM	35
Figure III.1: Le schéma synoptique de système à réaliser	40
Figure III.2: Schéma électrique du montage	41
Figure III.3: Montage en plaque d'essai	42
Figure III.4: Image réelle du capteur de Température et d'Humidité DHT22	43
Figure III.5: Image réelle du capteur de gaz/fumée MQ-2	44

Figure III.6: Image réelle du capteur de mouvement PIR	45
Figure III.7: Image réelle le module GSM SIM800L	46
Figure III.8 : Module de relais à 4 canaux	47
Figure III.9 : Ventilateur 5V	49
Figure III.10: Image réelle du module Wifi ESP8266 – ESP-D1	49
Figure III.11 : Photos d'extérieur de maquette de maison	51
Figure III.12: Photos d'intérieur de maquette de maison	51
Figure III.13: Photos de la réalisation finale de maquette de maison	52
Figure III.14 : Première interface de la création App Inventor	54
Figure III.15 : Schéma global du Screen 1	54
Figure III.16 : En-tête d'éditeur de blocks App Inventor	55
Figure III. 17 : Editeurs de blocks App Inventor	56
Figure III.18 : Echantillon d'un composant sous App Inventor	56
Figure III.19 : Schéma global du Scratch	57
Figure III.20 : Photo réelle de notre application	57
Figure III.21: Image réel de l'interface de l'application de commande	59
Figure III.22: Fenêtre d'affichage des résultats de système sur le moniteur sérié	60

## **Liste des tableaux**

Tableau II.1: Différents éléments de la carte de commande	24
Tableau II.2: Représente différents shields qui connectée avec arduino	25
Tableau II.3: Les types de base	31
Tableau II.4: Les structures de contrôle	32
Tableau III.1: Servomoteurs utilisés dans noter projet.	48

## **Introduction générale**

Aujourd'hui, l'électronique est de plus en plus remplacée par l'électronique programmée. On parle aussi de système embarquée ou d'informatique embarquée. Son but est de simplifier les schémas électroniques et par conséquent réduire l'utilisation des composants électroniques, réduisant ainsi le coût de fabrication d'un produit. Il en résulte des systèmes plus complexes et performants pour un espace réduit. [1]

L'évolution de la technologie et du mode de vie nous permet aujourd'hui de prévoir des espaces de travail et de logement mieux adaptés. De même, La majorité des individus, et plus particulièrement les personnes âgées, passent beaucoup de leur temps à domicile, d'où l'influence considérable de l'habitat sur la qualité de vie. L'amélioration du sentiment de sécurité et de confort dans l'habitat apparaît donc comme une tâche d'une grande importance sociale. En effet, 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. [2]

D'autre part, la forte augmentation des ventes de smart phone et de tablettes électronique se fait en même temps qu'une adoption rapide par le grand public des technologies de la domotique ainsi que l'autopilotage. Au fond, le smart phone devient une télécommande universelle pour toute la maison et les équipements électriques. Les utilisateurs pourront à terme contrôler à distance un très grand nombre de fon actions sans avoir à tenir compte de la marque ou de l'origine du produit qu'ils pilotent.

Dans ce travail nous essayons de réaliser un système de commande à distance (sous réseau Internet) à base d'un Arduino UNO pour commander des installations électriques pour la domotique (une maison intelligente) comprenant plusieurs capteurs et actionneurs tels que: capteur de Température, Humidité, Capteur de Gaz/Fumée, Servomoteurs, relais ... etc.

La carte d'acquisition à base d'Arduino UNO est pour but de transférer les données de ces capteurs vers un smart phone et commander des actionneurs aussi, Les cartes Arduino sont conçues pour réaliser des prototypes et des maquettes de cartes électroniques pour l'informatique embarquée.

Deux principaux objectifs sont visés: Le premier objectif est de regrouper suffisamment d'informations sur une grande catégorie de cartes d'interfaçage à base de l'Arduino: son langage de programmation, sa construction, son principe de fonctionnement.

Le deuxième objectif consiste à réaliser une connexion sans fil entre la carte Arduino et smart phone à travers la carte WIFI série sans fil.

Notre projet expérimental est donc consiste alors à concevoir et à réaliser un système de maison connectée contrôlable via une application mobile multiplateforme en local ou par internet. Les données sont traitées dans l'unité de traitement et de commande à base de l'Arduino UNO afin de les transférer par un protocole de transfère des données sans fils WLAN (Wireless Local Area Network) ou LAN (Local Area Network) à travers de la carte WIFI série sans fil de type ESP8266 (ESP-D1) au réseau internet.

Ce mémoire est organisé en trois chapitres, organisés comme suit:

- Dans un premiers temps on commence par une introduction générale.
- Dans le premier chapitre, nous allons voir une présentation générale de la domotique ainsi ses secteurs d'application et les différents types de technologies utilisées.
- Le deuxième chapitre est consacré à la présentation du système Arduino UNO, ses caractéristiques et présentation des quelques shields et on mettra la lumière sur le logiciel IDE Arduino.
- Le troisième chapitre est réservé pour l'étude et la réalisation de notre système; nous allons donner une description détaillée des shields utilisés et le mode de fonctionnement de ces shields dans notre système de commande, ensuite on va réaliser une petite application App Inventor, qui va nous permettre de connecter notre système par wifi. On va présenter à la fin de ce chapitre les résultats obtenu après la réalisation de ce système de commande à distance des installations électriques.
- Enfin, on termine par une conclusion générale.

## **I.1. Introduction**

Durant ces dernières années la technologie de fabrication des cartes de commande connue une évolution remarquable, cette révolution prodigieuse de la microélectronique a conduit à la fabrication des systèmes de commande de plus en plus complexes, offrant des avantages meilleurs tels que : la simplicité de la programmation, la vitesse d'exécution, les ports d'entrée/sortie... etc. [3]

Avec le développement des équipements électriques du logement, un nombre de plus en plus grand des systèmes électriques permettent de piloter de façon simple et confortable l'ensemble de ces équipements notamment l'éclairage, le climatiseur, les ouvrants, l'arrosage et le système alarme. On appelle ces systèmes « la domotique ».

Les progrès technologiques, notamment de l'informatique, de la télécommunication et de l'électronique ont permis le développement de systèmes de transmission, des commandes à distance et favorisé l'éclosion d'une offre abondante de nouveaux services pour les occupants des logements.

Dans ce chapitre, nous allons voir une présentation générale de la domotique ainsi ses secteurs d'application et les différents types des technologies utilisées.

## **I.2.Définitions de la domotique**

Le mot domotique vient de domus qui signifie «domicile » et du suffixe –tique qui fait référence à la technique. la domotique est l'ensemble des techniques de l'électronique, de physique du bâtiment, d'automatisme, de l'informatique et des télécommunications utilisées dans le bâtiment, plus ou moins « interopérables » et permettant de centraliser le contrôle des différents systèmes et sous-systèmes de la maison et de l'entreprise (chauffage, volets roulants, porte de garage, portail d'entrée, prise électrique, etc.). la domotique vise à apporter des solutions techniques pour répondre aux besoins de confort (gestion d'énergie, optimisation de l'éclairage et du chauffage), de sécurité (alarme) et de communication (commandes à distance, signaux visuels ou sonores, etc.) que l'on peut trouver dans les maisons, les hôtels, les lieux publics...etc.[4]

A l'origine, la domotique avait donc pour but d'automatiser sa maison : ouverture et fermeture automatiques des volets, ouverture du portail électrique, gestion du

chauffage, gestion de l'éclairage, etc. Ainsi avant l'ère des Smartphones, il était par exemple possible d'activer son chauffage à distance en passant un coup de téléphone à sa maison, ou encore en lui envoyant un SMS. C'était tout à fait réalisable. Seulement une telle installation était relativement compliquée à mettre en place et, il faut bien l'avouer, coûteuse. Cette époque a malheureusement laissé des traces, puisque pour beaucoup encore aujourd'hui, domotique rime avec cher et compliqué. Pourtant, ce domaine a énormément évolué et il existe de nombreuses solutions simples à mettre en place et tout à fait abordables pour le grand public. [4]

La domotique a surtout elle-même évolué, si bien que le terme est quelque peu dépassé. La domotique servait à automatiser sa maison ; aujourd'hui on parle de domotique 2.0, ou de « maison intelligente », pour bien marquer l'évolution de ce monde. Les différents domaines de la maison ne se contentent plus d'être automatisés et pilotables, ils communiquent ensemble, permettant à la maison de réagir selon différents évènements. [4]

### **I.3. Historique**

Les premiers travaux de domotique sont apparus dans les années 70 avec les problématiques énergétiques dues aux crises pétrolières. Ces crises marquent le début du développement de l'électronique pour les bâtiments. Au départ, la domotique contrôle seulement les prises, l'éclairage et les volets roulants grâce à une télécommande. Au fur et mesure, de nouveaux objets se mettent en réseau comme les thermostats et les alarmes.

Mais c'est véritablement à partir de la fin du 20e siècle, que la domotique va se démocratiser.

Deux raisons expliquent ce développement :

- l'arrivée de l'ordinateur et des technologies de communication dans la maison au début des années 1990; notamment, le déploiement d'Internet qui permet aux ordinateurs de communiquer entre eux.
- Le coût de l'énergie qui augmente suite aux deux crises pétrolières survenues dans les années 70. Désormais, de nouvelles normes forcent les constructeurs privilégier des bâtiments bien mieux isolés pour limiter leur utilisation chauffage.

La domotique intervient donc avec des appareils capables de communiquer entre eux pour surveiller et gérer cette énergie.

Depuis les années 2000, avec le développement des technologies sans fil comme le wifi ou le Bluetooth, la miniaturisation des composants électroniques, l'avènement des appareils mobiles, l'invasion des écrans tactiles et des télévisions connectées, les ingénieurs peuvent désormais proposer au public des produits - objets connectés ou systèmes domotiques – bien plus puissants et simples d'utilisation. [2]

#### **I.4.Fonctionnement de la domotique**

Aujourd'hui, les différents objets connectés de la maison ne se contentent plus d'être automatisés et pilotables ; ils interagissent ensemble pour notamment offrir aux habitants des maisons intelligentes un véritable confort d'usage, gagner en sécurité et optimiser la consommation énergétique des bâtiments. En communiquant avec l'habitat, il est possible de régler le chauffage par zones, de simuler à distance une présence, etc. En couplant l'installation avec une télécommande universelle ou avec un simple appui sur une touche sur son Smart phone, le pilotage s'effectue de n'importe où dans la maison, en fonction des besoins.

Concrètement, la domotique consiste à mettre en réseau différents appareils connectés dans une maison et à centraliser les commandes. Ces appareils sont déjà souvent existants : radiateurs, ventilation, éclairage, ... auxquels on ajoute des moyens de communiquer au sein de la maison. Chaque appareil est connecté avec d'autres via un appairage, qui consiste à associer deux ou plusieurs appareils entre eux. Cet appairage permet par exemple de dire à un interrupteur quel groupe de lampes il va devoir allumer. L'appairage peut se faire directement entre deux objets, ou via un boîtier domotique qui sert d'intermédiaire.

Chaque groupe d'appareils (éclairage, chauffage, volets roulants, ...) est pilotable via une ou plusieurs applications sur des appareils tels que les Smartphones, tablette, ordinateur ou télécommande. C'est cette application qui, à distance, permet de transmettre une demande (augmentation de la température, éclairage d'une pièce, démarrage de la télévision).

Les objets de l'habitat sont ainsi considérés comme intelligents. Ils sont équipés de capteurs tels que des capteurs de température et de présence pour un thermostat, qui

vont mesurer et détecter les habitudes des personnes vivant dans la maison. Les informations telles que les arrivées, sorties, temps passé dans une pièce des habitants sont toutes enregistrées et envoyées aux radiateurs pour faire adapter la température en fonction des scénarii programmés. Ainsi, on ne chauffe que quand c'est nécessaire.

Les objets peuvent communiquer entre eux par plusieurs moyens, dont les trois plus fréquents sont :

- L'envoi d'information par **un réseau filaire**, tel qu'un réseau informatique, un réseau téléphonique ou un câble dédié (un bus de données)
- Les informations peuvent aussi passer par **des câbles électriques**, ce qu'on appelle le courant porteur
- Ou alors le boîtier peut émettre **des ondes** comme le wifi, le Bluetooth ou les ondes radio.

Le câblage reste, à ce jour, la solution la plus fiable. Mais la domotique sans fil est plus simple à installer. On choisira l'un ou l'autre de ces moyens de communication en fonction des caractéristiques de l'habitat (ancien ou neuf).

Enfin, il est possible d'utiliser plusieurs types de communication en utilisant un boîtier domotique pour transmettre les informations d'un réseau à un autre. [5]

## **I.5. Différents domaines d'application de la domotique**

### **I.5.1. Domotique pour le confort**

Gestion de l'éclairage, gestion du chauffage, gestion des volets roulants, par simple action d'une commande, toutes ces tâches sont simplifiées grâce à la domotique. La domotique permet d'améliorer le confort d'usage. Grâce à une application installée sur son Smartphone, par exemple, les habitants d'une maison connectée peuvent décider de l'heure d'ouverture des volets, de la température des pièces selon l'heure de la journée. Des capteurs installés un peu partout dans la maison détectent la présence des individus et peuvent ainsi donner le signal pour allumer ou éteindre les lumières dans une pièce, activer la température optimale et même aller jusqu'à démarrer une musique d'ambiance dans le salon si les habitants l'ont choisie. [5]



**Figure I.1** : illustre du confort

### **I.5.2.Domotique pour l'énergie**

L'un des enjeux de la domotique est d'améliorer significativement l'efficacité énergétique des bâtiments. Les maisons dites « intelligentes » ou connectées sont équipées d'un ensemble de technologies innovantes permettant d'améliorer de manière globale leurs performances énergétiques sans perte de confort.

Parmi ces technologies, de nombreux automatismes : gestion des volets, de la ventilation, gestion des équipements de chauffage rendent les maisons réactives aux conditions extérieures (climat) et intérieures (usage), l'objectif final étant de réduire les dépenses quotidiennes d'énergie tout en préservant le confort des habitants. [5]

### **I.5.3.Domotique pour la sécurité**

Un des domaines d'application de la domotique est la sécurité des biens et des personnes par des systèmes d'alarme qui préviennent d'une part des risques techniques (pannes ou dysfonctionnements des appareils) et d'autre part des éventuelles intrusions dans la maison (cambriolage). En général on trouve :

**Alarmes techniques :** Les alarmes techniques sont basées sur des capteurs capables de détecter différents incidents tels que des dégagements toxiques, incendie, fuite d'eau, fuite de gaz, etc. Ces différents capteurs sont raccordés à une centrale d'alarme. Les sécurités anti-noyade des piscines font également partie de ces systèmes d'alarme.

Ainsi que certains détecteurs de pannes sur les équipements domestiques (chaudière par exemple).



**Figure I.2 :** détecteur de fumée



**Figure I.3:** alarme piscine

**Alarmes anti-intrusion :** Ce sont en général des capteurs sur les portes (détection d'ouverture) ou dans les pièces détection de présence) qui sont reliés eux aussi à une centrale d'alarme. Ces capteurs peuvent être couplés avec un réseau de caméras numériques de surveillance. Lors d'une intrusion, un message d'alerte peut être envoyé par e-mail ou sur un téléphone portable. [6]



**Figure I.4 :** alarmes anti-intrusion

### **I.5.4.Domotique pour la santé**

La domotique trouve aujourd'hui de nouvelles applications dans le domaine de la santé. En installant des systèmes domotiques dans les maisons des personnes en situation de handicap, atteintes de maladies neurodégénératives telles que la maladie d'Alzheimer ou encore des personnes âgées, il est possible de les aider dans leur quotidien en automatisant le plus possible des tâches considérées comme complexes.

Cela permet également à la personne de rester à son domicile plus longtemps et d'être suivie à distance. Par exemple, grâce à la domotique, on peut détecter quand une personne ne boit pas assez d'eau ou quand elle oublie de se nourrir. Si le comportement est considéré comme « préoccupant », il est alors possible d'alerter la famille ou les secours selon les scénarii programmés dans l'interface de commande. [5]

### **I.6.Recherche de la domotique**

Depuis quelques années, avec la démocratisation des Smartphones, des tablettes et des objets communicants, les maisons connectées se sont développées à des coûts plus raisonnables que par le passé. Les recherches menées actuellement dans le domaine de la domotique consistent à proposer aux consommateurs des solutions domotiques meilleur marché, plus facile à installer et à développer des applications et interfaces de gestion plus ergonomiques et simples à utiliser, adaptées aux particularités des marchés locaux.

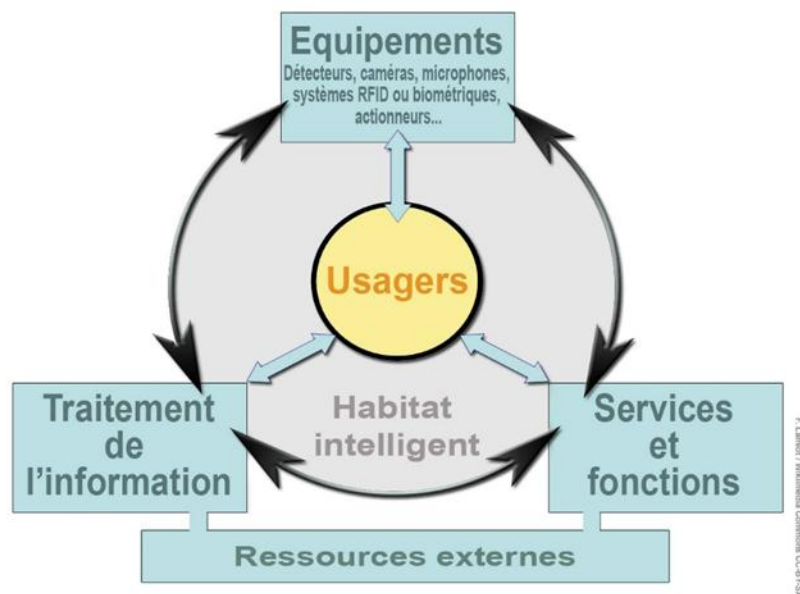
La gestion de l'énergie est un enjeu historique de la domotique. Profitant de l'essor des énergies renouvelables telles que le solaire et l'éolien, les technologies domotiques permettent une maîtrise des consommations tout en respectant le confort des usagers. L'intégration de ces énergies dépasse le cadre de la maison et s'étend également au domaine des transports. Par exemple, aujourd'hui, en cas de panne électrique, les véhicules électriques branchés peuvent prendre le relais et alimenter la maison.

Par ailleurs, de nombreuses recherches en domotique portent sur l'axe « santé ». En lien étroit avec les usagers (personnes âgées, personnes en situation de handicap) et les fournisseurs de produits et services, les ingénieurs et chercheurs développent des solutions fiables et sécurisées pour renforcer l'autonomie de ces personnes : domotique innovante, systèmes de surveillance et d'alerte adaptés (capteurs de chutes), aides

techniques connectées (robotique, automatique), solutions de gestion énergétique efficaces, solutions pour la mobilité et le lien social. L'objectif est de permettre à ces individus de rester le plus longtemps possible à domicile en toute sécurité. [5]

## I.7.Maison intelligente

Pour faire simple, une maison intelligente est une maison dans laquelle plusieurs objets et appareils sont connectés à votre Smartphone. Du thermostat à l'éclairage, en passant par le système d'alarme ou le réfrigérateur, tous ces appareils intelligents (« smart devices ») communiquent entre eux par le biais d'une connexion internet sans fil. Une maison intelligente est une maison partiellement ou totalement automatisée. Dans une maison intelligente, plusieurs petites applications peuvent être connectées. La transformation de votre habitation en maison intelligente peut donc se faire progressivement. Le grand intérêt d'une maison connectée est que la communication se fait de manière bilatérale. À l'aide de votre Smartphone, vous gérez vos appareils à distance et ces derniers vous envoient des informations pratiques. [7]



**Figure I.5 :** schéma représentatif de fonctionnement général des équipements d'un maison intelligente

## I.8. L'internet des objets (IOT)

IOT est l'acronyme de « Internet Of Things », ou internet des objets en français. IOT est l'extension d'internet qui n'était qu'un monde virtuel et qui n'interagissait pas ou peu avec le monde physique à des entités et des emplacements existants sur terre. Les données générées par ces entités (objets) sont échangées via internet afin d'être exploitées dans divers domaines ; la santé, la domotique, l'agriculture...etc. Les géants de l'informatique parlent de maisons connectées, de villes intelligentes, et de véhicules autonomes. [8]

### I.8.1. Objet connecté

Un objet connecté est un objet électronique relié à internet et capable de communiquer des informations, apportant ainsi un service ou une valeur ajoutée. Le premier objet connecté était la lampe DAL (figure I.6), lancée en 2003 par RAFI Haladjan. Sensible au toucher et au bruit, cette lampe communiquait des informations sur la météo, la bourse, la pollution, des alertes Google et même des messages grâce à neuf LED de couleur. Les fonctions proposées aujourd'hui vont beaucoup plus loin que la simple annonce de la météo. Prenons l'exemple du thermostat Qivivo (figure I.7), qui permet non seulement de piloter son chauffage à distance, mais également d'obtenir un diagnostic de sa consommation d'énergie, des conseils d'optimisation et même d'être mis en relation avec des professionnels afin d'étudier les possibilités d'amélioration de son logement.

Les usages ont donc pu se développer et, aujourd'hui, les objets connectés sont partout. Ils sont particulièrement appropriés dans certains domaines. [8]



**Figure I.6 :** La lampe DAL (premier objet connecté).



**Figure I.7 :** Thermostat Qivivo.

## **I.9.Maison communicante**

L'homme avait imaginé qu'il est impossible de se communiquer avec son habitat, le contrôler à distance. Maintenant, la communication tient une place de plus en plus importante dans le logement. Une installation domotique adaptée, avec les appareils de la maison montés en réseau, satisfait aux besoins et aux loisirs de chaque personne du foyer.

La centralisation des commandes est le corps du système domotique. Les appareils mis en réseau se reconnaissent et dialoguent entre eux, se déclenchant par simple appui sur une touche. Par le biais d'un interrupteur centralisé, les éclairages et volets motorisés peuvent être actionnés individuellement, par groupes de pièces ou simultanément.

En communiquant avec l'habitat, il est possible de régler le chauffage par zones, de simuler à distance une présence, etc. En couplant l'installation avec une télécommande universelle ou avec un simple appui sur une touche sur son Smart phone, le pilotage s'effectue de n'importe où dans la maison, en fonction des besoins.

Afin d'adapter la domotique à chaque logement et utilisation, plusieurs configurations sont à disposition :

- En domotique sans fil (Wifi, ondes radio,...).
- Par domotique CPL ou à courant porteur (appelé X10).
- Avec un câblage domotique bien pensé. [2]

### I.9.1. Domotique sans fil

La domotique sans fil utilise plusieurs supports technologiques : les ondes radio ou RF (sur des fréquences en MHz) et l'infrarouge ou IR, qui a pour inconvénient de ne pas traverser les murs. Il est conseillé, pour une meilleure stabilité du système, de ne pas mixer le sans fil avec un autre type de technologie, le CPL par exemple. Cela peut nuire à l'installation et à la qualité de la communication entre les équipements. [2]

Les ondes radio sont employées par de multiples protocoles comme le X10 RF, le HomeEasy, le X2D, le Zigbee, le Zwave, ou encore le Bluetooth.

Les principales fréquences utilisées dans la domotique sont le 433 MHz et le 868MHz. On trouve parmi les protocoles sans fil :

- Le protocole radio Zwave, fréquence 868,42 MHz en Europe, répercute un ordre reçu vers les modules voisins. La portée du contrôleur Zwave peut équiper toute la maison sans risquer de problèmes de transmission.



**Figure I.8 :** Le protocole radio Zwave.

- Le HomeEasy, lui, utilise la fréquence 433 MHz qui est règlementée par l'UIT (Union internationale des télécommunications).



**Figure I.9 :** Le HomeEasy.

- Le X2D est mixte (courant porteur ou radio 868 MHz) convient à la domotique de sécurité et la domotique du chauffage.



**Figure I.10:** Le X2D (courant porteur).

- L'io-Home Control utilise les fréquences allant de 868 MHz à 870MHz, il possède un véritable retour d'informations grâce à son protocole bidirectionnel. Cette technologie est ouverte à différents fabricants leaders dans l'habitat.



**Figure I.11:** L'io-Home.

- Le réseau Zigbee, basé sur le standard 802.15.4, ratifié par l'IEEE (Institute of Electrical and Electronics Engineers), a de plus en plus de fidèles. Il fonctionne avec des piles très longues durées d'autonomie, sur 866 MHz (bande libre en Europe) et 915 MHz (aux États-Unis).



**Figure I.12:** Le réseau Zigbee.

### **I.9.2. Domotique à courant porteur CPL**

L'utilisation de la domotique à courant porteur revient à transformer son habitat en maison communicante par le biais d'une installation domotique; c'est-à-dire que l'on utilise le réseau électrique déjà existant.

La domotique CPL est aussi connue sous l'acronyme de X10, qui est un protocole de communication et de contrôle de plusieurs appareils domotiques.



**Figure I.13:** courant porteur CPL

Les CPL, c'est la possibilité de faire passer de l'information numérique (voix, donnée, image) sur le réseau électrique ordinaire. Ils s'avèrent très utiles en cas de rénovation.

L'intérêt de cette technologie porte sur l'utilisation d'un réseau filaire structuré déjà existant et parfaitement distribué dans toute la maison ou le bâtiment : le réseau électrique et ses points d'accès constitués par les prises électriques.

Toutefois, la fiabilité de la domotique CPL est contestable. Cette technologie peut parasiter le réseau et perturber les autres transmissions. De plus, cet équipement est encore coûteux. À performances équivalentes, il est en effet plus cher que le sans fil. Enfin, ce système est aussi moins rapide, et il n'a pas de mobilité par construction.

### I.9.3. Domotique câblée

Certains professionnels ne sont pas favorables, au sein d'une installation domotique, aux approches sans fil ou CPL. Ils leur préfèrent une domotique par câbles. Le pré-câblage doit être souple et évolutif, car la technologie ne cesse d'évoluer.

Il faut ainsi prévoir un local technique, le «local de répartition», qui centralise les points d'arrivée de toutes les liaisons externes (électricité, téléphone, Internet, télévision, fibre optique ...).

Dans les logements, le Bus de terrain KNX est une excellente solution domotique. Ce Bus est constitué d'un câble fait de conducteurs torsadés par paires (deux au minimum) alimenté en très basse tension (courant faible).



**Figure I.13:** Le Bus de terrain KNX

Le réseau a pour but d'empêcher les interférences électriques reprochées au CPL. Cependant, tout repose sur la qualité des câbles choisis.

Trois types de câbles sont fréquemment rencontrés, le câble UTP, le câble STP et le câble FTP. Les meilleurs câbles sont blindés ou écrantés, de type STP ou FTP.

Il est recommandable de choisir un réseau électrique, car c'est le plus simple à installer (et le mieux connu par les artisans et les architectes). Il doit respecter toute fois la norme NFC15-100. Il est aussi préférable d'installer un panneau de brassage équipé de prises RJ45.

Ensuite, il faut prévoir un onduleur pour les équipements du réseau (modem ADSL, routeur, switches) et les équipements de la domotique de sécurité. [5]

## I.10.Maison intelligente et la domotique

Bien que foncièrement liés, les concepts de maison intelligente et de domotique présentent des différences notoires. Un système domotique constitue un ensemble intégré reprenant tous les composants de votre habitation. Un tel système exige une installation spécifique effectuée par un installateur qualifié. C'est là que se situe la principale différence avec les nouvelles applications de la maison intelligente. Celles-ci sont faciles à utiliser et ne nécessitent aucune installation sophistiquée. Il suffit de posséder un Smartphone, une connexion internet et, bien entendu, une... maison !

Aujourd'hui, tous les appareils et tous les éléments d'habitation sont proposés en version connectée. Exemples : [5]

- **Sécurité et contrôle** : systèmes d'alarme intelligents, détecteurs de mouvements et caméras connectés, volets roulants commandés à partir de votre Smartphone, détecteurs de fumée et d'incendie qui envoient un message d'alerte sur votre Smartphone...



**Figure I.14** : Kit Alarme sans fil Vidéo Domonial Standard HONEYWELL

**Appareils électroménagers** : aspirateurs-robots, lave-linge, réfrigérateurs intelligents...



**Figure I.15** : Aspirateur robot Wi-Fi Roomba 695 d'iRobot

- **Éclairage** : lampes intelligentes qui, via votre Smartphone, s'allument et s'éteignent automatiquement lorsque vous pénétrez dans votre habitation ou que vous la quittez.



Figure I.16: Led ampoule sans fil

- **Énergie** : des moniteurs d'énergie permettant de contrôler à tout moment et à distance votre consommation énergétique, ainsi que celle de vos appareils électroménagers



Figure I.17 : moniteur d'énergie connecté

## I.11. Avantages et inconvénients de la domotique

### I.11. 1. Avantages :

- Le principal avantage de la domotique est l'amélioration du quotidien au sein de la maison, du point de vue du confort, de la sécurité et de la gestion de l'énergie.
- Ce type d'équipement vous simplifie la vie et optimise votre confort en adaptant votre maison à différents scénarios de la vie quotidienne.
- Il vous permet notamment d'éteindre tous vos appareils électriques et de mettre l'alarme quand vous quittez votre domicile, de régler des ambiances lumineuses

(ambiance lecture, ambiance relaxation avec lumières tamisées), de vous réveiller dans un habitat chauffé où le café est déjà prêt, d'enclencher automatiquement l'arrosage ou l'ouverture des volets chaque matin.

- La domotique permet aussi de réaliser des économies d'énergie grâce à la gestion automatique du chauffage, de la climatisation et de l'éclairage et à la programmation des appareils électroménagers en heures creuses.
- Elle a pour avantage d'améliorer la sécurité grâce à des alarmes, des systèmes d'ouverture automatique de la porte (reconnaissance vocale, carte magnétique...)
- En cas de tentative d'intrusion dans la maison, un appel téléphonique automatique peut contacter le propriétaire ou une entreprise de sécurité.
- Enfin, ces différentes technologies constituent une aide précieuse pour les personnes dépendantes et handicapées. [6]

### **I.11. 2. Inconvénients**

Le principal inconvénient est le prix d'achat et d'installation. Le prix est beaucoup plus élevé mais vos factures d'énergie baisseront. Il faut donc le prendre en compte dans le budget initial. Le deuxième inconvénient est le verrouillage qu'offrent certaines marques dans leurs produits ne permettant pas d'avoir un logiciel ouvert. [6]

### **I.12. Conclusion**

Dans ce chapitre nous avons commencé par une présentation générale de la domotique ainsi ses secteurs d'application et les différents types de technologies utilisées. Ensuite on a mis la lumière sur la maison intelligente et ses avantages/inconvénients.

Dans le chapitre qui suit on va mettre le point sur la présentation de la carte Arduino UNO, ses caractéristiques et présentation de quelques shields.

## II.1. Introduction

Le système Arduino donne la possibilité d'allier les performances de la programmation à celles de l'électronique. Plus précisément, pour programmer des systèmes électroniques. Le gros avantage de l'électronique programmée c'est qu'elle simplifie grandement les schémas électroniques et par conséquent, le coût de la réalisation, mais aussi la charge de travail à la conception d'une carte électronique

Arduino est un circuit imprimé en matériel libre sur lequel se trouve un microcontrôleur qui peut être programmé pour analyser et produire des signaux électriques, de manière à effectuer des tâches très diverses comme la domotique (le contrôle des appareils domestiques - éclairage, chauffage...), le pilotage d'un robot, etc. [1]

Le langage Arduino se distingue des langages utilisés dans l'industrie de l'informatique embarquée de par sa simplicité. En effet, beaucoup de bibliothèques et de fonctionnalités de base occultent certains aspects de la programmation de logiciel embarquée afin de gagner en simplicité. Cela en fait un langage parfait pour réaliser des prototypes ou des petites applications.

Les possibilités des cartes Arduino sont énormes, un grand nombre d'applications ont déjà été réalisées et testées par bon nombre d'internautes. [9]

Dans ce chapitre nous allons voir une présentation générale du système Arduino UNO, ses caractéristiques et présentation des quelques shields et on mettra la lumière sur le logiciel IDE Arduino.

## II.2. Domaine d'utilisation et ces applications

Le système Arduino nous permet de réaliser un grand nombre de choses, qui ont une application dans tous les domaines, l'étendue de l'utilisation de l'Arduino est gigantesque. Pour vous donner quelques exemples, vous pouvez :

- Électronique industrielle et embarquée.
- Contrôler les appareils domestiques.
- Fabriquer votre propre robot.
- Faire un jeu de lumières.
- Communiquer avec l'ordinateur.

- Télécommander un appareil mobile (modélisme).
- Physical computing: Au sens large, construire des systèmes physiques interactifs qui utilisent des logiciels et du matériel pouvant s'interfacer avec des capteurs et des actionneurs.
- Hacker, Prototypage, Education, etc. [10]

### **II.3. Avantages d'Arduino**

Le système Arduino, nous donne la possibilité d'allier les performances de la programmation à celles de l'électronique. Plus précisément, nous allons programmer des systèmes électroniques. Les principaux avantages de l'électronique programmée sont :

- Simplifie grandement les schémas électroniques.
- Diminué le coût de la réalisation.
- La charge de travail à la conception d'une carte électronique.
- Environnement de programmation clair et simple.
- Multiplateforme : tourne sous Windows, Macintosh et Linux.
- Nombreuses bibliothèques disponibles avec diverses fonctions implémentées.
- Logiciel et matériel open source et extensible.
- Nombreux conseils, tutoriaux et exemples en ligne (forums, site perso, etc.).
- Existence de « shield » (boucliers en français).

### **II.4. Présentation de la carte**

#### **II.4.1. Qu'est ce qu'un microcontrôleur**

Les cartes Arduino font partie de la famille des microcontrôleurs. Un microcontrôleur est une petite unité de calcul accompagné de mémoire, de ports d'entrée/sortie et de périphériques permettant d'interagir avec son environnement. Parmi les périphériques, on recense généralement des Timers, des convertisseurs analogique-numérique, des liaisons Séries, ... etc. On peut comparer un micro contrôleur à un ordinateur classique, mais avec un autre système d'exploitation et avec une puissance de calcul considérablement plus faible.

Les microcontrôleurs sont inévitables dans les domaines de l'informatique embarquée, de l'automatique et de l'informatique industrielle. Ils permettent de réduire

le nombre de composant et de simplifier la création de cartes électroniques logiques [11].

#### II.4.2. Microcontrôleur ATMEL ATmega328

Le microcontrôleur utilise sur la carte Arduino UNO est un microcontrôleur ATmega328. C'est un microcontrôleur ATMEL de la famille AVR 8bits.

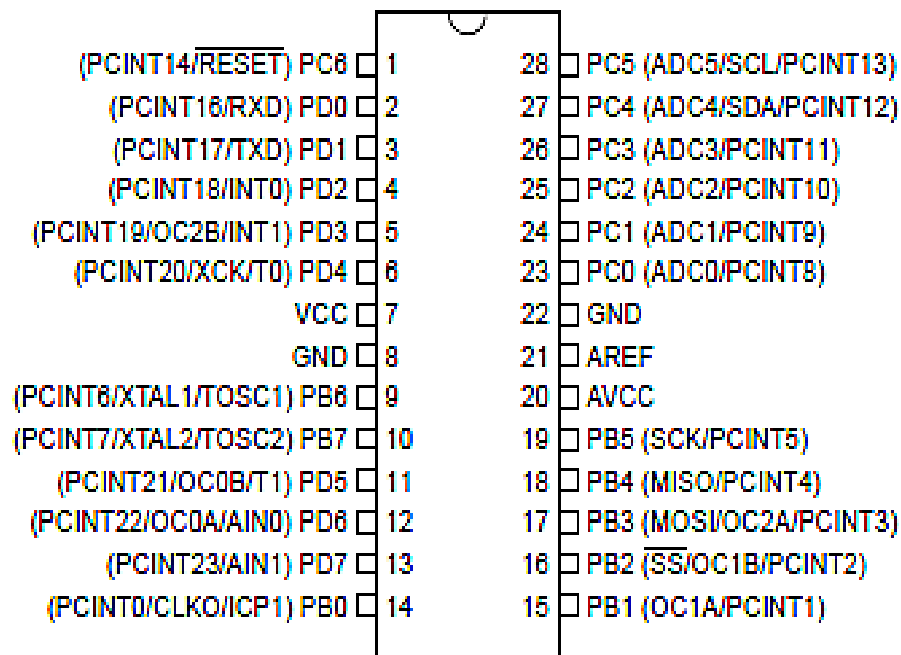


Figure II.1: Datasheet ATmega328.

Les principales caractéristiques d'ATmega328 sont :

- **FLASH** = mémoire programme de 32Ko.
- **SRAM** = données (volatiles) 2Ko.
- **EEPROM** = données (non volatiles) 1Ko.
- **Digital I/O (entrées-sorties Tout Ou Rien)** = 3 ports Port B, Port C, Port D (soit 23 broches en tout I/O).
- **Timers/Counters** : Timer0 et Timer2 (comptage 8 bits), Timer1 (comptage 16 bits) Chaque timer peut être utilisé pour générer deux signaux PWM. (6 broches OCxA/OCxB).
- **Plusieurs broches multi-fonctions**: certaines broches peuvent avoir plusieurs fonctions différentes choisies par programmation.
- **PWM** = 6 broches OC0A(PD6), OC0B(PD5), OC1A(PB1), OC1B(PB3), OC2A(PB3), OC2B(PD3)

- **Analog to Digital Converter** (resolution 10 bits) = 6 entrees multiplexes ADC0(PC0) à ADC5(PC5).
- **Gestion bus I2C** (TWI Two Wire Interface) = le bus est exploite via les broches SDA(PC5)/SCL(PC4).
- **Port série (USART)** = émission/réception série via les broches TXD(PD1)/RXD(PD0)
- **Comparateur Analogique** = broches AIN0(PD6) et AIN1 (PD7) peut déclencher interruption
- **Watch dog Trimer programmable.**
- **Gestion d'interruptions (24 sources possibles (cf. interruptif vecteurs))** : en résumé
  - Interruptions liées aux entrées INT0 (PD2) et INT1 (PD3).
  - Interruptions sur changement d'état des broches PCINT0 a PCINT23.
  - Interruptions liées aux Timers 0, 1 et 2 (plusieurs causes configurables).
  - Interruption liée au comparateur analogique.
  - Interruption de fin de conversion ADC.
  - Interruptions du port série USART.
  - Interruption du bus TWI (I2C). [12]

### II.4.3. Caractéristiques techniques de l'Arduino UNO

Un des modèles les plus répandu de carte Arduino est l'Arduino UNO (voir Fig. II.2). C'est la première version stable de carte Arduino. Elle possède toutes les fonctionnalités d'un microcontrôleur classique en plus de sa simplicité d'utilisation. Elle utilise une puce ATmega328P (1) cadencée à 16 Mhz. Elle possède 32 ko de mémoire flash destinée à recevoir le programme, 2 ko de SRAM (mémoire vive) et 1 ko d'EEPROM (mémoire morte destinée aux données). Elle offre 14 pins (broches) d'entrée/sortie numérique (données acceptée 0 ou 1) (2) dont 6 pouvant générer des PWM (Pulse Width Modulation, détaillé plus tard). Elle permet aussi de mesurer des grandeurs analogiques grâce à ces 6 entrées analogiques (3). Chaque broche est capable de délivrer un courant de 40 mA pour une tension de 5 V.

La carte Arduino peut aussi s'alimenter et communiquer avec un ordinateur grâce à son port USB (4). On peut aussi l'alimenter avec une alimentation comprise en 7V et 12V grâce à sa connecteur Power Jack (5). [11]

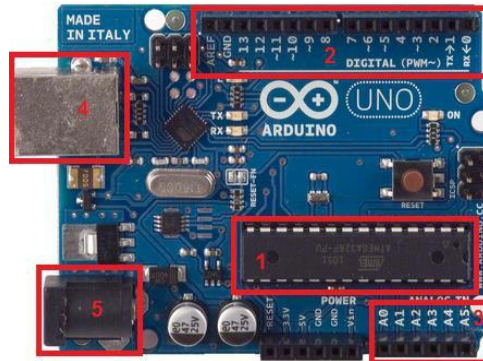


Figure II.2: La carte Arduino UNO.

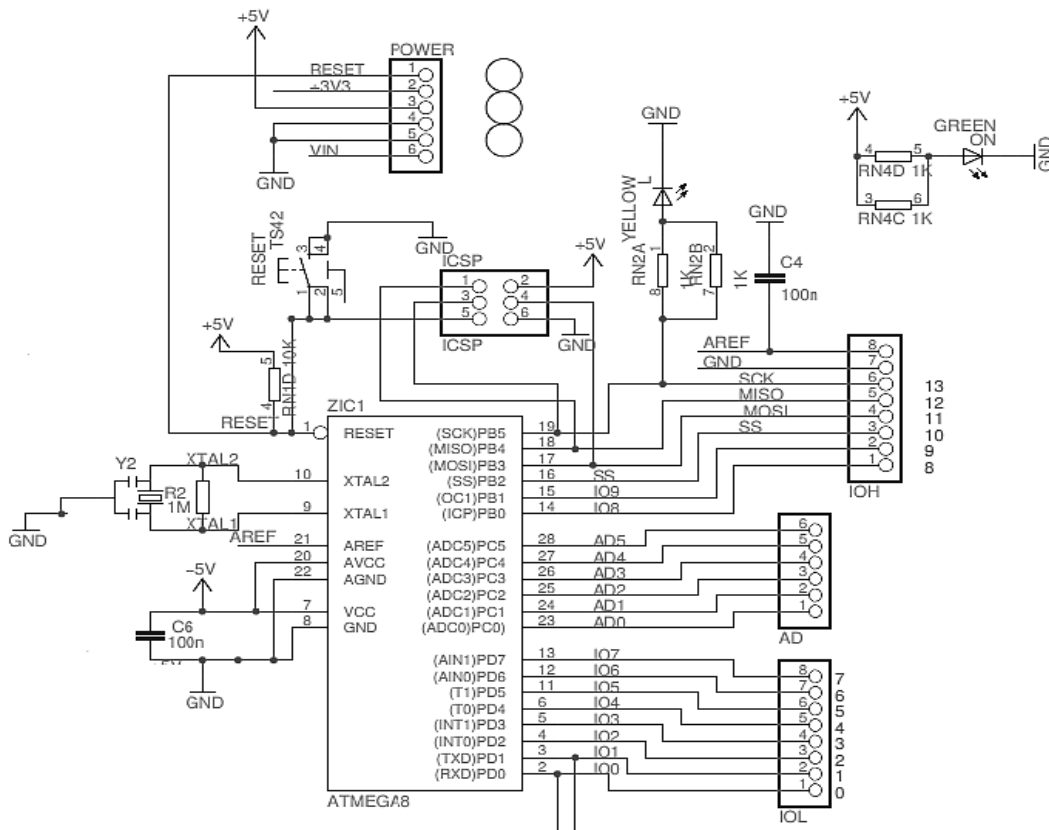


Figure II.3: Schéma simplifié de la carte Arduino UNO.

#### II.4.4. Les différents éléments de la carte de commande sont regroupés dans le Tableau

<b>Tension de fonctionnement</b>	5 V
<b>Tension d'alimentation (recommandée)</b>	7-16 V
<b>Tension d'alimentation (limites)</b>	6-20 V
<b>Broches E/S numériques</b>	14(dont 6 disposent d'une sortie PWM).
<b>Broches d'entrées analogiques</b>	6 (utilisables en broches E/S numériques).
<b>Intensité maxi disponible par broche E/S (5 V)</b>	40 mA (attention : 200 mA cumulé pour l'ensemble des broches E/S ).
<b>Intensité maxi disponible pour la sortie 3.3 V</b>	50 mA
<b>Intensité maxi disponible pour la sortie 5 V</b>	Fonction de l'alimentation utilisée - 500 mA max si port USB utilisée seul

<b>Mémoire programme flash</b>	32 KB (ATmega328) dont 0.5 KB sont utilisés par le boot loader (programme de base préprogrammé conçu pour établir la communication entre l' ATmega et le logiciel Arduino).
<b>Mémoire SRAM (mémoire volatile)</b>	2 KB (ATmega).
<b>Mémoire EEPROM (mémoire non volatile)</b>	1 KB (ATmega).
<b>Vitesse d'horloge</b>	16 Mhz

**Tableau II.1:** Différents éléments de la carte de commande.

### I.5. Capteurs


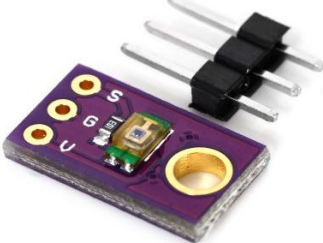
Un capteur est un dispositif ayant pour tâche de transformer une mesure physique observée en une mesure généralement électrique qui sera à son tour traduite en une donnée binaire exploitable et compréhensible par un système d'information.



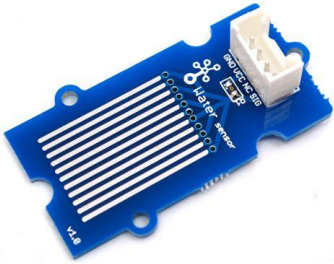
### II.5. 1.Présentation des capteurs

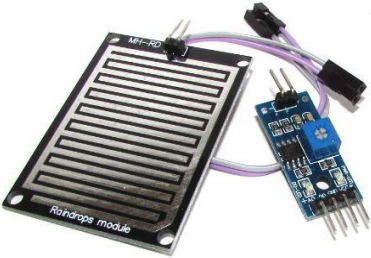


Pour la plupart des projets, il est souvent nécessaire d'ajouter des fonctionnalités aux cartes Arduino. Plutôt que d'ajouter soit même des composants extérieurs (sur une platine d'essai, circuit imprimé, etc.), il est possible d'ajouter des shields.

Un shield est une carte que l'on connecte directement sur la carte Arduino qui a pour but d'ajouter des composants sur la carte. Ces shields viennent généralement avec une librairie permettant de les contrôler. On retrouve par exemple, des shields Ethernet, de contrôle de moteur, lecteur de carte SD, etc.

Le principal avantage de ces shields est leurs simplicités d'utilisation. Il suffit des les emboîter sur la carte Arduino pour les connecter, les circuits électronique et les logiciels sont déjà faits et on peut en empiler plusieurs. C'est un atout majeur pour ces cartes pour pouvoir tester facilement de nouvelles fonctionnalités. Cependant il faut bien garder à l'esprit que les shields ont un prix. Suivant les composants qu'ils apportent, leurs prix peuvent aller de 2 à 100\$.

Nom de capteurs	Image de capteurs	Principales caractéristiques
capteur de Pression BMP180		<ul style="list-style-type: none"> <li>✓ <math>V_{in}</math>: 3 V à 5 V.</li> <li>✓ Plage de détection de pression: 30-110 KPa (9000 m à -500 m au-dessus du niveau de la mer).</li> <li>✓ Jusqu'à 0.003 KPa / 0.25 m de résolution.</li> <li>✓ plage de mesure de -40 à + 85 ° C de, +2 ° C de précision de température [13].</li> </ul>
capteur d'ensoleillement TEMT6000	<p>TEMT6000 Light Sensor</p> 	<ul style="list-style-type: none"> <li>✓ Alimentation: 3 à 5 V</li> <li>✓ Adapté à la réactivité des yeux humains.</li> <li>✓ Température de service: -40°C à +85°C</li> <li>✓ Composant sans plomb [13].</li> </ul>

<p>Capteur de distance a ultrason SRF05</p>		<ul style="list-style-type: none"> <li>✓ tension d'entrée : 5v dc</li> <li>✓ Courant de repos: &lt; 2ma</li> <li>✓ niveau (élevé) de sortie: 5v</li> <li>✓ niveau de sortie (de bas): 0v</li> <li>✓ angle induction: &lt; 15 °</li> <li>✓ distance de détection: 2cm - 450cm</li> <li>✓ Précision: jusqu'à 0.3cm</li> <li>✓ taches aveugles: 2cm. [14]</li> </ul>
<p>Photosensible (module de capteur de luminosité résistance)</p>		<ul style="list-style-type: none"> <li>✓ en utilisant capteur photosensible résistance sensible type;</li> <li>✓ tension de Fonctionnement: 3.5 V, 5 V;</li> <li>✓ forme de sortie: numérique de commutation spectacle (0 et 1);</li> <li>✓ trou de vis fixe, installation facile;</li> <li>✓ petit PCB Taille: 3.2 cm * 1.7 cm;</li> <li>✓ la sortie du comparateur, le signal propre, bonne vague, la capacité de conduire est forte, pour plus que 15 mA;</li> <li>✓ avec réglable potentiomètre pour régler la luminosité de la lumière. [15]</li> </ul>
<p>Détecteur de niveau d'eau Grove 101020018</p>		<ul style="list-style-type: none"> <li>✓ Interface: compatible Grove</li> <li>✓ Signal: digital ou analogique</li> <li>✓ Faible consommation</li> <li>✓ Dimensions: 40 x 20 x 12 mm</li> <li>✓ Température de service: 0°C à +40°C</li> <li>✓ Connectique non compatible avec Tinker Kit</li> <li>✓ Référence Speede studio: 101020018 (remplace SEN11304P). [16]</li> </ul>

<p>capteur de pluie double face</p>		<ul style="list-style-type: none"> <li>✓ Tension : 3,3V-5V</li> <li>✓ Dimension du capteur : ~ 40 x 54 mm</li> <li>✓ Pistes sur capteur : Double face. Les deux faces du capteur sont actives.</li> <li>✓ Potentiomètre de sensibilité et 2 LED de contrôle intégré au module</li> <li>✓ Module livré avec deux câbles Femelle / Femelle. [13]</li> </ul>
<p>capteur infrarouge d'évitement d'obstacles</p>		<ul style="list-style-type: none"> <li>✓ VCC: alimentation 3v-12v (pouvant se connecter directement à 5V ou 3,3V microcontrôleur)</li> <li>✓ GND: Masse</li> <li>✓ OUT: interface de sortie numérique de la carte (0 et 1). [13]</li> </ul>
<p>capteur de flamme</p>		<ul style="list-style-type: none"> <li>✓ haute sensibilité du récepteur IR</li> <li>✓ extrêmement sensibles aux ondes entre 760-1100nm</li> <li>✓ voyant d'alimentation</li> <li>✓ voyant comparateur de sortie</li> <li>✓ sortie analogique quantité</li> <li>✓ seuil de renversement du niveau de sortie électrique</li> <li>✓ seuil réglé par potentiomètre</li> <li>✓ Gamme de détection d'angle: environ 60 degrés</li> <li>✓ Alimentation: 0-15 V DC. [13]</li> </ul>

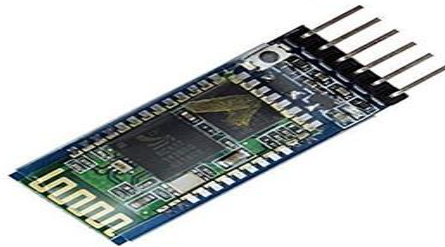
**Tableau II.2:** Représente différents shields qui connectée avec arduino

## I.5.2.Communication

Le constructeur a suggéré qu'une telle carte doit être dotée de plusieurs ports de Communications; on peut éclaircir actuellement quelques types.

### I.5.2.1.Module Arduino Bluetooth

Le Module Microcontrôleur Arduino Bluetooth est la plateforme populaire Arduino avec une connexion sérielle Bluetooth à la place d'une connexion USB, très faible Consommation d'énergie, très faible portée (sur un rayon de l'ordre d'une dizaine de mètres), faible débit, très bon marché et peu encombrant.



**FigureII.4 :** Type de modules Bluetooth

### I.5.2.2. Module shield Arduino Wifi

Le module Shield Arduino Wifi permet de connecter une carte Arduino à un réseau internet sans fil Wifi.



**Figure II.5:** Module shield wifi

## II.6. Présentation du logiciel

### II.6.1. IDE Arduino

Un IDE (environnement de développement) libre et gratuit est distribué sur le site d'Arduino (compatible Windows, Linux et Mac) à l'adresse <http://arduino.cc/en/main/software>. D'autres alternatives existent pour développer l'Arduino (extensions pour Code Blocks, Visual Studio, Eclipse, XCode, etc.) mais nous n'aborderons dans ce chapitre que l'IDE officiel.

L'interface de l'IDE Arduino est plutôt simple (Figure II.6), il offre une interface minimale et épurée pour développer un programme sur les cartes Arduino. Il est doté d'un éditeur de code avec coloration syntaxique (1) et d'une barre d'outils rapide (2). Ce sont les deux éléments les plus importants de l'interface, c'est ceux que l'on utilise le plus souvent. On retrouve aussi une barre de menus (3) plus classique qui est utilisée pour accéder aux fonctions avancées de l'IDE. Enfin, une console (4) affichant les résultats de la compilation du code source, des opérations sur la carte, etc.

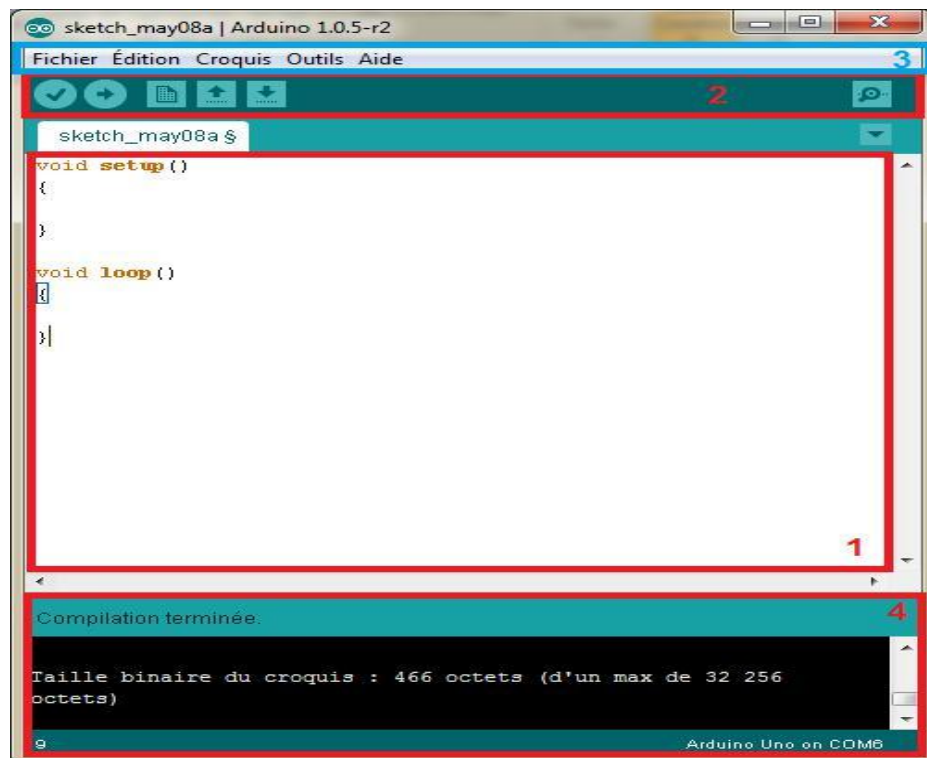


Figure II.6 : IDE Arduino.

### II.6.1.1. Les boutons

Voyons à présent à quoi servent les boutons numérotés en rouge.



**Figure II.7 :** Présentation des boutons.

- Bouton 1 (verify): Ce bouton permet de vérifier le programme, il actionne un module qui cherche les erreurs dans votre programme.
- Bouton 2 (upload): Compiler et envoyer le programme vers la carte.
- Bouton 3 (new): Créer un nouveau fichier.
- Bouton 4 (open): Charger un programme existant.
- Bouton 5 (save): Sauvegarder le programme en cours.
- Bouton 6 (serial monitor): Permet d'accéder au port série (en RX/TX).

### II.6.2. Langage Arduino

Le langage Arduino est inspiré de plusieurs langages. On retrouve notamment des similarités avec le C, le C++, le Java et le Processing. Le langage impose une structure particulière typique de l'informatique embarquée. La fonction `setup` (voir Fig.8) contiendra toutes les opérations nécessaires à la configuration de la carte (directions des entrées sorties, débits de communications série, etc.). La fonction `loop`, elle est exécutée en boucle après l'exécution de la fonction `setup`. Elle continuera de boucler tant que la carte n'est pas mise hors tension, redémarrée (par le bouton *reset*). Cette boucle est absolument nécessaire sur les microcontrôleurs étant donné qu'il n'y a pas de système d'exploitation. En effet, si l'on omettait cette boucle, à la fin du code produit, il sera impossible de reprendre la main sur la carte Arduino qu'exécuterait alors du code aléatoire.

```
void setup()           //fonction d'initialisation de la carte
{
    //contenu de l'initialisation
}

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

**Figure II.8 :** Un code minimal.

Au niveau de la syntaxe, on retrouve des similarités avec les langages précédemment cités. La déclaration des variables se fait généralement dans l'espace global (de façon à partager les variables les plus importantes entre les deux fonctions principales). On retrouve les types de base suivant (voir Tableau II.3):

Nom	Contenu	Taille (en octet)	Plage de valeurs
<i>(unsigned) char</i>	Entier ou caractère	1	(0->255) -128 -> 127
<i>(unsigned) int</i>	Entier	2	(0->65 535) -32 768 -> 32 767
<i>(unsigned) long</i>	Entier	4	(0 -> 4 294 967 295) -2 147 483 648 -> 2 147 483 647
<i>float/double</i>	Nombre à virgule flottante	4	-3,4028235E+38 -> 3,4028235E+38
<i>String</i>	Chaine de caractères (Objet)	variable	Aucune
<i>boolean</i>	Booléen	1	True / False

**Tableau II.3:** Les types de base.

Il existe d'autres types de base mais ils ne sont qu'un alias de ceux cités précédemment, la liste des types est disponible sur la page des références du site Arduino (<http://arduino.cc/en/Reference/HomePage>). La déclaration des variables suit cette syntaxe:

*(const) <type> <nom>([<longueur du tableau>]) (= valeur);*

### **Exemples**

*const int constante = 12 ;*

*float univers = 42.0 ;    char lettre = 'b' ;*

*String chaine = "Hello World ";*

*long tableau[12] ;*

*boolean vrai = true ;*

On retrouve les opérateurs les plus courants pour les types de bases. Parmi eux, = (affectation), == (comparaison), != (différence), <, >, <=, >=, && (et logique), || (ou logique), ! (non logique). On retrouve aussi les opérateurs mathématiques (+, -, \*, /, %) et les opérateurs logiques bit à bit (^ (XOR), & (et), | (ou), ~(non), << (décalage logique à gauche), >> (décalage logique à droite)).

Les structures de contrôle sont elles aussi similaires aux langages de références. On y retrouve toutes les structures de contrôle standard, conditions, boucle, switch,

fonctions, etc. On peut aussi écrire des structures et des classes. Chaque structure de contrôle est suivie d'un bloc d'instructions délimitées par des accolades. Voici une liste des structures de contrôles les plus utilisées :

Nom	Utilité	Syntaxe
If-else	Condition logique	<b>If</b> ( <valeur booléenne> ) { <instruction> } <b>else</b> { <instruction> }
If-else if - else	Condition logique multiples	<b>If</b> ( <valeur booléenne> ) { <instruction> } <b>else if</b> ( <valeur booléenne> ) { <instruction> } <b>else</b> { <instruction> }
Switch	Sélecteur	<b>Switch</b> ( <variable> ) { <b>case</b> <valeur> : <instruction> <b>break</b> ; <b>default</b> : <instruction> }
While	Boucle	<b>While</b> ( <valeur booléenne> ) { <instruction> }
For	Boucle itérative	<b>For</b> ( <initialisation> ; <valeur booléenne> ; <évolution> ) { <instruction> }

**Tableau II.4:** Les structures de contrôle.

Les fonctions, les structures et les classes se déclarent de la même façon qu'en C++. Elles se déclarent sous cette forme :

Structur:

**struct**

{ <nom><type><nomduchp>; }

Fonction :

<type de reto <nom>( <paramètre> )

{ <instruction> }

Classe : `class <nom> { public : <attributs et champ publics>`

`private :<attributs et champ privés>`

`protected :<attribut et champ privés> ;}`

### II.6.3. Fonctionnalité de base

#### II.6.3.1. Les entrées/sorties

Le langage Arduino vient avec un nombre important de fonction de base permettant d'interagir avec son environnement. Les fonctions les plus utilisées sont les fonctions d'entrée/sorties. Ce sont elles qui permettent d'envoyer ou de mesurer une tension sur une des broches de la carte.

Dans un premier temps, avant d'effectuer une mesure ou d'envoyer une commande. Il est nécessaire de définir la direction des broches utilisées. Pour cela on fait appel à la fonction `pin Mode` en lui donnant d'une part, la broche concernée, et d'autre part, la direction :

```
Void setup() {  
  
  pinMode(1, OUTPUT) ; // Broche 1 en sortie  
  
  pinMode(2, INPUT) ; // Broche 2 en entrée  
  
}
```

Une fois cette configuration faite, on peut procéder à l'utilisation des broches. Toutes les broches sont capables d'écrire et de lire des données numériques (c'est-à-dire des 0 (0V) ou des 1 (5V)). Mais, certaines disposent de fonctionnalité supplémentaire.

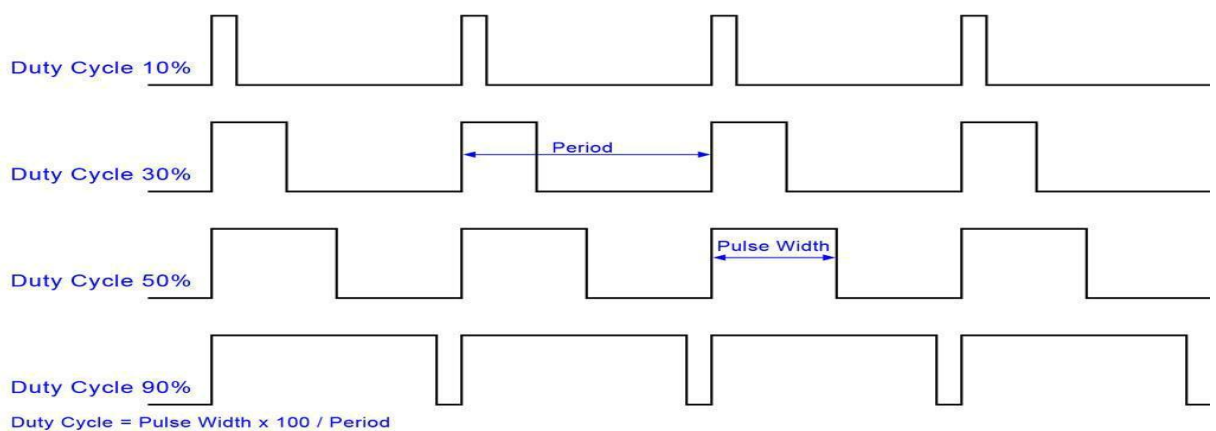
Tout d'abord, toutes les cartes Arduino possèdent des entrées analogiques. Ce sont les broches A0-A1-A2 etc. Elles permettent de lire des tensions analogiques (comprise entre 0 et 5V) et de le convertir en entier (compris entre 0 et 1023) proportionnellement à la tension mesurée. Certaines cartes Arduino possède des sorties analogique faisant l'opération inverse (met une tension sur la broche proportionnellement à l'entier donné), mais ce n'est pas le cas pour l'Arduino UNO.

Pour pouvoir tout de même contrôler des composants autrement qu'en « tout ou rien » il est possible d'utiliser des broches PWM. Ce sont les broches annotés par un tilde ~ sur la carte. Les PWM (Pulse Width Modulation) sont utilisées pour synthétiser

des signaux analogiques en modulant le temps passé à l'état 1 (5V). Le signal obtenu est représenté (figure II.9). En utilisant une fréquence relativement élevée, les PWM permettent de commander certains composants comme si il recevait une tension analogique. Cela provient du fait que les composants utilisés dans l'électronique analogique, ne changes pas d'états instantanément. Par exemple, une ampoule à incandescence reste chaude et éclaire un court instant après avoir été éteinte. Ce phénomène est généralement invisible à l'œil nu. Grâce à elles, on pourra par exemple faire varier l'intensité d'une LED. La plupart des cartes Arduino utilisent des PWM cadencées à 490 Hz environ.

Toutes ces fonctionnalités sur les broches d'entrées sorties sont utilisables par le biais de quatre fonctions :

- **digitalRead(pin)**: mesure une donnée numérique sur une des broches, la broche en question doit être réglée en entrée.
- **digitalWrite(pin, value)**: écrit une donnée numérique sur une des broches, la broche concernée doit être réglée en sortie. Le paramètre *value* doit être égal à *HIGH* (état 1 soit 5V) ou *LOW* (état 0 soit 0V).
- **analogRead(pin)**: mesure une donnée analogique sur une des broches (compatible seulement), la broche doit être réglée sur entrée.
- **analogWrite(pin, value)**: écrit une donnée sous forme de PWM sur une des broches (compatible uniquement), la broche doit être réglée en sortie. Le paramètre *value* doit être compris dans l'intervalle  $[0;255]$ .



**Figure II.9:** Signal PWM.

### II.6.3.2. Gestion du temps

Pour la plupart des applications de domotique, il est nécessaire de faire intervenir des intervalles de temps. Par exemple, pour gérer le temps d'appui sur un bouton ou pour faire une sonnerie qui se répète un certain nombre de fois. Le langage Arduino fournit quelques fonctions permettant de gérer le temps.

Il est possible d'insérer une pause dans son programme pendant un instant. Pour cela, on utilise les fonctions `delay` et `delayMicroseconds` qui insère une pause suivant le paramètre passé (en milliseconde pour l'un, en microseconde pour l'autre). Cependant ces fonctions bloquent le microcontrôleur, on ne peut alors plus effectuer aucune action.

En plus d'insérer une pause, il est possible de mesurer le temps. De la même manière que les fonctions de délai, on utilise les fonctions `millis` et `micros` qui donnent le nombre de milliseconde (respectivement microseconde) depuis le lancement de la carte. Attention, ces fonctions incrémentent une variable (interne). Ces variables se remettent à zéro une fois le maximum atteint (overflow). La variable utilisée pour les millisecondes atteint son maximum au bout de 49 jours et 17 heures et la variable utilisée pour les microsecondes au bout de 71 minutes et 34 secondes environ. Il faut donc faire attention lors de l'utilisation de ces fonctions pour des utilisations longues durées.

### II.6.3.3. Les interruptions

Il est parfois nécessaire en informatique embarquée, d'attendre un événement externe (appui sur un bouton, données d'un capteur, etc.) pour effectuer une action. Pour ce type de problème, on utilise les interruptions. Les interruptions sont des portions de code (fonctions) appelés lorsque qu'un événement (interne ou externe) survient et a besoin d'être traité sur le champ. Il faut cependant faire attention, ce mécanisme interrompt le code exécuté, il est prioritaire par rapport au reste du code. Vu qu'il est possible de mesurer les événements ponctuellement (via les fonctions d'entrées/sorties) on utilise généralement les interruptions pour du code critique (arrêt d'urgence par exemple) ou des événements non-ponctuels (transmissions de données depuis un ordinateur par exemple).

Aussi, le nombre d'interruption externe est limité sur à 2 sur la plupart des cartes Arduino. Les interruptions sont utilisables sur les broches compatibles seulement (broches 2 et 3 sur l'Arduino UNO). Pour choisir la fonction et la broche utilisée pour

l'interruption, on utilise la fonction `attachInterrupt`. On peut utiliser `detachInterrupt` pour supprimer l'interruption. Il est possible de partir en interruptions sur 4 types d'événements :

- **LOW** : Lorsque la broche est à l'état 0 (0V)
- **RISING** : Lorsque la broche passe de l'état 0 (0V) à l'état 1 (5V) (front montant).
- **FALLING** : Lorsque la broche passe de l'état 1 (5V) à l'état 0 (0V) (front descendant).
- **CHANGE** : Lorsque la broche change d'état (front montant et front descendant).

Voici un exemple d'utilisation :

```
volatile boolean etat = false ;
void appuiBouton() {
    etat = !etat ; // Changement d'état }
void setup() {
    pinMode(2,INPUT) ; // Broche 2 en entrée
    attachInterrupt(0,appuiBouton,RISING) ; // On attache à l'interruption 0 (broche 2) la
                                             fonction appui Bouton sur un front
                                             montant
}
```

On remarque l'apparition du mot clef `volatile` avant la déclaration de la variable `etat`. Ce mot clef est nécessaire pour toutes les variables qui sont modifiées dans une interruption. Cela a une incidence sur la manière dont le compilateur traite l'accès à la variable.

Il est parfois nécessaire de désactiver temporairement les interruptions par exemple lorsque l'on exécute du code critique (activation d'un moteur, etc.). Deux fonctions permettent de changer l'activation des interruptions `interrupts` et `noInterrupts` pour activer (respectivement désactiver) les interruptions.

### Exemple

Nous allons maintenant passer à un exemple. Il est de faire clignoter une LED de façon régulière. On doit utiliser une LED avec une résistance en série sur une breadboard. Ou on peut utiliser la LED directement sur l'Arduino (broche 13 sur la carte Arduino UNO).

**Code avec delay :**

```
const int pinLed = 13;

void setup()

{ pinMode(pinLed, OUTPUT); // Broche 13 en sortie }

void loop()

{

delay(500); // Attente d'une demi seconde
digitalWrite(pinLed, HIGH); // Allumage de laLED
delay(500);

digitalWrite(pinLed, LOW); // Eteignage de laLED

}
```

**Code avec millis :**

```
const int
pinLed=13;

int temps;

int etat;

void setup(){

    pinMode(pinLed, OUTPUT); // Broche 13 en sortie

    etat = LOW; // LED éteinte }

void loop()

{ int present = millis();

    if (temps+500 < present) // Vérification du chronomètre

{ temps = present; // Actualisation du chronomètre

    etat = !etat; // Changement d'état
```

```
digitalWrite(pinLed,etat); // Changement d'état de la LED  
  
} }
```

#### II.6.3.4. Quelques bibliothèques

En plus de la simplicité du langage et des nombreuses fonctionnalités qu'offre. L'IDE vient avec un nombre important de bibliothèques évitant ainsi d'implémenter des fonctions courantes dans l'informatique embarquée.

Une des bibliothèques les plus utilisées est celle implémentant la communication série. La majorité des cartes Arduino possède un émulateur de connexion série pour communiquer au travers de l'USB. Ainsi, on peut communiquer avec l'ordinateur sur lequel la carte Arduino est connectée. Cela permet, par exemple, de déboguer un programme en affichant la valeur des variables ou simplement afficher la valeur des capteurs. Cette bibliothèque a été directement implémentée dans le langage Arduino. On peut accéder à la communication série (au travers l'USB) grâce à l'objet Serial. Une autre bibliothèque existe pour communiquer par liaison série via une des broches de la carte.

Il est parfois nécessaire de stocker des informations même après l'arrêt de la carte. Il est possible de stocker une petite quantité d'information sur la mémoire EEPROM (Electrically Erasable Programmable Read Only Memory) intégrée. Une bibliothèque est aussi fournie pour dialoguer avec cette mémoire. Il est possible de lire et d'écrire sur cette mémoire sans rien ajouter sur la carte. Cette mémoire est accessible via l'objet EEPROM et en ajoutant la bibliothèque du même nom. Attention, la mémoire EEPROM a une durée de vie limitée (environ 100 000 cycles d'écritures). Il faut donc veiller à ne pas écrire répétitivement les données. D'autres alternatives existent pour ajouter de plus grandes quantités de mémoires mortes à l'Arduino (carte SD notamment) mais cela demande l'ajout de composants externes.

Pour terminer, il existe aussi des bibliothèques permettant de contrôler des composants externes. Parmi ces bibliothèques, une des plus populaires est celle contrôlant des servomoteurs.

Un servomoteur est un composant électronique composé d'un moteur et d'une carte d'asservissement. On peut le contrôler en position (c'est-à-dire choisir l'angle du moteur) grâce aux PWM. Une bibliothèque Arduino est implémentée pour contrôler simplement ces moteurs : la bibliothèque servo. Il faut toutes fois faire attention à éviter

d'alimenter les servomoteurs directement sur une des broches d'alimentation de la carte. En effet, un servomoteur consomme une quantité important de courant (suivant le couple exercé par le moteur). Il est donc fortement conseillé de relier les servomoteurs sur des alimentations externes (batteries, piles de 9V, etc.). [17]

## **II.7. Conclusion**

Dans ce chapitre nous avons vu une présentation générale de la carte Arduino UNO, ses caractéristiques et présentation des quelques shields et on mettra la lumière sur le logiciel IDE Arduino.

On peut conclure sur le fait que les cartes Arduino sont des outils puissants pour les cartes électroniques Mais aussi, elles permettent un accès facile et intuitif à l'informatique embarqué. On pourra ainsi enrichir tout ces projets à base d'un microcontrôleur pour leurs donner une valeur plus importante.

La carte d'Arduino UNO est la plus simple carte et la plus courante. Il existe d'autres versions de cartes Arduino plus adaptées pour certains projets. Dans notre travail on va utiliser la carte d'Arduino UNO pour réaliser un système de commande à distance des installations électrique pour la domotique connectée par wifi. Pour certains projets plus complexes, il va falloir d'utiliser l'Arduino Méga.

### III.1. Introduction

Dans la vie moderne, on utilise pas mal d'outils et d'accessoires de commande à distance afin de simplifier notre contrôle, donc nous chercherons toujours à se concentrer sur la souplesse de la commande et de contrôler sur une zone bien définie le plus grand nombre possible d'accessoires. [1]

Le smart phone occupe la premier place d'objets qui ne nous quittent pas donc notre travail se concentre sur l'utilisation de ce dernier avec bien sur sa liaison avec un système ou une carte de commande tell que l'Arduino.

Dans ce chapitre nous allons donner une description détaillée des shields utilisés et le mode de fonctionnement de ces shields dans notre système de commande, ensuite on va réalésée de faire une petite application App Inventor, qui va nous permettre de connecter notre système par wifi. On va présenter à la fin de ce chapitre les résultats obtenu après la réalisation de ce système de commande à distance des installations électriques

### III.2. Schéma synoptique du système à réaliser

Le schéma synoptique suivant va nous permettre de mieux comprendre le fonctionnement global du système étudié :

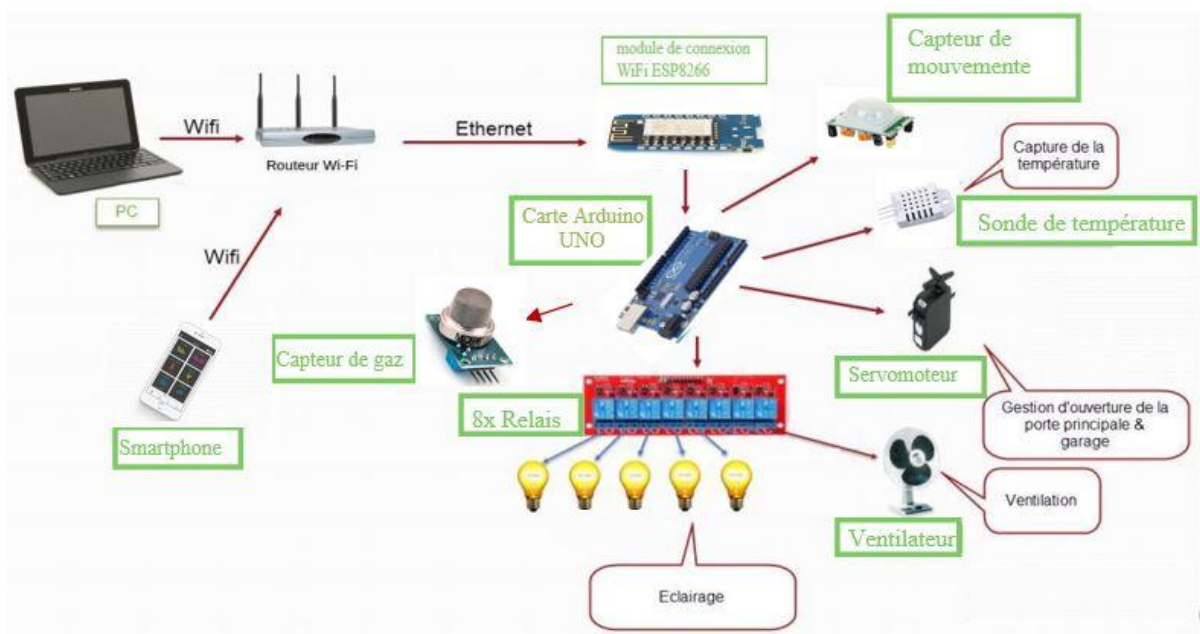


Figure III.1: Le schéma synoptique de système à réaliser.

Ce système est divisé en trois parties, la première partie est une carte qui reçoit les phénomènes physiques (température, humidité, gaz/fumée) et la deuxième partie l'unité de traitement des données (Arduino UNO) et la dernière partie est l'unité de gestion à distance des données (ESP8266).

### III.3. Schéma électrique du montage sur Fritzing

Tous d'abord on a tracé le schéma électrique du système sur le logiciel Fritzing. Le logiciel Fritzing est un outil de création des projets et des circuits électroniques, il permet aussi l'édition de circuit imprimé, il est disponible gratuitement sur internet. Il a notamment pour vocation de favoriser l'échange de circuits électroniques libres et d'accompagner l'apprentissage de la conception de circuits.

Le logiciel comporte trois vues principales:

- La « Platine d'essai », où l'on voit les composants tels qu'ils sont dans la réalité et où l'on construit le montage.
- La « Vue schématique », représentant le schéma fonctionnel du circuit.
- Le « Circuit imprimé », représentant la vue du circuit imprimé tel qu'il sera sorti en PDF pour être imprimé [18]. Dans les figures III.2 et III.3 on présente respectivement, le schéma électrique tracé sur Fritzing et le montage de notre système sur plaque d'essai obtenu aussi par Fritzing.

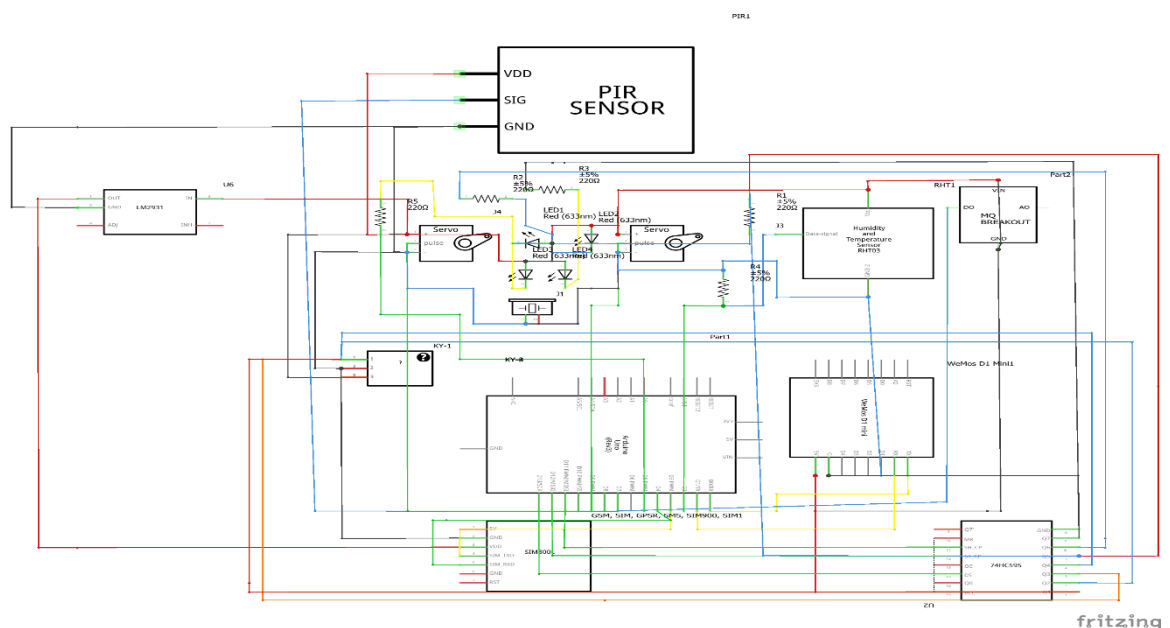
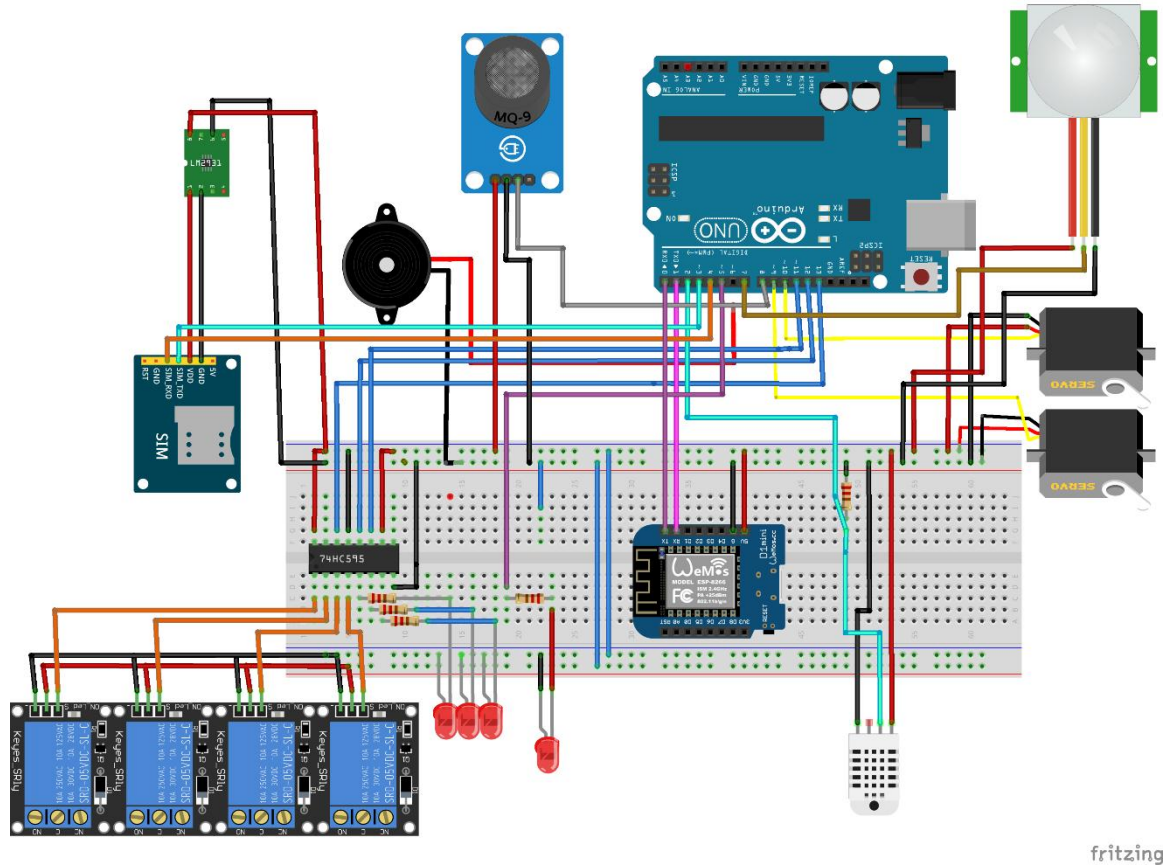


Figure III.2: Schéma électrique du montage.



fritzing

Figure III.3: Montage en plaque d'essai.

### III.4. Description des shields utilisés.

Pour réaliser étude et réalisation d'un système de commande à distance des installations électriques pour la domotique, il faut tout d'abord choisir les composants sensibles aux la maison intelligente que l'on souhaite commander. On a fixé le choix sur la ouverture et fermeture la porte et le garage, l'éclairage extérieur et intérieur les systèmes de sécurité,... etc. Dans ce qui suit, on va décrire une description détaillée des différents shields utilisés.

### III.4. 1.Capteur de Température et d'Humidité (DHT22)

#### III.4.1.1. Description du capteur

Le DHT22 est un capteur de température et d'humidité numérique basique à faible coût. Il utilise un capteur d'humidité capacitif et une thermistance pour mesurer l'air environnant et émet un signal numérique sur la broche de données (aucune broche d'entrée analogique n'est requise). Il est assez simple à utiliser (voir Figure III.4), mais nécessite un temps minutieux pour saisir les données. Le seul inconvénient réel de ce capteur est que vous ne pouvez obtenir de nouvelles données qu'une fois toutes les 2 secondes, alors lorsqu'on déclare la bibliothèque de ce capteur dans le programme, la lecture de ce capteur peut durer jusqu'à 2 secondes.



**Figure III.4:** Image réelle du capteur de Température et d'Humidité DHT22.

Il suffit de connecter la première broche à gauche à l'alimentation de 3 à 5 V, la deuxième broche doit brancher à la carte Arduino, juste au pin 2 numérique. La troisième broche est montée à la masse GND.[13]

#### III.4.1.2. Principales caractéristiques de DHT22

- ❖ À bas prix.
- ❖ Puissance et E / S de 3 à 5 V.
- ❖ 2.5 mA max courant d'utilisation pendant la conversion (lors de la demande de données).
- ❖ Bon pour 0-100% de lectures d'humidité avec 2-5% de précision.
- ❖ Bon pour les lectures de température de -40 à 80 ° ± Précision ± 0,5 ° C.
- ❖ Pas plus de 0,5 Hz de fréquence d'échantillonnage (une fois toutes les 2 secondes).
- ❖ Taille du corps 27 mm × 59 mm × 13.5 mm (1.05 "× 2.32" × 0.53 ").

- ❖ Poids (juste le DHT22): 2,4 g. [15]

### III.4.2. Capteur de gaz (MQ2)

#### III.4.2.1. Description du capteur

Le capteur de gaz MQ2 est utilisé pour la détection des fuites de gaz pour les équipements des marchés de grandes consommations et industriel. Ce capteur est conçu pour détecter le LPG, i-butane, propane, méthane, alcool, hydrogène et la fumée. Il a une grande sensibilité et un temps de réponse rapide. Sa sensibilité peut d'ailleurs être ajustée par potentiomètre. [19]



**Figure III.5:** Image réelle du capteur de gaz/fumée MQ-2.

#### III.4.2.2. Principales caractéristiques de gaz/fumée MQ-2

- ❖ Alimentation: 5V
- ❖ Type d'Interface: Analogique
- ❖ Connectique: 1-Sortie 2-GND 3-VCC
- ❖ Large panel de détection
- ❖ Réponse rapide et haute sensibilité
- ❖ Circuit de Contrôle Simple
- ❖ Système stable à longue durée de vie
- ❖ Dimensions: 40x20mm

### III.4.3. Capteur de mouvement (PIR)

#### III.4.3.1. Description du capteur

Les capteurs PIR (« passive infra éd », « pyroélectriques » ou « à mouvement infrarouge »), sont utilisés comme des capteurs de mouvement. Ils permettent de déterminer si un être humain (et, dans le cas du capteur PIR d'Ada fruit, un animal) est entrée ou sorti du champ de détection du module. [20]



**Figure III.6:** Image réelle du capteur de mouvement PIR.

#### III.4.3.2. Principales caractéristiques de mouvement PIR.

- ❖ Alimentation : 5 V
- ❖ Signal de sortie numérique : 3,3V
- ❖ Portée : 7m
- ❖ Cône de détection : 120°
- ❖ Sensibilité et délai de réponse (2-4 s) ajustables
- ❖ Longueur : 24,03 mm
- ❖ Profondeur : 32,34 mm
- ❖ Distance des trous de vissage : 28 mm
- ❖ Diamètre des trous de vissage : 2 mm
- ❖ Hauteur (avec lentille) : 24,66 mm
- ❖ Poids : 5,87 g

### III.4.4. Module GSM SIM800L

Le module GSM SIM800L est l'un des plus petits modules GSM du monde avec une taille de 2.2 cm x 1.8 cm. C'est un module puissant qui démarre automatiquement et recherche automatiquement le réseau. Il inclut notamment le Bluetooth 3.0+EDR et la radio FM (récepteur uniquement). Il vous permettra d'échanger des SMS, de passer des appels mais aussi, et c'est nouveau, de récupérer de la data en GPRS 2G+. Ainsi vous pourrez faire transiter des données sur une très longue distance, si par exemple la radio FM ou le Bluetooth ne vous suffit plus. [21]

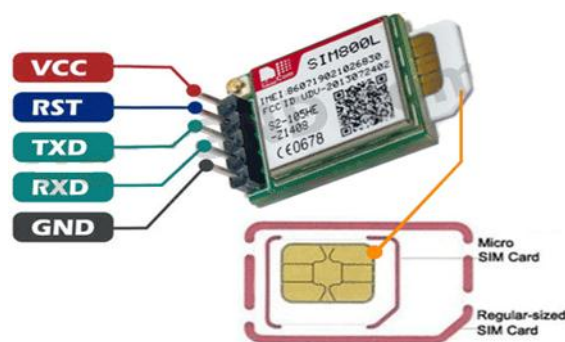


Figure III.7: Image réelle le module GSM SIM800L

#### III.4.4.1. Principales caractéristiques module GSM SIM800L

- ❖ Fréquence 850/900/1800 / 1900MHz
- ❖ Tension de travail 3,5 ~ 4,2 V
- ❖ Dimensions 0.98 in x 0.91 in x 0.28 in (2.5 cm x 2.3 cm x 0.7 cm)
- ❖ Poids 0,18 oz (5,2 g)

#### III.4.5. Module de Relais à 4 canaux

Il s'agit d'une carte d'interface de relais, qui peuvent être contrôlé directement par un large éventail de microcontrôleurs comme Arduino, AVR, PIC, ARM, API, etc.

Ce module de relais est bas actif 5V. Il est également capable de contrôler appareils divers et autres équipements avec le grand courant. Cette interface standard peut être connectée directement avec les microcontrôleurs. Le voyant rouge qui indique l'état de travail est propice à l'utilisation de sécurité. Le module de relais est largement utilisé pour tout contrôle MCU, le secteur industriel, contrôle PLC, contrôle de la maison intelligente.[13]



**Figure III.8 :** Module de relais à 4 canaux

### III.4.6. Servomoteurs

Pour motoriser nos volets des fenêtres ainsi de la porte principale et le garage, on a pensé à utiliser des servomoteurs vue de leur souplesse, simplicité de commande et de leur couple acceptable.

Les servomoteurs servent en principe à actionner les parties mobiles d'un système. Ils sont prévus pour être commandés facilement en position ou en vitesse. En effet, ils sont équipés d'un système d'asservissement basé sur un potentiomètre rotatif qui sert de capteur de rotation. [22]

C'est un ensemble mécanique et électronique comprenant :



- Un moteur à courant continu de petite taille ;
- Un réducteur en sortie de ce moteur diminuant la vitesse mais augmentant le couple.
- Un potentiomètre (faisant fonction de diviseur résistif) qui génère une tension variable, proportionnelle à l'angle de l'axe de sortie ;
- Un dispositif électronique d'asservissement ;
- Un axe dépassant hors du boîtier avec différents bras ou roues de fixation.

Ils sont faciles à utiliser car ils ne nécessitent que :

- Une source de tension continue.
- Une sortie PWM (Pulse Width Modulation) du microcontrôleur de notre robot

### III.4.6.Fonctionnement de servomoteurs

Les servomoteurs sont commandés par l'intermédiaire d'un câble électrique à 3 fils qui permettent d'alimenter le moteur et de lui transmettre des ordres de positions sous forme d'un signal codé en largeur d'impulsion plus communément appelés PWM (Pulse Width Modulation ou Modulation de Largeur d'Impulsion) ou RCO (Rapport Cyclique d'Ouverture). Cela signifie que c'est la durée des impulsions qui détermine l'angle absolu de l'axe de sortie et donc la position du bras de commande du servomoteur. Le signal est répété périodiquement, en général toutes les 20 millisecondes, ce qui permet à l'électronique de contrôler et de corriger continuellement la position angulaire de l'axe de sortie, cette dernière étant mesurée par le potentiomètre.

Servomoteur à rotation continue	Servomoteur à rotation angulaire (Micro-Servo)
	
<p><b>Caractéristiques :</b></p> <p>Alimentation : 4,8 à 6V</p> <p>Angle de rotation : 360°</p> <p>Couple : 3,3 à 4,8 kg.cm</p> <p>Vitesse : 60 à 70 tr/min</p> <p>Dimensions : 41 x 20 x 40 mm</p> <p>Poids : 44 g</p>	<p><b>Caractéristiques :</b></p> <p>Alimentation : 4,8 à 6V</p> <p>Angle de rotation : 180°</p> <p>Couple : 1,3 kg.cm</p> <p>Vitesse : 0,12 sec/ 60°</p> <p>Dimensions : 23,2 X 12,5 X 22 mm</p> <p>Poids : 9g</p>

**Tableau III.1:** Servomoteurs utilisés dans notre projet.

### III.4.7.-Ventilateur

Une installation domotique dédiée permettra de programmer les aspirations et extractions d'air, notamment à des endroits stratégiques et nécessaires comme la salle de bains, connue pour sa grande humidité. Le tout grâce à un simple bouton sur l'application que nous avons adopter derrière lequel se trouve le module d'interface de ventilation.



**Figure III.9 :** Ventilateur 5V.

### III.4.8. Description du module de connexion Wifi ESP8266

Les cartes wifi basées sur le microcontrôleur ESP8266 sont programmables comme les cartes Arduino et peuvent communiquer par wifi avec d'autres appareils (ordinateurs, Smartphones, etc.). Il existe plusieurs modèles : l'ESP-D1, l'ESP-03, l'ESP-12 ... etc. L'ESP-01 que nous allons utiliser dans notre travail est présenté dans la figure III.10.



**Figure III.10:** Image réelle du module Wifi ESP8266 – ESP-D1.

Le module Wifi ESP8266 est un réseau autonome avec une pile de protocoles TCP/IP intégrée qui peut donner accès au réseau Wifi. L'ESP8266 est capable d'héberger une application ou de décharger toutes les fonctions de réseau Wifi d'un autre processeur d'application. Chaque module ESP8266 est préprogrammé avec un microprogramme de configuration de commande AT (voir l'annexe pour la liste des commandes de l'ESP8266), ce qui signifie que vous pouvez simplement le brancher sur votre appareil Arduino et obtenir autant de fonctionnalités Wifi. Le module ESP8266 est un tableau extrêmement rentable avec un ensemble des commandes. [23]

#### III.4.8.1. Mise sous tension

- Alimenter le module en reliant  $V_{cc}$  et CH PD, et câbler GND;
- La LED rouge doit être allumée et le rester;
- La LED bleue peut clignoter au démarrage;
- Vérifiez si votre module ne présente pas son réseau dans la liste des réseaux WIFI (par exemple un nouveau réseau ESP\_99b22. [24])

#### III.4.8.2. Principales caractéristiques d'ESP8266

- ❖ Norme sans fil: IEEE 802.11.
- ❖ Gamme de fréquence: 2.4 GHz ~ 2.5 GHz.
- ❖ Tension de fonctionnement: 3.3v ou 5v.
- ❖ Courant de fonctionnement moyenne 80MA.
- ❖ Taille de paquet: 14.3 mm × 24.8 mm × 3 mm.
- ❖ Sans fil mode réseau: station/softAP/SoftAP + station.

### III.5.Réalisation de la maison intelligente



Figure III.11 : Photos d'extérieur de maquette de maison



Figure III.12: Photos d'intérieur de maquette de maison



**Figure III.13:** Photos de la réalisation finale de maquette de maison

### III.5.1. Développement d'une application Androïde

#### III.5.1.1. le système Androïde

Androïde est un système d'exploitation développé initialement pour les Smart phones. Il utilise un noyau Linux qui est un système d'exploitation libre pour PC et intègre tous utilitaires et les périphériques nécessaires à un smart phone. Il est optimisé pour les outils Gmail. Aussi, l'androïde est libre et gratuit et a été ainsi rapidement adopté par des fabricants.

La société Androïde a été rachetée en 2007 par Google. Mais aujourd'hui, l'Androïde est utilisé dans de nombreux appareils mobiles (smart phones). Les applications sont exécutées par un processeur de type ARM à travers un interpréteur JAVA. En plus de cela, l'androïde concurrence l'opérateur système d'Apple qu'il tend à dépasser en nombre d'utilisateurs. Androïde évolue pour mieux gérer l'hétérogénéité des appareils qu'il utilise. [25]

### III.5.1.2. Application Android dans les Smartphones

Pour faciliter l'accès aux résultats sur un Smartphone nous avons ajouté une application sur Android qui s'appelle " **App Inventor**".

App Inventor est un outil de développement des applications en ligne pour les smart phones sous androïde et permet à chacun de créer son application personnelle pour le système d'exploitation Androïde qui est développée par Google

La plateforme de développement est offerte à tous les utilisateurs possédant un compte Gmail. Elle rappelle certains langages de programmation simplifiés basés sur une interface graphique similaire à Scratch. Les informations des applications sont stockées sur des serveurs distants. [25]

Elles sont actuellement entretenues par le Massachusetts Institute of Technologie (MIT).

L'environnement d'App Inventor contient trois fenêtres qui sont proposées pendant le développement :

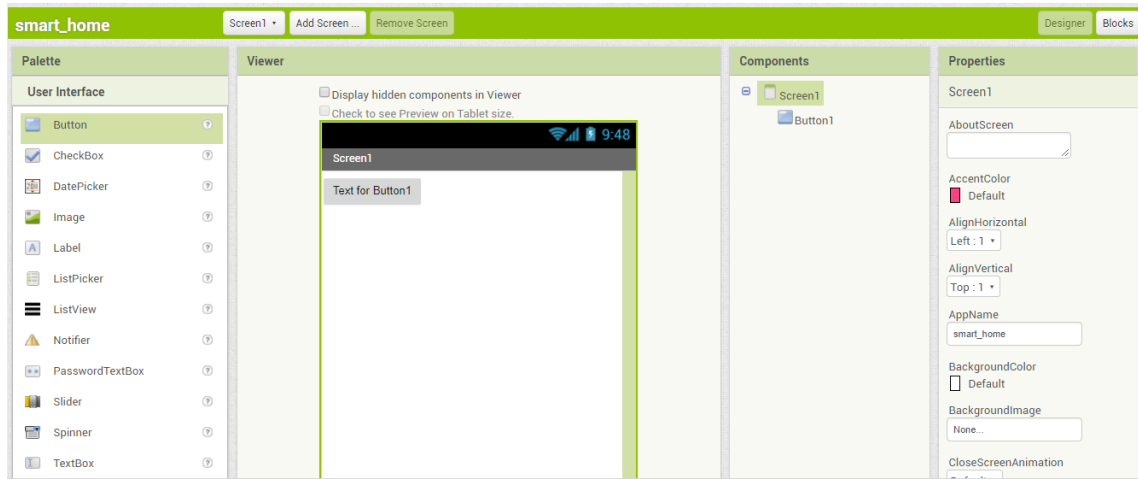
- ✓ Une pour la création de l'interface homme machine : permet de créer l'allure de l'application (App Inventor Designer) ;
- ✓ Une pour la programmation par elle-même : elle permet, par l'assemblage des blocs de créer le comportement de l'application (App Inventor Block Editor) ;
- ✓ Une pour l'émulateur : qui permet de remplacer un terminal réel pour vérifier le bon Fonctionnement du programme.

La connexion d'un terminal réel sous Androïde permettra ensuite d'y télécharger le programme pour un test réel. Ce terminal pourra aussi bien être un téléphone qu'une tablette ; le comportement du programme sera identique. [25]

### III.5.2. Structure d'une application App Inventor

Une application développée sous App Inventor est constituée de deux parties distinctes mais étroitement liées.

- A. L'interface graphique** : Pour créer l'application sous App Inventor l'interface graphique contient nos propriétés (taille, couleurs, position, textes).



**Figure III.14 :** Première interface de la création App Inventor

Cette partie contient des composants (visibles ou invisibles). [25]

Cette interface graphique contient quatre parties :

**Partie 1 :** Une palette sous App Inventor contenant tous les éléments qui peuvent être positionnés sur l'écran du téléphone.

**Partie2 :** C'est la surface du téléphone ajusté automatiquement par App inventer ou manuellement par nous-même en utilisant le composant « Screene arrangement ».

**Partie 3 :** La liste des éléments et des médias utilisés sur l'écran.

**Partie 4 :** Les propriétés des différents éléments utilisés par exemple la couleur et la taille du bouton ou texture.

Après l'assemblage des différents composants qui constituent notre application, on peut résumer cette phase « Scratch » par la figure qui suit.



**Figure III.15 :** Schéma global du Screene 1

## B. Editeur de blocs (Fenêtre Scratch)

Une fois les composants de l'écran de téléphone mis en place et désigné, nous passons à la deuxième phase de développement d'une application via App Inventor : l'interface Scratch, pour cela, il faut cliquer sur «Open the Blocks Editor» en haut à droite de la page. L'interface Scratch permet d'imbriquer des éléments graphiques entre eux pour effectuer la partie programmation de l'application à développer. De cette partie, on peut assembler les différents blocs de l'application et indiquer comment les composants doivent se comporter et qui s'affichent dans l'émulateur virtuel. [25]

Pour ouvrir ce Block, on a cliqué en haut à gauche sur « Open the Blocks Editor ». App Inventor, lancé le Blocks Editor depuis le serveur du MIT et proposé l'exécution du fichier App Inventor For Android Code blocks.

Comme indiqué sur la figure (III.16) App Inventor, éditeur de blocks, cette interface est très et épurée. En effet, en haut ; on retrouve des éléments classiques «Save», «Undo» et «Redo» ainsi qu'un bouton de test pour lancer l'application sur le mobile ou sur l'émulateur.

Cette interface contient des parties Built-in et blocks.

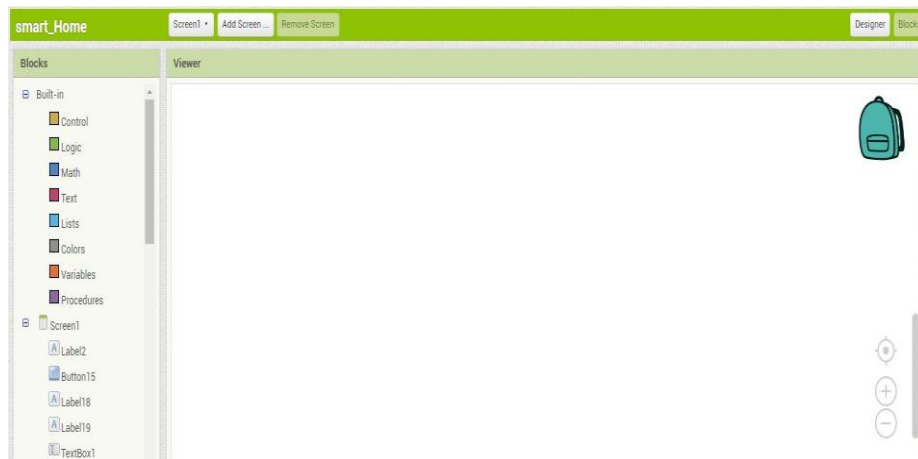


**Figure III.16 :** En-tête d'éditeur de blocks App Inventor

La fenêtre à gauche contient les blocs qui assemblent la partie droite de la fenêtre pour décrire le comportement de notre application. Les blocs peuvent être standards (dans l'onglet « Built-in ») ou définis spécifiquement pour l'application (dans l'onglet « Mye Blocks »). Aussi, la corbeille est utilisée pour jeter des morceaux de pseudocode et la loupe sert à changer la taille l'écran de l'éditeur.

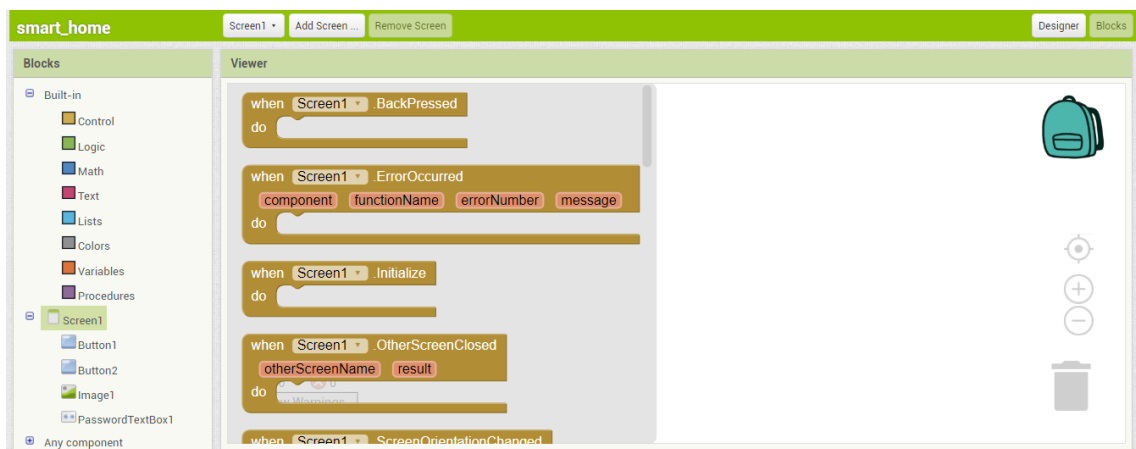
Chaque élément contient des blocks : (ces pages décrivent les blocs que nous pouvons utiliser lorsque nous construisons nos applications App Inventor. Dans l'onglet « Built-in ».

Lorsqu' on clique sur «My Blocks» en haut et à droite de la page : nous obtenons la figure suivante :



**Figure III. 17 :** Editeurs de blocks App Inventor

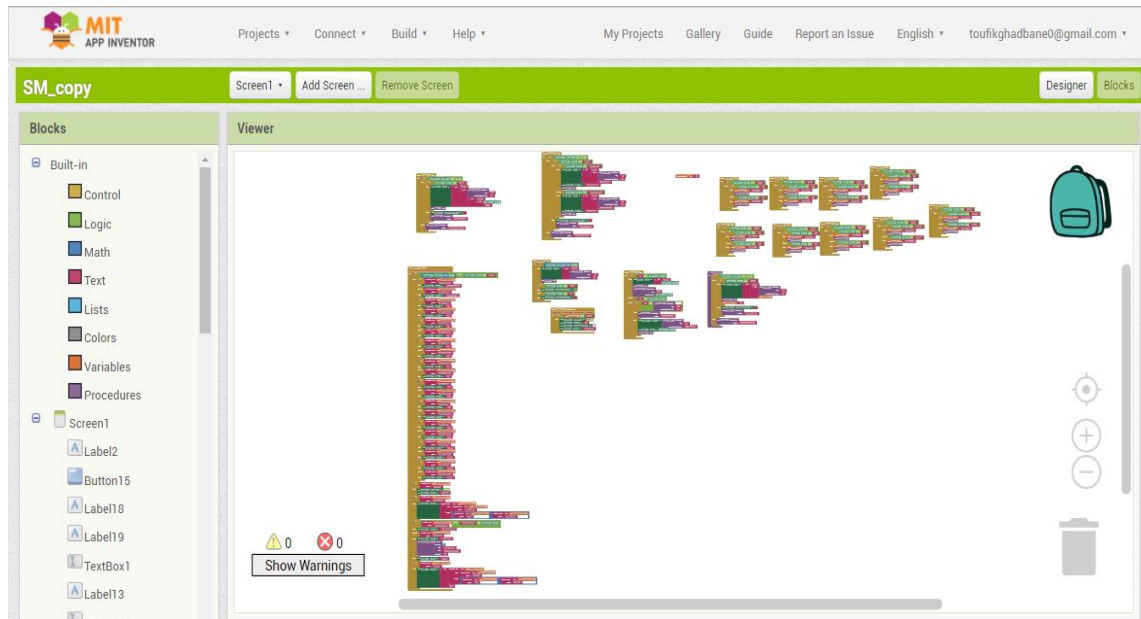
La palette des variables et fonctions est à gauche, l'onglet My blocks propose les fonctions associées aux éléments déposés sur notre écran au préalable. Dans l'onglet «My Blocks», Lorsque on Clique sur l'élément « Button1 » on obtient la figure suivante est cela comme échantillon pour n'importe quel composant associé dans notre application.



**Figure III.18 :** Echantillon d'un composant sous App Inventor

A partir de l'onglet «Package for phone» ce test assure trois solutions accessibles :

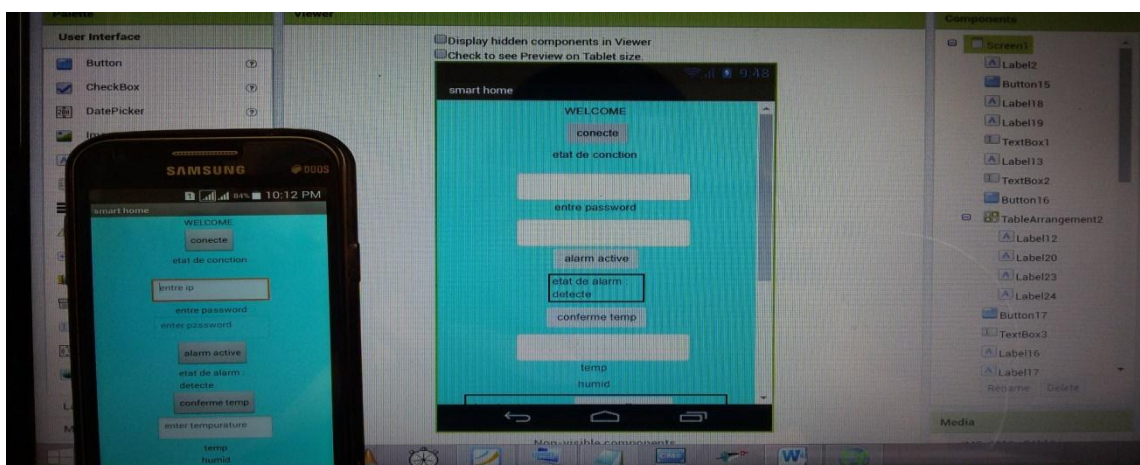
1. L'émulateur : un écran s'affichera sur l'ordinateur.
2. la connexion se fera directement sur le smart phone en wifi.
3. USB : la connexion se fera sur le smart phone via un câble USB



**Figure III.19** : Schéma global du Scratch

### III.6. Explication et démarche

Notre interface contient une bouton afficher l'état de connexion donc il faut d'abord s'assurer que le smart phone est connecté. Un mot de passe pour raison de protection a été programmé afin d'exécuter un ordre quelconque comme contrôler l'éclairage de chaque pièce, ouvrir le garage et la porte tout par un simple appui sur un bouton. En peut lire la température, humidité et l'état de l'alarme (la détection de gaz et le mouvement). Cette figure montre que notre application nommée «smart home» peut être installée sur smart phone.



**Figure III.20** : Photo réelle de notre application

### **III.6.1.Présentation des fonctions de système**

#### **III.6.1.1.Fonction de point d'accès (Passerelle)**

Pour se connecter au réseau local du système domotique via le module Wifi ESP8266, on a besoin d'une passerelle pour que l'utilisateur puisse accéder à l'application de commande.

#### **III.6.1.2.Fonction de la sécurité**

Cette fonction permet de détecter s'il y a une intrusion à l'aide du capteur de mouvement (PIR) qui lance une alarme pour informer l'utilisateur, en affichant sur l'application de commande la détection de mouvement et aussi envoyé un appel d'alerte sur la Smartphone à l'aide du module GSM SIM800L. L'accès sécurisé à l'habitat est assuré en introduisant le code correct par le smart phone.

#### **III.6.1.3.Fonction d'ouverture du garage et la porte principale**

La commande d'ouverture du portail du garage et la porte principale est réalisée à distance via l'application de commande en agissant sur le contrôle du servomoteur pour faire ouvrir/fermer le portail du garage et la porte principale.

#### **III.6.1.4.Fonction d'éclairage**

La fonction d'éclairage est assurée via l'application par les boutons ON/OFF qui va être commandée à travers le module relais qui commande les lampes de l'éclairage.

#### **III.6.1.5.Fonction de l'acquisition de la température et d'humidité**

La fonction de l'acquisition de la température et d'humidité est réalisée via le capteur DHT22 par la suite les valeurs seront affichées dans le smart phone.

#### **III.6.1.6.Fonction de ventilation**

La ventilation de l'intérieur de l'habitat est assurée par l'intermédiaire de l'application ou automatiquement vis-à-vis la variation de la température (déclaration d'un seuil de température pour lancer la ventilation)

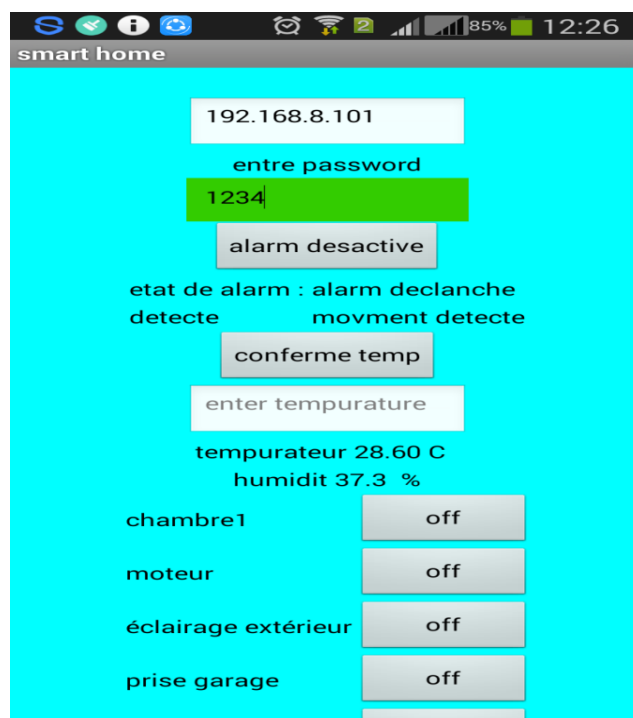
### III.6.1.7.Fonction de détection de gaz/fumée

Cette fonction permet de détecter s'il y a des fuites de gaz dans la cuisine à l'aide du capteur MQ-2 et lancer une alarme pour informer l'utilisateur en affichant sur l'application de commande la détection de gaz et aussi envoient un message d'alerte sur la Smartphone.

### III.6.2.Présentation de l'interfaçage Arduino-PC

L'interfaçage permet de faire une connexion entre notre carte Arduino UNO et le PC à travers le module Wifi ESP8266 et commander par le smart phone.

Le programme de gestion des données permet d'afficher toutes les valeurs en temps réel et chaque 5 secondes sur une fenêtre du moniteur sérié (Voir figure III.21et III.22). Dans les annexes vous allez trouver le programme exécuté dans Arduino uno et le module Wifi ESP8266 (voir l'annexe programme 01et 2).



**Figure III.21:** Image réel de l'interface de l'application de commande.

```
mov detecte
Call #
Calling 0669422800
    ---> ATD0669422800;
    <--- OK
Sent!
alarm-declanche
tempurateur 28.60
humidity 37.00
tempurateur 28.60
humidity 37.00
mow acune tempurateur 28.60
humidity 37.10
tempurateur 28.60
humidity 36.80

BUSY
alarm repose mow acune tempurateur 28.60
humidity 37.10
tempurateur 28.60
humidity 37.30
tempurateur 28.60
humidity 37.50
alarm repose    ---> ATH0
    <--- OK
OK!
tempurateur 28.60
humidity 37.40
alarm-active
alarm repose    ---> ATH0
    <--- OK
```

**Figure III.22:** Fenêtre d'affichage des résultats de système sur le moniteur sérié.

### III.7.Conclusion

Au début de ce chapitre, nous avons commencé par un schéma synoptique détaillé, ensuite donné une description et des caractéristiques de toutes les shields utilisées, puis nous avons fourni une explication détaillée de la méthode de conception et réalisation de la maison intelligente et l'application androïde.

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

Dans ce projet nous nous sommes concentrés à la conception et la réalisation d'un système de commande à distance pour commander des installations électriques pour la domotique, utilisant un protocole de communication sans fil (wifi) avec un Smartphone.

Nous avons réalisé un système de mesure en temps réel de l'ensemble des phénomènes physiques à base d'une carte Arduino UNO comme une unité de commande, le rôle de la carte Arduino UNO est de traiter les données délivrées par les capteurs et commander des différents actionneurs utilisés.

Au début on a essayé de relier le système au support d'affichage (PC) par un câble USB pour assurer le bon fonctionnement des capteurs. Le programme écrit sur IDE Arduino permet d'afficher les résultats sur le moniteur série.

En deuxième lieu, nous avons développé une interface de commande sous Androïde (une application sur Smart-phone) avec l'environnement MIT App Inventor. Nous avons préparé les icônes et les labelles des commandes pour permettre au système de superviser les résultats attendus programmés dans la carte Arduino.

La troisième partie de ce travail consiste à faire la réalisation pratique de notre système. Pour connecter le système au réseau internet on a utilisé la connexion sans fil (par wifi) à travers le module ESP8266. Pour la visualisation des grandeurs physiques mesurées on a exécuté un programme capable d'afficher les résultats en temps réel sur l'application de Smart-phone.

Une telle réalisation n'est pas dénuée de difficultés. Il est à noter que nous nous sommes confrontés à plusieurs problèmes surtout dans la partie de la connexion sans fil. Cependant, on peut dire que malgré ces difficultés, les résultats obtenus à travers cette étude qu'ils soient pratiques ou théoriques, permettent d'ouvrir la porte à d'autres études. Nous espérons que ce mémoire sera une référence aux personnes désirant développer et réaliser des projets et systèmes à base d'une carte Arduino.

Ce travail de réalisation ouvre la voie à de plusieurs perspectives dans le domaine de la commande des systèmes embarqués et de la communication des données. Nous suggérons de: a) Faire la conception et la réalisation d'un robot mobil autonome à base

d'un petit panneau solaire gérée par une carte Arduino avec orientation automatique de la direction du panneau vers le soleil, b) Réaliser un système de vidéo surveillance dans les laboratoires pédagogiques et de recherches universitaires ou dans les maisons intelligentes, c) Optimiser les systèmes de commande par Arduino pour la domotique sans fils.

## Bibliographie

- [1] Krama. A, Gougui. A, "Etude et réalisation d'une carte de contrôle par Arduino via le système Androïde", Mémoire Master Académique, Université KasdiMerbah Ouargla, Algérie, 2015.
- [2] HAMOUCHI. H, "Conception et réalisation d'une centrale embarquée de la domotique «Smart Home »", Mémoire Master, Université Mohammed V de Rabat, Rabat, 2015.
- [3] Yacine. Y, "Minimisation d'énergie dans un réseau de capteur", Mémoire de Magister, Université Mouloud Mammeri de TiziOuzou, Algérie, 2012.
- [4] Serge Darrieumerlou, " le guide de la maison et des objets connectés", Edition Eyrolles,2016.
- [5] <http://www.cea.fr/comprendre/Pages/nouvelles-technologies/essentiel-sur-domotique-maison-connectee>.
- [6] Mr: METAHRIM, Melle ABDELLIS, " Smart House ", Mémoire Master, Université ABOU BEKR BELKAID de Tlemcen, Algérie, 2017.
- [7] Boudellal. M, "Smart home - Habitat connecté, 361 installations domotiques et multimédia". Dunod, 2014.
- [8] [http://www.cisco.com/c/dam/global/en\\_ca/solutions/executive/assets/pdf/internet-of-things-fr.pdf](http://www.cisco.com/c/dam/global/en_ca/solutions/executive/assets/pdf/internet-of-things-fr.pdf), Avril 2017.
- [9] Ming. Y, "Cobalt oxide nano sheet humidity sensor integrated with circuit on chip" Microelectronic Engineering, 2011.
- [10] Astalaseven, Eskimon et Olyte, "Arduino pour bien commencer en électronique et en programmation", Licence Créative Commons BY-NC-SA 2.0.
- [11] Lechalupé. J, "cours d'initiation à Arduino", Université Paul Sabatier, Mai 2014.
- [12] Cottenceau. B, "Carte ARDUINO UNO Microcontrôleur ATmega328".
- [13] <https://www.adafruit.com/product>.
- [14] <https://drotek.com/shop/fr/capteurs/326-capteur-de-distance-srf05.html>.
- [15] <https://fr.aliexpress.com/item/20Pcs-Photosensitive-brightness-resistance-sensor-module>.
- [16] <https://www.gotronic.fr/art-detecteur-de-niveau-d-eau-grove-101020018-19048.htm>.

- [17] Lechalupé. J, "cours d'initiation à Arduino", Université Paul Sabatier, Mai 2014.
- [18] <https://fr.wikipedia.org/wiki/Fritzing>.
- [19] <http://www.arobose.com/shop/temperature-pression/217-capteur-analogique-de-gaz-mq2.html>.
- [20] <https://www.generationrobots.com/fr/402273-capteur-pir-de-mouvement.html>.
- [21] <https://letmeknow.fr/blog/2015/10/14/tuto-module-gsm-sim800l-prise-en-main>.
- [22] <https://fr.wikipedia.org/wiki/Servomoteur#D%C3%A9finition>.
- [23] <https://sites.google.com/site/arduinoencore/esp8266>.
- [24] <https://www.sparkfun.com/products/13678>.
- [25] [sig.fgranotier.info/IMG/pdf/debuter-app\\_inventor](http://sig.fgranotier.info/IMG/pdf/debuter-app_inventor).

# Annexes

## Programme (01) avec câble USB

```

#include "DHT.h" // sensor temperature & humidity
#define DHTPIN 2 // what digital pin we're connected to
                // Connect a 10K resistor from pin 2 (data) to pin 1(power)
                // of the sensor
#define DHTTYPE DHT22 // #define DHTTYPE DHT22 (AM2302), AM2321

#include <Wire.h>
#include <Adafruit_BMP085.h>
Adafruit_BMP085 bmp; // sensor Presion
DHT dht(DHTPIN, DHTTYPE);
float temt6000Pin = 0; // sensor flux
float a;
//-----
void setup()
{
  Serial.begin(9600);
  Serial.println("DHT22 & TEMT6000 & BMP180 ");
  dht.begin();
  bmp.begin();
}
void loop()
{
  // Wait a few seconds between measurements.
  delay(2000);
  // Reading temperature or humidity takes about 250 milliseconds!
  // Sensor readings may also be up to 2 sec 'old' (its a very slow sensor)

  float h = dht.readHumidity();
  float t = dht.readTemperature(); //Read temperature as Celsius
  float f = dht.readTemperature(true); // Read temperature as Fahrenheit
(isFahrenheit = true)
  // Check if any reads failed and exit early (to try again).
  if (isnan(h) || isnan(t) || isnan(f))
  {
    Serial.println("Failed to read from DHT sensor!");
    return;
  }
  float value = analogRead(temt6000Pin);
  float a = value*100/1000;

  Serial.print("Humidity: ");
  Serial.print(h);
  Serial.println(" %\t");
  Serial.print("Temperature: ");
  Serial.print(t);
  Serial.print(" *C ");
  Serial.print(f);
  Serial.println(" *F\t");

  Serial.print("flux : ");
  Serial.print(value);
  Serial.print("\t");
  Serial.print(a);
  Serial.println("%");
}

```

```

Serial.print("Pressure = ");
Serial.print(bmp.readPressure());
Serial.println(" Pa");

    // Calculate altitude assuming 'standard' barometric
    // pressure of 1013.25 millibar = 101325 Pascal
Serial.print("Altitude = ");
Serial.print(bmp.readAltitude());
Serial.println(" meters");

    delay(2000);
}

```

## Programme (02) sans câble USB ( par WIFI )

```

#include <SoftwareSerial.h>
SoftwareSerial espSerial = SoftwareSerial(10,11); // esp8266-01 : RX to
pin=11 , TX to pin=10

//DHT22=====
#include <DHT.h>
#define DHTPIN 2 // Connect the signal pin of DHT22 sensor to digital pin 2
#define DHTTYPE DHT22
DHT dht(DHTPIN, DHTTYPE);

//temt6000=====
float temt6000Pin = 0;
float a;
//bmp180 =====
#include <Wire.h>
#include <Adafruit_BMP085.h>
Adafruit_BMP085 bmp;
//=====
String apiKey = "VK5D36WY9SAUABAM"; // replace with your channel's
thingspeak WRITE API key

String ssid="LG"; // Wifi network SSID
String password ="20091993"; // Wifi network password

boolean DEBUG=true;

//showResponse=====
void showResponse(int waitTime){
    long t=millis();
    char c;
    while (t+waitTime>millis()){
        if (espSerial.available()){
            c=espSerial.read();
            if (DEBUG) Serial.print(c);
        }
    }
}

//=====
boolean thingSpeakWrite(float value1, float value2, float value3, float
value4, float value5)
{
    String cmd = "AT+CIPSTART=\"TCP\", \""; // TCP connection
    cmd += "184.106.153.149"; // api.thingspeak.com
    cmd += "\",80";
    espSerial.println(cmd);
}

```

```

if (DEBUG) Serial.println(cmd);
if(espSerial.find("Error")){
  if (DEBUG) Serial.println("AT+CIPSTART error");
  return false;
}
String getStr = "GET /update?api_key="; // prepare GET string
getStr += apiKey;

getStr += "&field1=";
getStr += String(value1); // t
getStr += "&field2=";
getStr += String(value2); // h
getStr += "&field3=";
getStr += String(value3); // a
getStr += "&field4=";
getStr += String(value4); // pression
getStr += "&field5=";
getStr += String(value5); // altitude
// ...
getStr += "\r\n\r\n";

// send data length
cmd = "AT+CIPSEND=";
cmd += String(getStr.length());
espSerial.println(cmd);
if (DEBUG) Serial.println(cmd);

delay(100);
if(espSerial.find(">")){
  espSerial.print(getStr);
  if (DEBUG) Serial.print(getStr);
}
else{
  espSerial.println("AT+CIPCLOSE");
  // alert user
  if (DEBUG) Serial.println("AT+CIPCLOSE");
  return false;
}
return true;
}
//setup=====
void setup() {
  DEBUG=true; // enable debug serial
  Serial.begin(9600);

  dht.begin(); // Start DHT22 sensor
  bmp.begin(); // Start bmp180 sensor
  espSerial.begin(115200); // enable software serial
  // Your esp8266 module's speed is probably at 115200.
  // For this reason the first time set the speed to 115200 or to your
  esp8266 configured speed and upload. Then change to 9600 and upload again

  espSerial.println("AT+RST"); // Enable this line to reset the module
  showResponse(1000);

  //espSerial.println("AT+UART_CUR=9600,8,1,0,0"); // Enable this line to
  set esp8266 serial speed to 9600 bps
  //showResponse(1000);

  espSerial.println("AT+CWMODE=1"); // set esp8266 as client
  showResponse(1000);

```

```
    espSerial.println("AT+CWJAP=\"" + ssid + "\", \"" + password + "\""); // set your
home router SSID and password
    showResponse(5000);

    if (DEBUG) Serial.println("Setup completed");
}
//Loop =====
void loop() {
    // Read sensor values
    float t = dht.readTemperature();
    float h = dht.readHumidity();
    float value = analogRead(temt6000Pin);
    float a = value*100/1000;

    if (isnan(t) || isnan(h)) {
        if (DEBUG) Serial.println("Failed to read !");
    }
    else {
        if (DEBUG) Serial.println("Temp="+String(t)+" *C");
        if (DEBUG) Serial.println("Humidity="+String(h)+" %");
        if (DEBUG) Serial.println("flux="+String(a)+" %");
        if (DEBUG) Serial.println("Pressure =
"+String(bmp.readPressure())+" Pa");
        if (DEBUG) Serial.println("Altitude =
"+String(bmp.readAltitude())+" meters");
        thingSpeakWrite(t,h,a,bmp.readPressure(),bmp.readAltitude());
    }
    // Write values to thingspeak
    // thingspeak needs 15 sec delay between updates,
    delay(20000); //50000
}
```

## la liste des commandes ESP8266

### ESP8266 AT Command Set

Function	AT Command	Response
Working	AT	OK
Restart	AT+RST	OK [System Ready, Vendor:www.ai-thinker.com]
Firmware version	AT+GMR	AT+GMR 0018000902 OK
List Access Points	AT+CWLAP	AT+CWLAP +CWLAP:{4,"RochefortSurLac",-38,"70:62:b8:6f:6d:58",1} +CWLAP:{4,"LiliPad2.4",-83,"f8:7b:8c:1e:7c:6d",1} OK
Join Access Point	AT+CWJAP? AT+CWJAP="SSID","Password"	Query AT+CWJAP? +CWJAP:"RochefortSurLac" OK
Quit Access Point	AT+CWQAP=? AT+CWQAP	Query OK
Get IP Address	AT+CIFSR	AT+CIFSR 192.168.0.105 OK
Set Parameters of Access Point	AT+ CWSAP? AT+ CWSAP= <ssid>,<pwd>,<chl>, <ecn>	Query ssid, pwd chl = channel, ecn = encryption
WiFi Mode	AT+CWMODE? AT+CWMODE=1 AT+CWMODE=2 AT+CWMODE=3	Query STA AP BOTH
Set up TCP or UDP connection	AT+CIPSTART=? (CIPMUX=0) AT+CIPSTART = <type>,<addr>,<port> (CIPMUX=1) AT+CIPSTART= <id><type>,<addr>, <port>	Query id = 0-4, type = TCP/UDP, addr = IP address, port= port
TCP/UDP Connections	AT+ CIPMUX? AT+ CIPMUX=0 AT+ CIPMUX=1	Query Single Multiple
Check join devices' IP	AT+CWLIF	
TCP/IP Connection Status	AT+CIPSTATUS	AT+CIPSTATUS? no this fun
Send TCP/IP data	(CIPMUX=0) AT+CIPSEND=<length>; (CIPMUX=1) AT+CIPSEND= <id>,<length>	
Close TCP / UDP connection	AT+CIPCLOSE=<id> or AT+CIPCLOSE	
Set as server	AT+ CIPSERVER= <mode>[,<port>]	mode 0 to close server mode; mode 1 to open; port = port
Set the server timeout	AT+CIPSTO? AT+CIPSTO=<time>	Query <time>0~28800 in seconds
Baud Rate*	AT+CIOBAUD? Supported: 9600, 19200, 38400, 74880, 115200, 230400, 460800, 921600	Query AT+CIOBAUD? +CIOBAUD:9600 OK
Check IP address	AT+CIFSR	AT+CIFSR 192.168.0.106 OK
Firmware Upgrade (from Cloud)	AT+CIUPDATE	1. +CIUPDATE:1 found server 2. +CIUPDATE:2 connect server 3. +CIUPDATE:3 got edition 4. +CIUPDATE:4 start update
Received data	+IPD	(CIPMUX=0): + IPD, <len>; (CIPMUX=1): + IPD, <id>, <len>; <data>
Watchdog Enable*	AT+CSYSWDTENABLE	Watchdog, auto restart when program errors occur: enable
Watchdog Disable*	AT+CSYSWDTDISABLE	Watchdog, auto restart when program errors occur: disable

\* New in V0.9.2.2 (from <http://www.electrodragon.com/w/Wi07c>)

## Résumé

Durant ces dernières années la conception des systèmes embarqués a connu une évolution remarquable, cette révolution prodigieuse de l'électricité a conduit à la conception des systèmes de commande plus en plus complexes, offrant certaines avantages évidentes en ce qui concerne la commande des systèmes à distance, à partir de notre Smartphone, par notre voie ...etc.

Le but principal de ce travail est la réalisation d'un système de commande à distance (sous réseau Internet) à base d'un Arduino UNO pour commander des installations électriques pour la domotique, comme par exemple : des interrupteurs, des électrovannes, des PCs, Téléviseur ....etc.

**Mots clés :** Commande à distance, Arduino UNO, Réseau Ethernet WLAN, Domotique.

## ملخص

في السنوات الأخيرة، شهد تصميم الأنظمة المدمجة تطوراً ملحوظاً، وقد أدت هذه الثورة الكهربائية الهائلة إلى تصميم أنظمة تحكم معقدة بشكل متزايد، مما يوفر بعض المزايا الواضحة فيما يتعلق بالتحكم عن بعد في الأنظمة الكهربائية، من الهاتف الذكي لدينا، بالصوت... الخ

الهدف الرئيسي من هذا العمل هو تحقيق نظام التحكم عن بعد (باستعمال الشبكة العنكبوتية) على أساس نظام Arduino UNO للتحكم في التركيبات الكهربائية للتشغيل الآلي للمنزل، مثل: المحولات، الصمامات الملف اللولبي، أجهزة الكمبيوتر والتلفزيون .... الخ

الكلمات المفتاحية: التحكم عن بعد، اردوينو اينو، شبكة إيثرنت WLAN, domotique المنزل.

## Summary

In recent years the design of embedded systems has undergone a remarkable evolution, this prodigious electric revolution has led to the design of increasingly complex control systems, offering some obvious advantages with regard to the control of remote systems, from our smartphone, by our voice... etc.

The main goal of this work is the realization of a remote control system (using Internet network) based on an Arduino UNO to control electrical installations for home automation, such as: switches, solenoid valves, PCs, TV.... etc.

**Keywords:** Remote Control, Arduino UNO, WLAN Ethernet Network, Home Automation.