



**LICENCE SCIENCES ET TECHNIQUES
Génie Electrique**

RAPPORT DE FIN D'ETUDES

Intitulé :

**Conception et réalisation d'une
Maison intelligente**

Réalisé Par :

**Souhail M'hid
Ahmed Mabrouk**

Encadré par :

P^r Mme. ERRAHIMI FATIMA (FST FES)

Soutenu le 8 Juin 2018 devant le jury

Pr Mr. Ali Ahaitouf (FST FES)

Pr Mr. Mhamed Lahbabi (FST FES)

Pr Mme. Najia Es Sbai (FST FES)

Dédicace

Je dédie ce modeste travail à :

Ma chérie Mère, mon cher père et mes frères pour leur soutien, affection et amour, leur confiance et patience et pour leurs sacrifices infinis.

Mes amis, qui représentent pour moi tous le sens de la sincérité et de la fidélité, et avec qui j'ai passé des moments inoubliables.

Mes professeurs, qui m'ont dirigé vers le chemin du succès, leur compréhension et leurs conseils m'ont permis de mieux apprécier la formation disposée au sein de la Faculté

Et à tous ceux qui ont contribué de près ou de loin à la réalisation de ce travail et à la élaboration de ce rapport.

Remerciement

Avant d'entamer la rédaction de ce rapport, il nous est agréable de présenter une dette de reconnaissance à toutes les personnes qui, par leur intervention ont favorisé l'aboutissement de ce projet de fin d'étude.

Nous tenons à remercier infiniment notre encadrante **Madame Errahimi Fatima**, qui nous a informé et accompagné tout au long de cette expérience professionnelle avec beaucoup de patience et de pédagogie. Nous sommes très reconnaissants à **madame Errahimi** de nous avoir donné l'opportunité de travailler sur un projet qui était tellement bénéfique. Nous la remercions pour tous ses efforts et ses conseils précieux et son soutien pour réussir un tel sujet et de nous avoir guidé dans l'élaboration de ce travail et contribuer largement à sa réalisation.

Nous tenons aussi à exprimer nos gratitude et nos remerciements aux membres du jury. Veuillez accepter dans ce travail nos sincères respects et nos profondes reconnaissances.

Table des matières

Liste des figures	4
Liste des tableaux	5
Liste des annexes	5
Introduction générale.....	6
Chapitre 1 : La domotique	7
I. Introduction	7
II. Définitions	7
III. Domaines de la domotique.....	7
1. Protection des personnes et des biens	8
2. Confort de la vie quotidienne	8
3. Les économies d'énergie.....	9
IV. La maison communicante	9
1. Techniques de mise en place d'une maison connectée	10
2. Techniques de liaison	10
V. Étude des protocoles de communication sans fil.....	11
1. Principe de la communication sans fil	11
2. Étude des différents protocoles de communication	11
VI. Conclusion.....	12
Chapitre 2 : Description du projet.....	13
I. Introduction	13
II. Présentation du cahier des charges.....	13
III. Présentation des fonctions de domotique proposées	15
1. Fonction de gestion d'éclairage :.....	15
2. Fonction de gestion d'ouverture des volets des fenêtres :.....	15
3. Fonction d'ouverture de la porte principale :.....	15
4. Fonction d'acquisition de la température et ventilation :.....	15
5. Fonction de détection de fuite de gaz :.....	15
Chapitre 3 : Étude de la partie matérielle et logicielle du projet	16
I. Introduction	16

II.	Carte Arduino	16
1.	Arduino Méga 2560	17
a.	Caractéristiques :.....	17
b.	Mémoire.....	18
c.	Entrées et sorties numériques.....	18
d.	Broches analogiques.....	18
e.	Autres broches.....	18
f.	Communications.....	18
2.	Pourquoi Arduino ?	19
III.	Choix des éléments du système domotique	19
1.	Module Afficheur LCD	19
a.	Connecteur de l’afficheur LCD.....	20
b.	Communication avec le LCD.....	20
c.	Connexion avec la carte Arduino Méga 2560.....	20
d.	Réglage du contraste.....	20
e.	Pilotage via le bus I2C.....	21
2.	Télécommande infrarouge	21
3.	Clavier matriciel	22
a.	Principe du clavier matriciel.....	22
b.	Détection des touches.....	22
c.	Branchement avec Arduino Méga 2560.....	22
4.	Ventilateur	23
5.	Capteur de température DS18B20	23
6.	Capteur de la lumière : la photorésistance (LDR)	24
7.	Capteur de mouvement (PIR)	25
8.	Capteur de gaz/fumée (MQ-2)	26
9.	Module Buzzer	26
10.	Moteur à courant continu	27
a.	La mécanique liée au moteur :.....	27
b.	Alimentation du moteur :.....	28
11.	Module Bluetooth HC-05	30
IV.	Étude de la partie logicielle	31

1. Plateforme de programmation Arduino	31
2. Plateforme de Logiciel LabVIEW	33
a. Fonctionnement :	34
b. Langage utilisé :.....	34
c. Contrôle à distance en utilisant LabVIEW :	35
V. Conclusion	35
Chapitre 4 : Réalisation du système domotique	36
I. Construction de la maison	36
II. Développement de l'application de commande	37
1. Détection de mouvement.....	37
2. Fonction d'accès sécurisé à l'habitat	37
3. Fonction d'ouverture des volets de la fenêtre	37
4. Fonction d'éclairage	38
5. Fonction de l'acquisition de la température.....	38
6. Fonction de détection de gaz/fumée	39
7. Fonction de ventilation.....	39
III. Centralisation des commandes : Interface	39
IV. Conclusion	40
Conclusion générale	41
Annexes	42
Bibliographie	50

Liste des figures

Figure 1 : Schéma globale du système domotique à concevoir	14
Figure 2 : Exemples des cartes Arduino	16
Figure 3 : Arduino Méga 2560.....	17
Figure 4 : Afficheurs LCD (16x2).....	19
Figure 5 : Connecteur de l'afficheur LC	20
Figure 6: Montage du Module I2C avec l'afficheur LCD	21
Figure 7 : télécommande infrarouge avec récepteur.....	21
Figure 8 : Clavier matriciel (4*4) et son schéma de principe	22
Figure 9 : Branchement du clavier avec l'Arduino.....	22
Figure 10 : Ventilateur 5V d'un PC.....	23
Figure 11 : Capteur DS18B20 en boîtier TO-92	23
Figure 12 : Capteur DS18B20 en format "sonde" étanche	24
Figure 13 : Capteur de la lumière.....	24
Figure 14 : Courbe de sensibilité en fonction de la longueur d'onde de la lumière	25
Figure 15 : Capteur de mouvement (PIR)	25
Figure 16 : Capteur de gaz/fumée MQ-2	26
Figure 17 : Branchement du MQ-2 avec Arduino	26
Figure 18 : le buzzer	26
Figure 19 : moteur à CC.....	27
Figure 20 : moteur + réducteur	28
Figure 21 : les pins de L293D	29
Figure 22 : brochage de deux moteurs avec l'Arduino	29
Figure 23 : module Bluetooth HC-05.....	30
Figure 24 : le logiciel Arduino	31
Figure 25 : Interface de la plateforme Arduino	32
Figure 26 : Barre de boutons Arduino.....	32
Figure 27 : HyperTerminal de l'Arduino (Moniteur Série)	32
Figure 28 : Structure d'un programme en Arduino	33
Figure 29 : front panel (page d'interface).....	34
Figure 30 : block Diagram.....	34
Figure 31 : Configuration du Web Server	35
Figure 32 : Photos de notre maison	36
Figure 33 : système de sécurité.....	37
Figure 34 : Commande d'éclairage.....	38
Figure 35 : l'acquisition de la température	38
Figure 36 : page HTML de commande à distance via une adresse IP	39
Figure 37 : application Android de commande à distance via Bluetooth	40

Liste des tableaux

Tableau 1 : Comparaison entre les différents protocoles	12
Tableau 2 : commande du L293D.....	29

Liste des annexes

Annexe 1 : Brochage de la carte Arduino Méga 2560	42
Annexe 2 : Organigramme de la fonction de sécurité	43
Annexe 3 : Code Arduino de la fonction d'accès sécurisé à l'habitat.....	44
Annexe 4 : Code d'acquisition de la température et de la détection de fuite de gaz.....	45
Annexe 5 : Code Arduino de l'éclairage automatique des 3 pièces aussi de l'éclairage manuel à partir de l'application HTML	46
Annexe 6 : Code Arduino de commande de l'application Android.....	48
Annexe 7 : Code Arduino de contrôle à distance via la télécommande infrarouge	49

Introduction générale

L'évolution de la technologie et du mode de vie nous permet aujourd'hui de prévoir des logements mieux adaptés. De même, La majorité des individus, et plus particulièrement les personnes âgées ou handicapées, 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.

La domotique ou encore la maison intelligente est définie comme une résidence équipée de technologies de l'électronique, de l'automatique, de l'informatique et des télécommunications permettant d'améliorer le confort, la sécurité, la communication et la gestion d'énergie d'une maison ou d'un lieu public. Elle assure différentes fonctions : La fonction de confort, la fonction d'économie d'énergie et la fonction de sécurité.

La domotique repose sur trois principes : La liaison entre les appareils, la communication entre l'utilisateur et les appareils et l'automatisation.

Dans ce travail, nous proposons de concevoir un modèle réduit d'une maison dite « intelligente ». Cette maquette permettrait d'implémenter des fonctions de domotique à savoir : La gestion d'éclairage, l'acquisition et le réglage de la température à l'intérieur de l'habitat, l'ouverture et la fermeture de la fenêtre et des portes puis la détection de mouvement, de fuite de gaz et de la fumée. Ces scénarios seront automatisés via des cartes « Arduino » exécutant des programmes informatiques et aussi via une application Android, une télécommande infrarouge et à distance via l'interface graphique avec le logiciel LabVIEW.

Le rapport est organisé en quatre chapitres, le premier fait objet d'une présentation générale de la domotique ainsi que ses applications et les différents types de technologies utilisées. Le second chapitre est dédiée à la description du projet ; la présentation du cahier des charges et les enjeux envisagés pour notre système domotique. Le troisième chapitre est consacré à la description de la partie matérielle et logicielle du projet, les composantes de notre solution vont être détaillées, les applications et modules basée sur la technologie Arduino et le logiciel LabVIEW. Le quatrième chapitre est consacré à la conception de notre projet Maison intelligente. Nous détaillerons les phases de la mise en place du système domotique, sa construction et les organigrammes décrivant le raisonnement du programme de commande implémenté sur la carte Arduino Méga 2560.

Chapitre 1

La domotique

I. Introduction

Les premiers développements de la domotique sont apparus dans les années 1980 grâce à la réduction des systèmes électroniques et informatiques. Dès lors, l'industrie a concentré ses expérimentations sur le développement d'automates, d'interfaces et d'outils apportant confort, sécurité et assistance au sein d'un édifice. Ce chapitre est consacré à des généralités sur la domotique, les maisons connectées et plus notamment les protocoles de communication sans fil.

II. Définitions

Le mot domotique a été introduit dans le dictionnaire « le petit Larousse » en 1988. Ce mot vient de « Domus », le domicile en latin, associé au suffixe « tique » qui fait référence à la technique.

La domotique regroupe donc l'ensemble des technologies de l'électronique, de l'informatique et des télécommunications qui sont utilisées dans les domiciles pour rendre ceux-ci plus « intelligents ». Elle vise à intégrer différents systèmes pouvant être connectés entre eux et à des réseaux internes et externes de communication. Parmi ces fonctions, on trouve notamment l'économie et la gestion d'énergie, l'information et la communication, la maîtrise du confort, la sécurité et l'assistance.

Câblée ou fonctionnant par ondes radio, la domotique investit notre univers quotidien pour nous faciliter la vie. Dans ses applications les plus évoluées, la domotique met en réseau et coordonne le fonctionnement de différents types d'équipements ménagers, de travail et de loisir.

Les applications possibles de la domotique concernent aussi bien la programmation, la surveillance, que le contrôle à distance.

III. Domaines de la domotique

Les services offerts par la domotique couvrent trois domaines principaux :

- Assurer la protection des personnes et des biens en domotique de sécurité.
- Veiller au confort de vie quotidien des personnes âgées, entre autres, en installant une domotique pour les personnes à mobilité réduite.
- Faciliter les économies d'énergie grâce à la réactivité maîtrisée d'une maison intelligente.

1. Protection des personnes et des biens

La domotique permet le suivi des personnes âgées ou handicapées. En matière de sécurité domestique. Alarmes, détecteurs de mouvement ou d'intrusion, interphones et portiers vidéo, téléphones, simulateurs de présence, etc. D'autres systèmes de détection sont prévus pour surveiller les enfants, prévenir les risques d'accident (incendie, fuite de gaz, etc.) et signaler des pannes (inondation, coupure de courant électrique, etc.).

La domotique de sécurité passe également par la centralisation de la surveillance et du contrôle de toutes les zones de la maison. Des capteurs de mouvements, de bris de glace, d'ouverture, etc., des poignées biométriques, l'automatisme des volets... sont installés sur les ouvertures et préviennent de toute intrusion, car l'ensemble est couplé à des alarmes silencieuses sans fil ou des sirènes. Pour l'intérieur des pièces, des micros ultrasensibles, des caméras invisibles, des champs magnétiques, des détecteurs de fumées assurent aussi une grande sécurité s'ils sont judicieusement positionnés.

2. Confort de la vie quotidienne

Toutes les actions que nous faisons machinalement peuvent être automatisées et intégrées dans des scénarios préprogrammés. L'élimination des gestes fastidieux et répétitifs peut nous faire gagner du temps, économiser l'énergie et nous tranquilliser l'esprit. Parmi ces scénarios préprogrammés, on peut citer :

- **La régulation en fonction de la luminosité extérieure** : un capteur de luminosité peut être installé pour piloter l'éclairage en fonction d'un seuil prédéfini ou le réguler de façon continue afin d'obtenir une luminosité constante. Les éclairages s'allument, s'éteignent ou s'ajustent alors en variation pour optimiser les conditions de luminosité.
- **La commande d'éclairage** : le capteur de présence permet de déclencher automatiquement un éclairage quand nous passons devant un garage, couloir, dressing, cave, etc. La minuterie permet d'interrompre un circuit après un laps de temps déterminé et la télécommande permet d'allumer la lumière depuis un canapé.
- **La programmation quotidienne et hebdomadaire** : Nos rythmes de vie étant relativement réguliers, il est simple d'associer automatiquement un niveau de chauffage à différentes périodes de la journée ou de la semaine. Un chauffage qui passe du mode réduit au mode confort quelques minutes avant notre réveil puis bascule à nouveau dans la journée, sauf pendant le week-end, et s'ajuste en fonction de la température extérieure pour nous accueillir

au retour du travail contribue de façon efficace et simple à notre bien-être tout en nous permettant de maîtriser les dépenses énergétiques.

- **Simplification de la vie :** Le matin, nous nous réveillons en douceur : notre réveil, qui s'est coordonné avec notre agenda, sonne, tandis que les volets s'ouvrent tous seuls, et que le chauffage tourne déjà dans la salle de bain. Au moment du départ, nous prévenons notre maison que nous partons grâce à un petit geste au moyen d'une télécommande (ou avec notre smartphone) : un programme se lance, les lumières s'éteignent, le chauffage s'arrête, les portes se verrouillent. Avant de rentrer, le soir, nous prévenons notre domicile à distance. Le chauffage sera à la bonne température lorsque nous arriverons, et quand nous rentrerons et sans descendre de notre voiture, nous désactivons l'alarme, ouvrons le portail, éclairons l'allée si nécessaire et ouvrons la porte du garage au moyen d'une télécommande.

3. Les économies d'énergie

La domotique permet de diminuer des factures d'énergie. Grâce aux automatismes et à des capteurs, les équipements électriques connectés pilotent la consommation énergétique (chauffage, éclairage, eau, ventilation, etc.), tout en gardant sous contrôle le confort des zones occupées.

Les systèmes de régulation permettent de maîtriser la consommation d'électricité, de gérer le chauffage et la production d'eau chaude sanitaire, avec un niveau de confort optimal. Un détecteur de présence placé dans chaque pièce, par exemple, commande instantanément l'allumage ou l'extinction des éclairages, la mise en route ou l'arrêt du chauffage.

Au jardin par exemple, l'arrosage s'automatise et le détecteur se charge d'allumer les lumières dès la tombée de la nuit et de lancer l'irrigation des plantes. La maison intelligente utilise la programmation domotique via des scénarios qu'on peut déterminer en fonction des besoins spécifiques, évitant les pertes thermiques et les risques d'oubli ou de sécurité.

IV. La maison communicante

La communication tient une place de plus en plus importante dans le logement. La maison est dotée de capteurs de tous genres, permettant la connexion avec une télécommande universelle ou un smartphone. En communiquant avec l'habitat, il est possible de régler le chauffage, de recevoir des alertes lorsque des intrus tentent de faire une intrusion, ou lorsqu'une personne âgée maintenue à domicile est en situation de détresse. Gérez les programmes des appareils de la maison n'importe où, en fonction des besoins.

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.

1. Techniques de mise en place d'une maison connectée

Un système de maison intelligente est généralement constitué des équipements suivants :

- Un cerveau, que ce soit un automate, un ordinateur, ou un « box domotique ».

C'est lui qui centralise toutes les informations de la maison et déclenche des actions.

- Les capteurs sont des périphériques permettant de relever la température, l'humidité, la luminosité, le niveau de CO₂, le niveau de bruit, détecter une présence, de la fumée, une fuite de gaz, etc... Grâce à eux la maison saura tout ce qui se passe.
- Les actionneurs sont donc des périphériques qui pilotent des appareils (radiateurs, chaudière, télévision, machine à laver...etc.), des lampes, ou encore des automatismes (volets, porte de garage, store banne, etc.).

Le cerveau déclenche des actions en fonction des informations recueillies par les différents capteurs disséminés à travers la maison. Par exemple si aucun détecteur ne détecte de présence dans la maison, le cerveau demande aux radiateurs de passer en mode économique et aux lampes de s'éteindre.

2. Techniques de liaison

Les solutions filaires relient le cerveau, ses capteurs et ses actionneurs par des câbles à travers toute la maison est difficilement envisageable pour beaucoup de monde, car le passage des câbles dans une habitation existante est une vraie contrainte. Ce type de solution est plutôt pour des constructions neuves ou de grosses rénovations.

Il existe aujourd'hui de nombreuses solutions sans fil, tout aussi fiables, mais nettement plus simples à mettre en œuvre implémentant l'un des protocoles suivants : Z-Wave, Zigbee, RTS, EnOcean, et le Io-Home control en sont les principaux exemples.

V. Étude des protocoles de communication sans fil

1. Principe de la communication sans fil

La domotique sans fil utilise plutôt les ondes électromagnétiques pour transmettre des données en utilisant l'air comme canal de transfert :

- L'émetteur applique une certaine variation de courant à son antenne.
- La variation de courant induit une onde électromagnétique.
- L'onde électromagnétique se propage à une vitesse proche de celle de la lumière dans l'air.
- Un courant électrique est induit dans l'antenne du récepteur par la variation de champs magnétique.
- Le récepteur lit la variation de courant et l'interprète selon le protocole de communication.

2. Étude des différents protocoles de communication

- **Infrarouge** : La technologie infrarouge utilise les longueurs d'ondes plus longues dans la zone rouge du spectre électromagnétique qui se situent au-delà du champ de vision humain. La technologie infrarouge est utilisée par la télécommande d'un téléviseur. Elle est peu coûteuse et facile à intégrer dans les périphériques par les fabricants, mais elle présente quelques inconvénients : Diminution des performances selon la distance, seulement deux périphériques peuvent communiquer l'un avec l'autre et blocage possible de la transmission du signal avec les obstacles (personnes, murs, plantes, etc.).
- **Bluetooth** : c'est un protocole sans fil d'échange de données sur de courtes distances entre des périphériques fixes et mobiles. Elle permet de connecter plusieurs périphériques en résolvant les problèmes de synchronisation. La norme BLUETOOTH est automatique, économique et présente une portée de transmission des données plus étendue par rapport à l'infrarouge.
- **Zigbee** : c'est une technologie sans fil radio, de basse puissance et de bas débits (en dessous de 150 kbps). Sa portée est variable. Elle oscille suivant l'environnement entre 10 et 50 mètres en intérieur. L'avantage majeur de Zigbee est que la technologie est peu consommatrice en énergie. Elle peut, de plus, être intégrée à bas coût dans les équipements. Il présente quelques inconvénients : Débit très bas, les signaux ne sont pas directement compatibles avec des systèmes « évolués » tels qu'ordinateurs, tablette ou smartphone. Il est donc nécessaire d'utiliser une Gateway qui sera connecté d'une part au réseau maillé et d'autre part au réseau local via Wifi ou un câble Ethernet.

- **Wifi** : c'est une technologie de réseau informatique sans fil mise en place pour fonctionner en réseau interne et, depuis, devenue un moyen d'accès à haut débit à Internet. En pratique, pour un usage informatique du réseau Wi-Fi, il est nécessaire de disposer au minimum de deux équipements Wi-Fi, par exemple un ordinateur, et un routeur ADSL.

Nous résumons dans ce tableau une comparaison entre les différentes technologies :

	Wifi	Bluetooth	Zigbee
Débit	>100 Mb/s	<3 Mb/s	<250Kb/s
Portée	~300m	~100m	~1km (beaucoup plus grâce au maillage réseau)
Autonomie	Heures	Jours	Mois/années
Nbr. de nœuds	32	7	65 000 +
Besoin mémoire	1 Mb +	250 Kb +	4 - 32 Kb
IEEE	802.11. a/b/g/n draft	802.15.1	802.15.4

Tableau 1 : Comparaison entre les différents protocoles

Toutes les caractéristiques du protocole Bluetooth sont bien adaptées aux systèmes embarqués. En effet, le protocole Bluetooth se distingue des autres protocoles par sa simple configuration, il est très facile de commander les appareils à l'aide d'une application Android en utilisant Bluetooth que le Wifi par ce qu'on n'aurait pas besoin d'accéder à l'internet, ainsi son faible besoin mémoire par rapport au Wifi. Ce protocole convient parfaitement aux applications nécessitant une vitesse moyenne de transfert de l'ordre de 3 Mb/s. En fin son prix qui est plus ou moins faible vis-à-vis les autres protocoles et leurs rendements.

Pour ses performances et ses grands avantages, nous avons choisi de travailler avec le protocole Bluetooth pour rendre notre maison communicante, on va utiliser également le protocole Infrarouge pour contrôler à distance quelques appareils (Lampes, fenêtres...) à l'aide d'une télécommande infrarouge. Pour la communication à distance nous avons pensé à travailler avec la commande « Web Publishing Tool » que propose le logiciel LabVIEW.

VI. Conclusion

Dans ce chapitre nous avons présenté la domotique, ses fonctionnalités et son impact sur la vie de l'individu. Dans le chapitre suivant nous donnons une présentation générale de notre projet ; les objectifs et les fonctions de domotique proposées.

Chapitre 2

Description du projet

I. Introduction

En plus de simplifier la vie, une maison intelligente devient tout à la fois confortable, communicante, évolutive, autonome, sûre et économe. Notre mission consiste à réaliser une maison intelligente en implémentant des fonctions de domotique à savoir la gestion d'éclairage, l'ouverture sécurisée de la porte principale, l'ouverture et fermeture des volets des fenêtres, l'acquisition de la température à l'intérieur de l'habitat et le contrôle de la ventilation.

II. Présentation du cahier des charges

L'objectif de notre travail est d'établir les fonctions de la domotique suivantes :

- Gestion d'éclairage assurée par l'intermédiaire du capteur PIR et de la photorésistance.
- Gestion d'ouverture/fermeture des volets des fenêtres assurée à l'aide des servomoteurs.
- Gestion d'ouverture sécurisée de la porte principale.
- Acquisition de la température par un capteur de température pour contrôler la ventilation.

La commande des organes du système domotique sera accomplie via l'application Android « Smart Home FST FES » en utilisant le protocole Bluetooth, la télécommande infrarouge et à distance via l'interface graphique avec le logiciel LabVIEW en utilisant le protocole Web Publishing Tool.

Le schéma synoptique suivant illustre le fonctionnement global du système étudié :

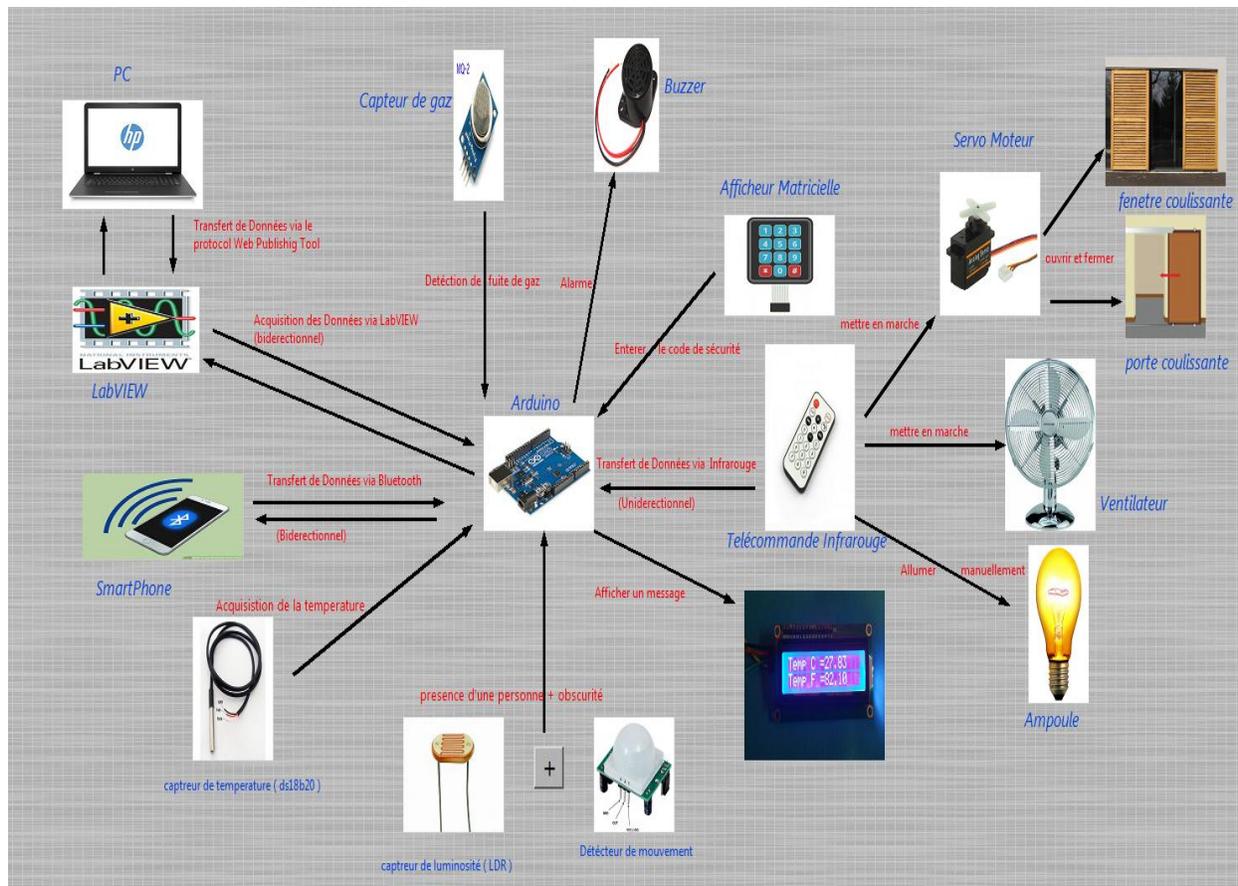


Figure 1 : Schéma globale du système domotique à concevoir

Développement du système domotique

La démarche à suivre pour la réalisation de notre projet est comme suit :

- Assembler les différents composants à savoir les capteurs, actionneurs, afficheur LCD et clavier matriciel.
- Acquisition des données (température, détection de fuite de gaz, détecteur de mouvement, capteur de luminosité, etc.) via le logiciel LabVIEW.
- Créer une application pour l'acquisition des données et la commande sur PC à partir du logiciel LabVIEW.
- Créer l'interface Web en HTML à l'aide du protocole « Web Publishing Tool » proposé par le Logiciel LabVIEW sur laquelle va figurer les liens permettant de commander le système domotique. Ensuite, il faut implanter cette interface dans le programme principal.
- Créer une application Android pour gérer via Bluetooth la maison intelligente.
- Réaliser un prototype d'une maison intelligente en bois.

III. Présentation des fonctions de domotique proposées

1. Fonction de gestion d'éclairage :

Cette fonction permet la gestion de l'éclairage de trois pièces dont le but d'économiser l'énergie électrique. Deux Capteurs réalisent cette fonction : le détecteur de mouvement PIR pour Capteur infrarouge Passif et la photorésistance LDR ' light dependent resistor' qui est un capteur de luminosité. Nous économisons de l'énergie en faisant allumer les lampes quand il y a quelqu'un et que le niveau de luminosité est très faible dans la pièce.



2. Fonction de gestion d'ouverture des volets des fenêtres :



L'utilisateur peut contrôler l'ouverture des fenêtres tout en appuyant sur un simple bouton sur l'application de commande avec son Smart phone ou sur tablette ou sur son PC ou encore sur la télécommande infrarouge.

3. Fonction d'ouverture de la porte principale :

Cette fonction assure l'ouverture de la porte principale de l'habitat d'une manière plus sécurisée en adoptant un système d'accès par clavier.



4. Fonction d'acquisition de la température et ventilation :

L'acquisition de la température se fait via un capteur de température ds18b20 pour contrôler le climat à l'intérieur de l'habitat.



5. Fonction de détection de fuite de gaz :

Cette fonction permet de détecter les fuites de gaz via un capteur de gaz MQ-2. Il est apte à détecter le GPL, le butane, le propane, le méthane, l'alcool, l'hydrogène, la fumée. On peut ajouter une alarme pour informer l'utilisateur en cas de danger.



Chapitre 3

Étude de la partie matérielle et logicielle du projet

I. Introduction

Ce chapitre est consacré à la description de la partie matérielle du projet, en identifiant le choix du cœur du système domotique la carte Arduino Méga 2560 vu ses performances techniques intéressantes. D'autre part, on va développer les organes constituant le système domotique. Par la suite on va voir les outils informatiques utilisés lors de la réalisation du projet.

II. Carte Arduino

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.



Figure 2 : Exemples des cartes Arduino

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

- **Côté d'entrées**, des capteurs qui collectent des informations sur leur environnement comme la variation de température via une sonde thermique, le mouvement via un détecteur de présence ou un accéléromètre, le contact via un bouton-poussoir, etc.

- **Côté des sorties**, des actionneurs qui agissent sur le monde physique telle une petite lampe qui produit de la lumière, un moteur qui actionne un bras articulé, etc.

La plateforme Arduino se présente sur plusieurs séries à savoir : Arduino UNO, Arduino Nano, Arduino LilyPad, Arduino DUE et Arduino Méga 2560 qui sera le cœur de notre système domotique.

1. Arduino Méga 2560

L'Arduino Méga 2560 est une carte électronique basée sur le microcontrôleur ATmega2560. Elle dispose de 54 broches numériques d'entrée / sortie (dont 15 disposent d'une sortie PWM), 16 entrées analogiques, un résonateur céramique (Quartz) à 16 MHz, une connexion USB, une prise d'alimentation, un connecteur ICSP, et un bouton de réinitialisation. Il contient tout le nécessaire pour soutenir le microcontrôleur, tout simplement le connecter à un ordinateur avec un câble USB ou allumez-le avec un adaptateur ou batterie pour commencer.



Figure 3 : Arduino Méga 2560

a. Caractéristiques :

- Tension de fonctionnement : 5V
- Tension d'entrée (recommandé) : 7-12V
- Tension d'entrée (limite) : 6-20V
- Digital I/O Pins : 54 (dont 15 fournissent sortie PWM)
- Broches d'entrée analogiques : 16
- DC Courant par I/O Pin : 40 mA
- Courant DC pour 3,3 Pin : 50 mA
- Mémoire Flash : 256 Ko (ATmega2560) dont 8 Kb utilisé par Boot Loader
- SRAM: 8 Kb(ATmega2560)
- EEPROM: 4 Kb (ATmega2560)
- Fréquence d'horloge : 16 MHz

b. Mémoire

La carte Arduino Méga 2560 à **256 Ko** de mémoire **FLASH** pour stocker le programme. Elle a également **8 ko** de mémoire **SRAM** (volatile) et **4 Ko d'EEPROM** non volatile- mémoire qui peut être lue à l'aide de la librairie EEPROM.h.

c. Entrées et sorties numériques

Chacune des 54 broches numériques de la carte Arduino Méga 2560 peut être utilisée soit comme une entrée numérique, soit comme une sortie numérique, en utilisant les instructions *pinMode ()*, *digitalWrite ()* et *digitalRead ()* du langage Arduino. Ces broches fonctionnent en 5V. Chaque broche peut fournir ou recevoir un maximum de 40mA d'intensité et dispose d'une résistance interne de "rappel au plus" (pull-up) (déconnectée par défaut) de 20-50 KΩ.

d. Broches analogiques

La carte Arduino Méga 2560 dispose de 16 entrées analogiques, chacune pouvant fournir une mesure d'une résolution de 10 bits à l'aide de la très utile fonction *analogRead ()* du langage Arduino. Par défaut, ces broches mesurent entre le 0V (valeur 0) et le 5V (valeur 1023), mais il est possible de modifier la référence supérieure de la plage de mesure en utilisant la broche AREF et l'instruction *analogReference ()* du langage Arduino. Les broches analogiques peuvent être utilisées en tant que broches numériques.

e. Autres broches

Il y a deux autres broches disponibles sur la carte :

- **AREF** : Tension de référence pour les entrées analogiques (si différent du 5V). Utilisée avec l'instruction *analogReference ()*.
- **RESET** : Mettre cette broche au niveau BAS entraîne la réinitialisation du microcontrôleur. Typiquement, cette broche est utilisée pour ajouter un bouton de réinitialisation sur le circuit qui bloque celui présent sur la carte.

f. Communications

La carte Arduino Méga 2560 dispose de toute une série de facilités pour communiquer avec un ordinateur, une autre carte Arduino, ou avec d'autres microcontrôleurs. Elle dispose de 4 UARTs (Universal Asynchronous Receiver Transmitter ou émetteur-récepteur asynchrone universel en français) pour communication série de niveau TTL (5V) et qui est disponible sur les broches 0 (RX) et 1 (TX).

Un circuit intégré ATmega8U2 sur la carte assure la connexion entre cette communication série de l'un des ports série de l'ATmega 2560 vers le port USB de l'ordinateur qui apparaît comme un port COM virtuel pour les logiciels de l'ordinateur.

Le logiciel Arduino inclut une fenêtre terminal série (ou moniteur série) sur l'ordinateur et qui permet d'envoyer des textes simples depuis et vers la carte Arduino.

Les deux LEDs RX et TX sur la carte clignote lorsque les données sont transmises via le circuit intégré ATmega8U2 utilisé en convertisseur USB-vers-série et la connexion USB vers l'ordinateur (mais pas pour les communications série sur les broches 0 et 1).

2. Pourquoi Arduino ?

Le système Arduino simplifie la façon de travailler avec les microcontrôleurs, tout en offrant plusieurs avantages pour les enseignants, les étudiants et les amateurs.

- Pas cher : les cartes Arduino sont peu coûteuses comparativement aux autres plateformes.
- Multiplateforme : Le logiciel Arduino, écrit en Java, tourne sous les systèmes d'exploitation Windows, Macintosh et Linux.
- Un environnement de programmation clair et simple : L'environnement de programmation Arduino est facile, tout en étant assez flexible.
- Logiciel Open Source et extensible : Le logiciel Arduino et le langage Arduino sont publiés sous licence open source, disponible pour être complété par des programmeurs expérimentés.
- Le langage peut être aussi étendu à l'aide de bibliothèques C++.

Ainsi, Pour notre projet nous avons opté pour l'Arduino et précisément l'Arduino Méga 2560.

Vous trouvez dans l'annexe 1 le brochage complet de la carte Arduino Méga 2560.

III. Choix des éléments du système domotique

1. Module Afficheur LCD



Figure 4 : Afficheurs LCD (16x2)

Les afficheurs LCD alphanumériques présentent une solution facile d'emploi et bon marché pour doter notre projet d'une interface indépendante de PC.

Vu de l'extérieur, les écrans LCD alphanumériques sont essentiellement caractérisés par leur taille. Deux modèles se rencontrent très fréquemment et sont les meilleurs marchés, celui ayant 2 lignes et 16 colonnes d'affichage et celui ayant 4 lignes et 20 colonnes d'affichage.

a. Connecteur de l'afficheur LCD

Ces deux écrans ont exactement la même connectique, c'est à dire un connecteur 16 broches. Ce connecteur véhicule plusieurs signaux dont une partie forme un bus de communication parallèle 4 ou 8 bits selon la configuration choisie ainsi que les signaux permettant de contrôler la communication entre l'Arduino et l'écran.

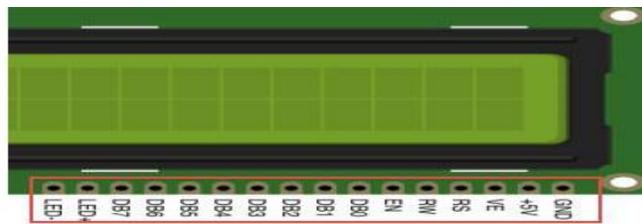


Figure 5 : Connecteur de l'afficheur LC

b. Communication avec le LCD

L'afficheur LCD peut fonctionner en mode 4 bits ou en mode 8 bits. En mode 8 bits, les octets sont transférés sur les lignes DB0 à DB7. En mode 4 bits les octets sont transférés en deux fois sur les lignes DB4 à DB7.

Piloter directement un LCD est un processus relativement compliqué. Toutefois, il y'a des bibliothèques qui permet de les utiliser aisément sans avoir à plonger dans la datasheet.

c. Connexion avec la carte Arduino Méga 2560

Généralement, on préfère une communication sur 4 bits car une communication sur 8 bits consomme 10 ou 11 broches, ce qui laisse peu de broches libres. Avec une communication 4 bits, 6 à 7 broches sont nécessaires. Le choix des broches est libre.

En mode 4 bits, les broches à connecter à l'Arduino sont donc RS, EN, DB4, DB5, DB6 et DB7 ainsi que, de façon optionnelle, RW.

d. Réglage du contraste

La broche VE permet de régler le contraste. Il est nécessaire d'y connecter un potentiomètre de réglage, un 10k Ω par exemple, dont les broches externes sont connectées à l'alimentation (+5V)

et à la masse (GND) et la broche centrale à VE. Il suffit ensuite de tourner ce potentiomètre dans tous les sens avec patience jusqu'à ce que le contraste soit correctement réglé.

e. Pilotage via le bus I2C

Il existe également de petits modules permettant d'interfacer un écran LCD avec un bus I2C. Cette solution peut être intéressante si on manque désespérément de broches sur la carte Arduino comme dans notre projet puisqu'au lieu de monopoliser 6 à 7 broches, l'écran n'en utilisera plus que 2. Toutefois, le module que l'on rencontre le plus souvent est construit autour du PCF8574P de NXP, un circuit permettant d'augmenter le nombre d'entrées sorties numériques via l'I2C.

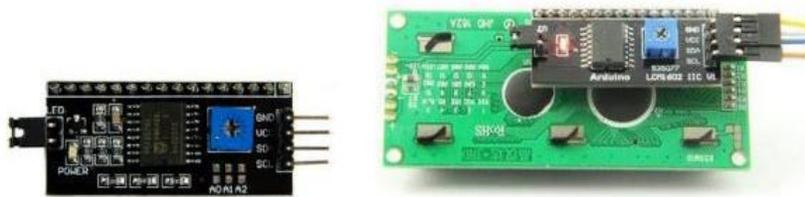


Figure 6: Montage du Module I2C avec l'afficheur LCD

2. Télécommande infrarouge

Ce système sert à donner l'ordre à un automatisme de porte de garage ou de voiture, utilisé pour les télécommandes de TV ou encore pour véhiculer un son dans les casques infrarouges. Elle est composée d'un émetteur et d'un récepteur.

- L'émetteur est une "diode électroluminescente infrarouge", placé à l'avant de la télécommande. Elle émet des rayons infrarouges qui voyagent dans l'air, dans la direction de la télécommande.
- Le récepteur est une photodiode, qui est capable de transformer le signal lumineux qu'elle reçoit en signal électrique.



Figure 7 : télécommande infrarouge avec récepteur

Ses caractéristiques sont :

- Distance de Transmission : 8 m
- Efficace angle : 60 degrés
- Collage Matériel : 0.125 mmPET, la durée de vie effective de 20,000 fois.
- Stable qualité, coût-efficace

- Courant de repos : 3-5uA
- Courant dynamique : 3-5mA.

3. Clavier matriciel

Afin d'assurer l'accès sécurisé à l'habitat, nous avons introduit dans notre système domotique, un sous-système permettant à l'utilisateur d'accéder chez lui tout en introduisant un code sécurisé. Pour cela on a besoin d'un clavier matriciel pour entrer le mot de passe d'ouverture de la porte.

a. Principe du clavier matriciel

Un clavier matriciel (dans notre cas 16 touches) dispose uniquement de 8 broches pour la gestion de ses touches. L'organisation est de 4 colonnes et 4 lignes. Les lignes ont un état de repos imposé (ici l'état bas).

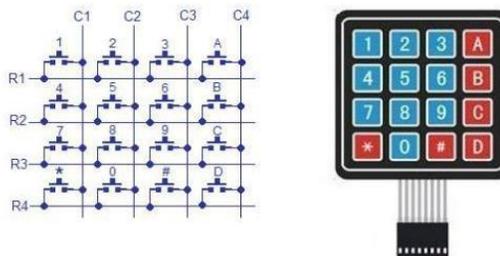


Figure 8 : Clavier matriciel (4*4) et son schéma de principe

b. Détection des touches

Il faut balayer les colonnes une à une par un état haut. Ainsi dès lors qu'une touche est pressée, l'état de la colonne est transmis sur la ligne. Nous pouvons ainsi détecter un état haut. Pour déterminer la touche pressée, il faut se synchroniser avec le signal envoyé sur les colonnes.

c. Branchement avec Arduino Méga 2560

Pour le branchement du clavier matriciel avec la carte Arduino Méga 2560, on a choisi les broches numériques de (D22 à D30) avec (D22 à D26 vers les colonnes et de D27 à D30 vers les lignes) :

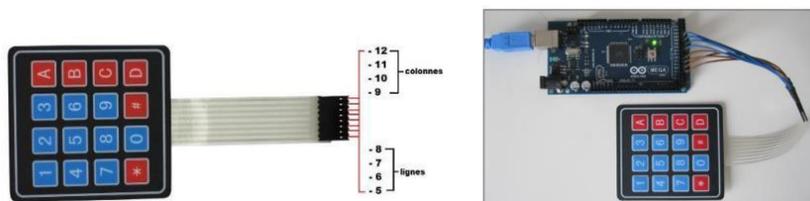


Figure 9 : Branchement du clavier avec l'Arduino

4. Ventilateur

La domotique s'applique également au chauffage et à la ventilation, des éléments indispensables au sein d'un logement. La solution est la régulation du chauffage, c'est-à-dire programmer un degré de température adapté à chaque pièce et notamment en fonction de moments de la journée, nuit et jour. Ces températures par zone sont contrôlées depuis l'application Android ou bien le serveur web de LabVIEW conçue pour donner de la fraîcheur à l'intérieur de l'habitat.

Dans notre projet, on a modélisé le ventilateur par un simple ventilateur de l'ordinateur familial :



Figure 10 : Ventilateur 5V d'un PC

5. Capteur de température DS18B20

Afin de garder de la fraîcheur au sein de l'habitat, on a intégré dans notre système domotique un sous-système d'acquisition de température avec le capteur de température ds18b20. La température acquise va être renvoyée vers l'utilisateur sur l'application sous Smart Phone ou son ordinateur d'une manière automatique.



Figure 11 : Capteur DS18B20 en boîtier TO-92

Le capteur DS18B20 du fabricant Maxim (anciennement Dallas Semiconducteur) est un capteur de température numérique intégrant tout le nécessaire requis pour faire la mesure : capteur analogique, convertisseur analogique/numérique, électronique de communication et alimentation.

Il communique via un bus 1-Wire et possède une résolution numérique de 12 bits avec une plage de mesure de -55°C à $+125^{\circ}\text{C}$. La précision analogique du capteur est de $0,5^{\circ}\text{C}$ entre -10°C et $+85^{\circ}\text{C}$, ce qui rend ce capteur très intéressant pour une utilisation "normale".



Figure 12 : Capteur DS18B20 en format "sonde" étanche

Le capteur DS18B20 existe dans le commerce en deux versions : en boîtier TO-92 (format transistor, en photo un peu plus haut) pour des utilisations standards en intérieur, ou en format "sonde étanche" pour des applications en milieu humide/extérieur.

6. Capteur de la lumière : la photorésistance (LDR)

Pour ne pas gaspiller de l'énergie pendant la journée, lorsque la lumière issue des fenêtres est suffisante pour éclairer les pièces de notre maison, les lampes seront éteintes jusqu'au soir où y aurait plus de lumières issues des fenêtres. La solution c'est d'utiliser un capteur de lumière.



Figure 13 : Capteur de la lumière

Le capteur LDR est une photorésistance, une cellule photo-électrique sensible à l'intensité lumineuse qu'elle traduit en produisant un petit courant électrique sensible à la quantité de lumière reçue. Une variation de lumière provoque une variation de signal. La sensibilité du capteur LDR est proportionnelle à la lumière reçue, plus le flux lumineux sera intense, plus le nombre d'électrons disponibles pour assurer la conduction sera grand. L'efficacité est fonction de la longueur d'onde de la lumière et la courbe de sensibilité est voisine de celle de l'œil humain.

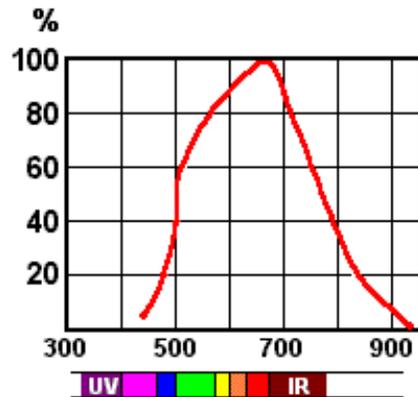


Figure 14 : Courbe de sensibilité en fonction de la longueur d'onde de la lumière

Ce module fournit une tension de 5V lorsqu'il ne reçoit aucune lumière (le circuit est ouvert) et 0V lorsqu'il reçoit une lumière importante.

7. Capteur de mouvement (PIR)

Les capteurs PIR (« passive infrared », « pyroélectriques » ou « à mouvement infrarouge »), sont utilisés comme des capteurs de mouvement. Ils permettent de déterminer si un être humain est entré ou sorti du champ de détection du capteur.



Figure 15 : Capteur de mouvement (PIR)

Spécifications techniques du capteur PIR :

- Alimentation : 5-16V
- Signal de sortie numérique : 3,3V
- Portée : 7m
- Cône de détection : 120°
- Sensibilité et délai de réponse (2-4 s) ajustables

- Câble de 30 cm inclus
- 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

8. Capteur de gaz/fumée (MQ-2)

Le capteur de gaz inflammable et de fumée MQ2 détecte la concentration des gaz combustibles dans l'air et renvoie sa lecture comme tension analogique. La sonde peut mesurer des concentrations du gaz inflammable de 300 à 10.000 ppm. Le capteur peut fonctionner à des températures de -20 à 50°C et consomme moins de 150 mA à 5 V.

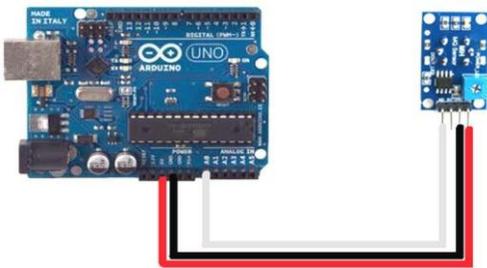


Figure 16 : Capteur de gaz/fumée MQ-2

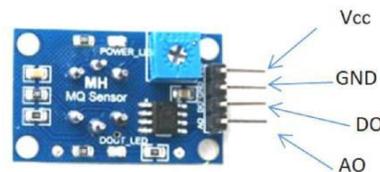


Figure 17 : Branchement du MQ-2 avec Arduino

9. Module Buzzer

Ce module est utilisé pour déclencher une alarme s'il y a un danger dans notre maison comme une fuite de gaz ou bien s'il y a une intrusion.

- Caractéristiques :**
- Voltage : 3.5 - 5 v
 - Courant : < 25mA
 - Height : 7mm

- Brochage :**
- VCC du buzzer → VCC de l'Arduino
 - GND du buzzer → GND de l'Arduino
 - I/O → un pin PWM de l'Arduino

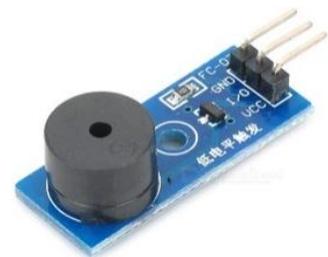


Figure 18 : le buzzer

10. Moteur à courant continu

Pour motoriser nos volets des fenêtres ainsi de la porte principale, on a pensé à utiliser des moteurs à courant continu.

Un moteur est un composant de conversion d'énergie électrique en énergie mécanique.

Les moteurs à courant continu transforment l'énergie électrique en énergie mécanique de rotation. Mais ils peuvent également servir de générateur d'électricité en convertissant une énergie mécanique de rotation en énergie électrique. C'est le cas par exemple de la dynamo d'un vélo !

Le moteur à courant continu est composé de deux parties principales : le rotor (partie qui tourne) et le stator (partie qui ne tourne pas, statique). En électrotechnique, le stator s'appelle aussi inducteur (qui fait l'action d'induire) et le rotor s'appelle l'induit (qui subit l'action d'induction).

a. La mécanique liée au moteur :

Le couple : c'est la capacité du moteur à faire tourner quelque chose sur son axe. Plus le couple est élevé et plus le moteur sera capable de mettre en mouvement quelque chose de lourd.

L'unité du couple est le Newton-Mètre La formule qui relie le couple et la force exercée sur l'axe de rotation d'un moteur et le rayon de l'action : $C = F \times r$

La vitesse de rotation : elle est mesurée par rapport à l'axe de rotation du moteur. On mesure une vitesse de rotation en mesurant l'angle en radians parcourus par cet axe pendant une seconde. La vitesse de rotation angulaire n'est pas donnée avec les caractéristiques du moteur. En revanche, on trouve une vitesse en tour/minutes (tr/mn). Voici la relation qui relie les deux vitesses : $1 \text{ rad/s} = 9,55 \text{ tr/mn}$

Les réducteurs : Un moteur électrique est bien souvent très rapide en rotation. Or dans notre cas, Le mouvement des volets et de la porte principale n'est pas rapide, il fait donc réduire la vitesse de rotation du moteur. On peut très bien mettre un "frein" pour empêcher le moteur de tourner vite. Néanmoins, le moteur ne va pas pouvoir supporter des charges lourdes. Nous avons besoin d'un couple élevé. On va utiliser alors un réducteur composé d'engrenages qui permet de réduire la vitesse de rotation de l'axe du moteur tout en augmentant le couple de sortie.

Sur la figure suivante, on peut observer un ensemble moteur + réducteur.



Figure 19 : moteur à CC

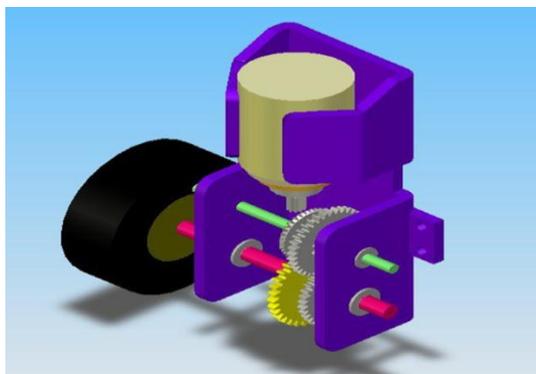


Figure 20 : moteur + réducteur

b. Alimentation du moteur :

Pour l'alimentation du moteur à courant continu, nous allons utiliser le « L293D »

Le composant L293D est un pont de puissance composé de plusieurs transistors et relais qui permet d'activer la rotation d'un moteur. Le L293D est un double pont-H, ce qui signifie qu'il est possible de l'utiliser pour commander quatre moteurs distincts (dans un seul sens) grâce à ses 4 canaux.

Il est également possible de constituer deux pont-H afin de piloter deux moteurs distincts, dans les deux sens et indépendamment l'un de l'autre.

- **Caractéristiques techniques du L293D**

- Nbre de pont-H : 2
- Courant Max Régime continu : 600mA (x2)
- Courant de pointeMax < 2ms : 1200mA
- VS Max Alim moteur : 36v
- VSS Max Alim logique : 7v
- Nbre de Broches : 16 DIP
- Perte de tension : 1.3v

Le schéma suivant définit les différentes broches du composant L293D :

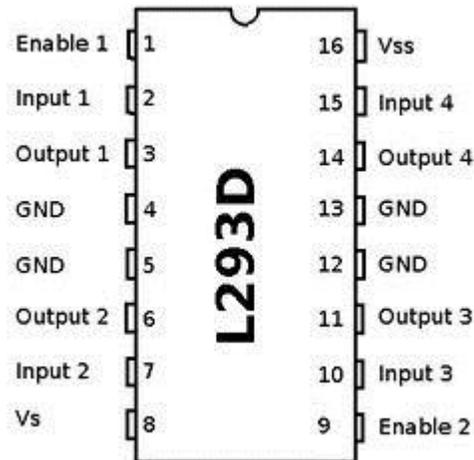


Figure 21 : les pins de L293D

Les pins Enable1 et Enable2 permettent moduler la vitesse du moteur en utilisant des broches **PWM** de l'Arduino.

- **Fonctionnement du L293D :**

Le tableau suivant montre le fonctionnement de moteur DC en utilisant le composant L293D :

Enable 1	Input 1	Input 2	Fonction
High	Low	High	Tourne dans le sens horlogique
High	High	Low	tourne dans le sens anti-horlogique
High	Low	Low	Stop
High	High	High	Stop
Low	Non applicable	Non applicable	Stop

Tableau 2 : commande du L293D

- **Schéma pour brancher deux moteurs à l'Arduino :**

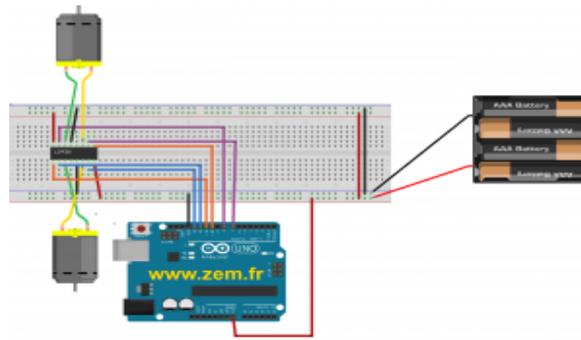


Figure 22 : brochage de deux moteurs avec l'Arduino

11. Module Bluetooth HC-05

Ce module communique via une liaison série avec une carte Arduino. Cette liaison s'établit sur deux broches RX et TX.

C'est une communication bidirectionnelle, deux modules peuvent communiquer ensemble en même temps. Le comportement utilisé est "maître/esclave". Un esclave pourra parler avec un seul maître, mais un maître pourra dialoguer avec plusieurs esclaves. Pour son utilisation, elle se passe en plusieurs étapes :

1. Le maître se met en mode "reconnaisable"
2. L'esclave trouve le maître et demande à s'y connecter
3. Le maître accepte la connexion
4. Les périphériques sont alors associés
5. La communication peut commencer

- **Caractéristiques :**

- Fonctions maître / esclave.
- Tension de fonctionnement de 3.3V
- Tension d'entrée 3.3 à 6V.
- Baud rate par défaut de 9600, modifiable par l'utilisateur.
- Taille 39 x 15 mm
- Intensité 30 mA en fonctionnement,
- 8mA pour établir la communication
- Utilisation pour communication par Bluetooth avec téléphones portables, tablettes, ordinateurs
- Une LED indique le fonctionnement : clignote si non connecté ; éclairage fixe quand il est connecté.
- Utilise un régulateur 150 mA 3.3V

- **Branchement du HC-05 :**

- RXD -> TXD de l'Arduino.
- TXD -> RXD de l'Arduino.
- GND -> GND
- VCC -> 5V

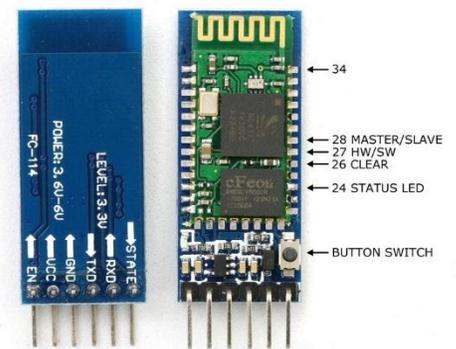


Figure 23 : module Bluetooth HC-05

IV. Étude de la partie logicielle

Cette partie est dédiée à la représentation des plateformes informatiques utilisées dans le développement du système domotique.

1. Plateforme de programmation Arduino

L'interface de l'IDE Arduino est plutôt simple, 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 et d'une barre d'outils rapide. 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, plus classique qui est utilisé pour accéder aux fonctions avancées de l'IDE. Enfin, une console affichant les résultats de la compilation du code source, des opérations sur la carte, etc.



Figure 24 : le logiciel 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** » contient toutes les opérations nécessaires à la configuration de la carte (directions des entrées sorties, débits de communications série, etc.).
- La fonction « **loop** » est exécutée en boucle après l'exécution de la fonction setup. Elle continue 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'ils n'ont pas de système d'exploitation.

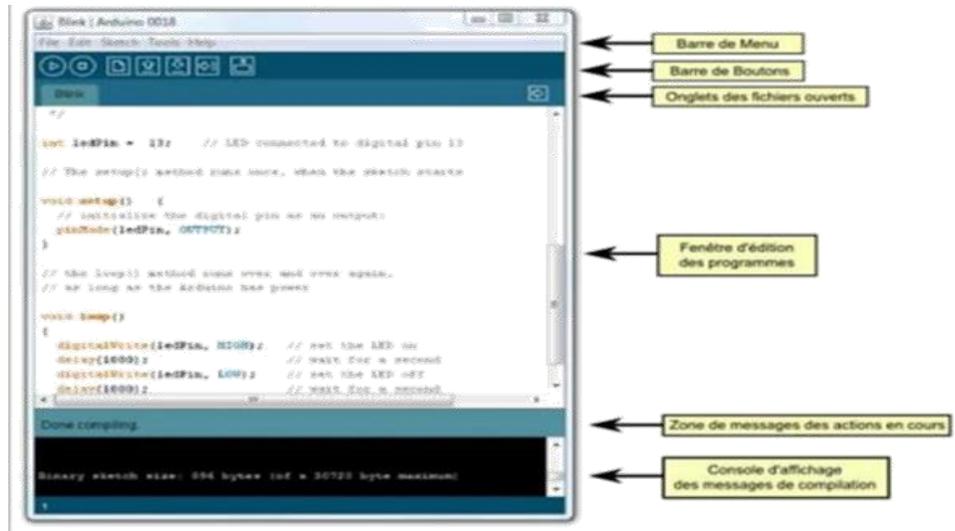


Figure 25 : Interface de la plateforme Arduino

Le logiciel comprend aussi un moniteur série (équivalent à HyperTerminal) qui permet d'afficher des messages textes émis par la carte Arduino et d'envoyer des caractères vers la carte Arduino :

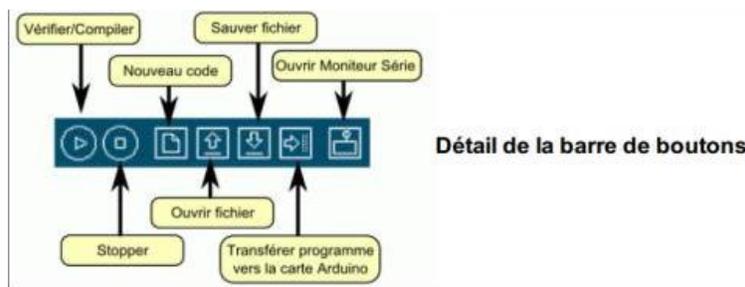


Figure 26 : Barre de boutons Arduino



Figure 27 : HyperTerminal de l'Arduino (Moniteur Série)

Un programme utilisateur Arduino est une suite d'instructions élémentaires sous forme textuelle, ligne par ligne. La carte lit puis effectue les instructions les unes après les autres, dans l'ordre défini par les lignes de code. La structure d'écriture d'un programme sous Arduino est de la forme suivante :

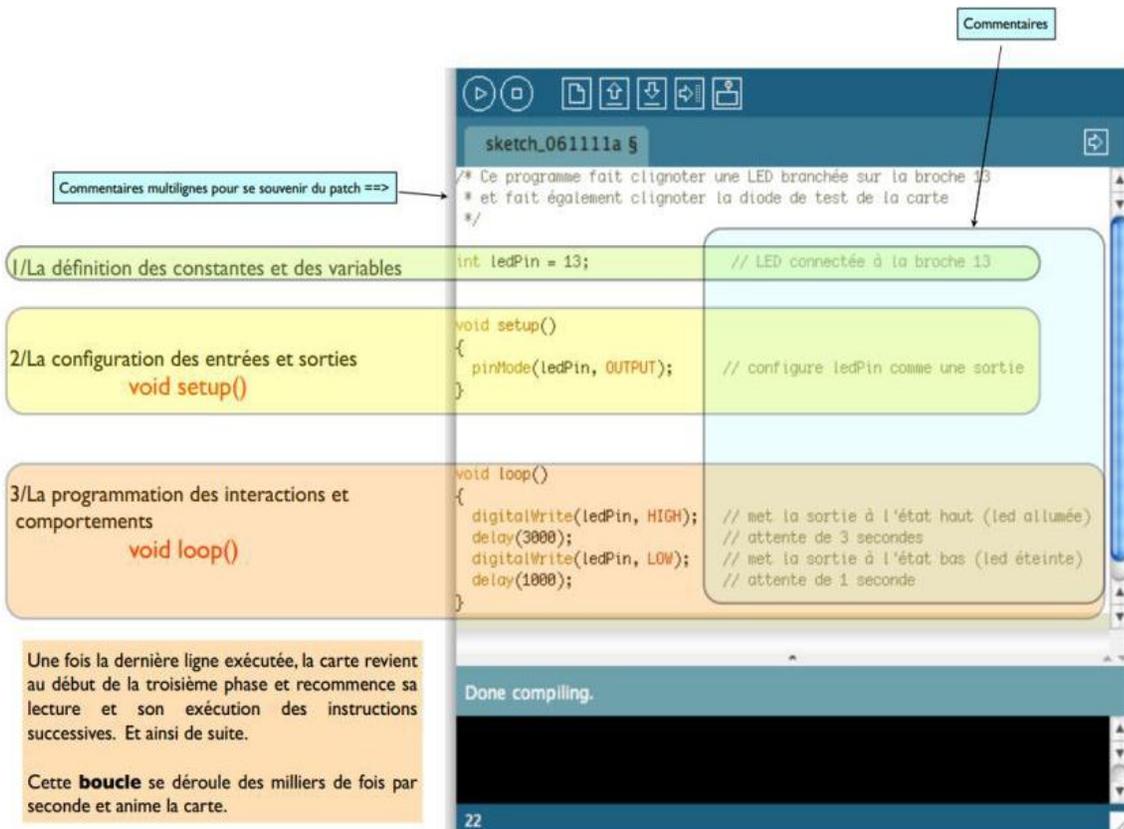


Figure 28 : Structure d'un programme en Arduino

2. Plateforme de Logiciel LabVIEW

LabVIEW (Laboratory Virtual Instrument Engineering WorkBench) est un logiciel de développement de programmes d'application. Il utilise un langage de programmation essentiellement graphique dédié au contrôle, à l'acquisition, l'analyse et la présentation de données. Il est dédié à la programmation conçue pour le pilotage d'instruments électronique. Son principe de programmation est basé sur l'assemblage graphique de modules logiciels appelés « Instruments Visuels (« VI »). Le rôle d'un VI est d'acquérir des données issues par exemple de fichiers, du clavier ou encore de cartes électroniques d'Entrée/Sorties », de les analyser, et de les présenter au travers d'interfaces hommes-machines graphiques (encore appelées « face avant » par analogie avec la face avant permettant de piloter un appareil électronique).

a. Fonctionnement :

Une application LabVIEW est représentée sous forme d'un VI. Il y a deux étapes : la définition du GUI (*front panel*) qui est la conception de l'interface utilisateur où l'on dessine et place tous les éléments visuels (contrôles) :

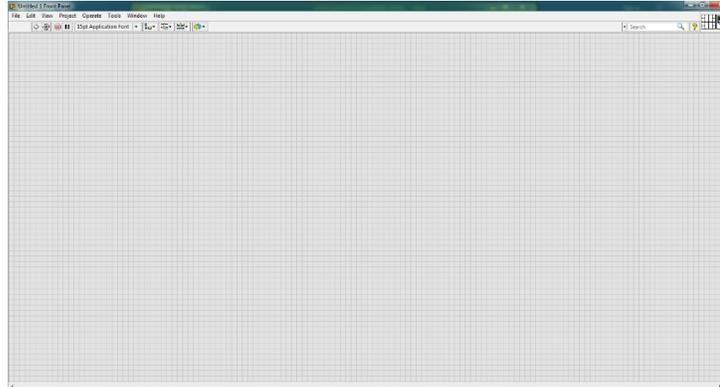


Figure 29 : front panel (page d'interface)

- Contrôles d'entrées de l'utilisateur : bouton, interrupteur, potentiomètre, curseur, zone de saisie de valeur/listes ...
- Contrôles de sortie du programme : voyant, graphe, thermomètre, zone de texte ...

Chaque contrôle utilisé dans le front panel va créer un objet *terminal* dans le Block Diagram.

Le diagramme est présenté dans une fenêtre séparée. Il contient le code source graphique du VI, il permet d'effectuer des traitements sur les entrées et sorties créées dans la face avant

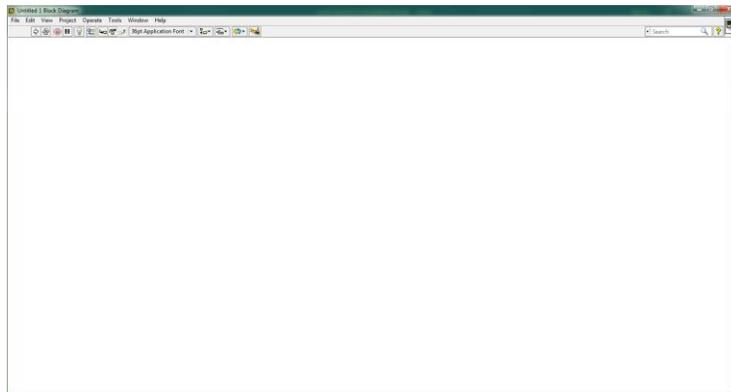


Figure 30 : block Diagram

b. Langage utilisé :

Le diagramme utilise le langage de programmation G qui est un langage graphique, différent des autres langages de programmation (« Pascal », « C », ou encore Fortran), qui eux sont des

langages textuels. La programmation est de type évènementiel avec retour au système après exécution des chemins. Il n'y pas d'ordre d'exécution du code a priori. Plusieurs chemins câblés de même niveau s'exécutent en parallèle au fur et à mesure que leurs données d'entrée soient disponibles. Un mode dynamique permet de visualiser l'exécution : une bulle se déplace sur les branches au fur et à mesure de la progression.

c. Contrôle à distance en utilisant LabVIEW :

Pour rendre notre maison communicante, nous avons besoin de la contrôler à distance et connaître par exemple la température et la régler avant notre rentrée le soir ou bien allumer la lampe d'une chambre avec un petit geste sur notre pc ou tablette.

Pour cela nous allons profiter des avantages que propose ce logiciel, en exploitant le Protocol de communication avec internet appelé : WEB PUBLISHING TOOL

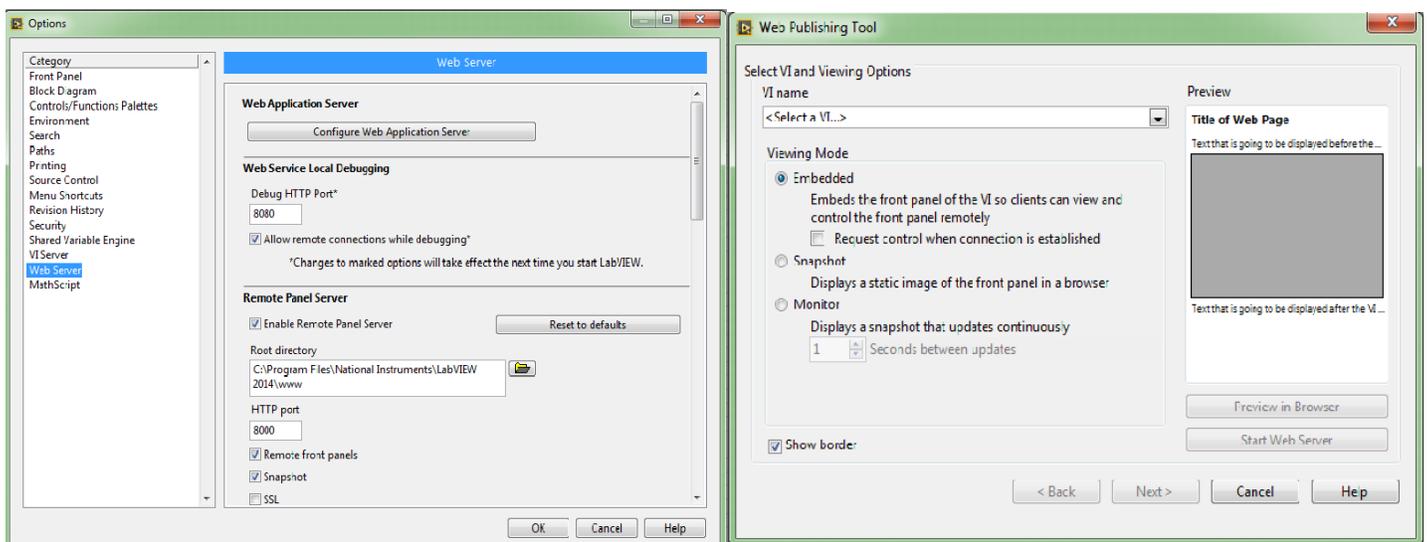


Figure 31 : Configuration du Web Server

V. Conclusion

Dans ce chapitre nous avons vu la partie matérielle et logicielle dédiée à notre projet. Les composantes de notre solution ont été détaillées puis, les applications et modules basée sur la technologie Arduino et le logiciel LabVIEW ont été présentées.

Ces outils matérielles et logiciels sont nécessaires pour concevoir notre maison intelligente dans le chapitre suivant.

Chapitre 4

Réalisation du système domotique

Dans ce chapitre nous allons développer les commandes permettant à l'utilisateur d'optimiser sa consommation d'énergie, de sécuriser sa maison et de contrôler son système domotique à distance d'une manière fiable et automatique.

I. Construction de la maison

La première étape consiste à fabriquer un modèle réduit d'une maison. Nous avons choisi une maison de trois pièces, munie d'une fenêtre et d'une porte. Cette maquette, permettrait de présenter certaines fonctionnalités de la domotique à travers sept scénarios : détection de mouvement, accès sécurisé à l'habitat, ouverture des volets de la fenêtre, éclairage à distance, acquisition de la température, détection de gaz/fumée et la ventilation. Ces scénarios seront automatisés via des cartes Arduino exécutant des programmes informatiques.



Figure 32 : Photos de notre maison

II. Développement de l'application de commande

1. Détection de mouvement

La gestion de la consommation d'énergie est une tendance actuelle, c'est pourquoi nous avons modélisé un premier scénario qui respecte cette tendance. Notre dispositif permet d'allumer automatiquement la lumière en cas de présence dans une pièce si la luminosité est faible. Il permet aussi d'éteindre celle-ci en l'absence de mouvement, le code est en annexe 5.

2. Fonction d'accès sécurisé à l'habitat

La sécurité est un élément primordial dans une maison. L'accès sécurisé à l'habitat est assuré en introduisant le code correct par clavier. L'organigramme se trouve en annexe 2 et le code en annexe 3. Cette fonction sera indépendante des autres fonctions qui seront commandées à distance.



Figure 33 : système de sécurité

3. Fonction d'ouverture des volets de la fenêtre

La commande d'ouverture de la fenêtre est réalisée à distance via l'application HTML, l'application Android avec Bluetooth et la télécommande Infrarouge, en agissant sur le contrôle du moteur courant continue pour faire monter/descendre le volet de la fenêtre, (Annexe 6,7).

4. Fonction d'éclairage

Notre dispositif permet d'allumer à distance la lumière via l'application HTML, commandée à travers la carte Arduino Méga sous LabVIEW. La fonction d'éclairage est assurée aussi à l'aide de l'application Android à travers le modèle Bluetooth HC-05, ou encore par les rayons infrarouges en utilisant une télécommande, (Annexe 5,6,7).

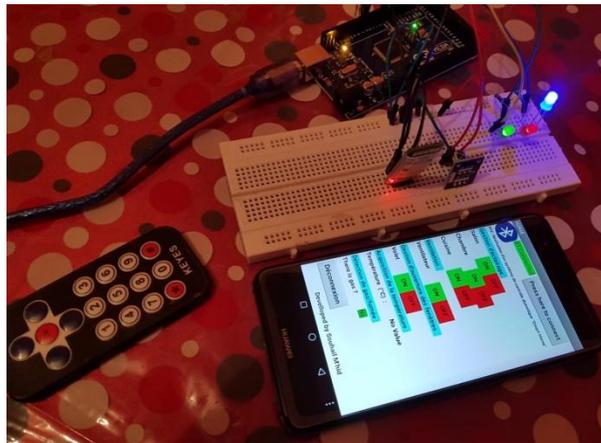


Figure 34 : Commande d'éclairage

5. Fonction de l'acquisition de la température

La fonction de l'acquisition de la température est réalisée via la sonde ds18b20. Les valeurs en degré seront affichées sur l'afficheur LCD, sur la page HTML et aussi sur l'application Android. Ensuite, on peut lancer la commande d'ouverture ou fermeture de la fenêtre, (Annexe 4,6).

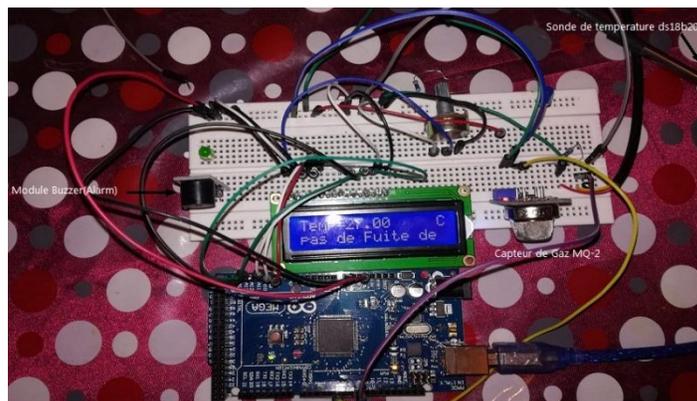


Figure 35 : l'acquisition de la température

6. Fonction de détection de gaz/fumée

Elle permet de détecter s'il y a des fuites de gaz dans la cuisine à l'aide du capteur MQ-2 en affichant sur les applications de commande un message. Par la suite, on peut lancer une alarme pour informer l'utilisateur en cas de danger, (Annexe 4,6).

7. Fonction de ventilation

La ventilation de l'intérieur de l'habitat est assurée par l'intermédiaire des applications ou automatiquement quand il y a détection des fuites de gaz (déclaration d'une fuite de gaz pour lancer la ventilation), et quand la température de la maison dépasse un certain degré (nous avons mis 30°C comme seuil), (Annexe 4,7).

III. Centralisation des commandes : Interface

Nous avons développé une interface centralisant les différents modules permettant à la fois de procéder à des ajouts ou des retraites de module ainsi que d'offrir à l'utilisateur (de la maison) une interface afin d'interagir directement et facilement avec les modules.

Nous avons procédé de deux manières :

- En créant une page HTML avec LabVIEW à travers le protocole WEB PUBLISHING TOOL, (figure 36).

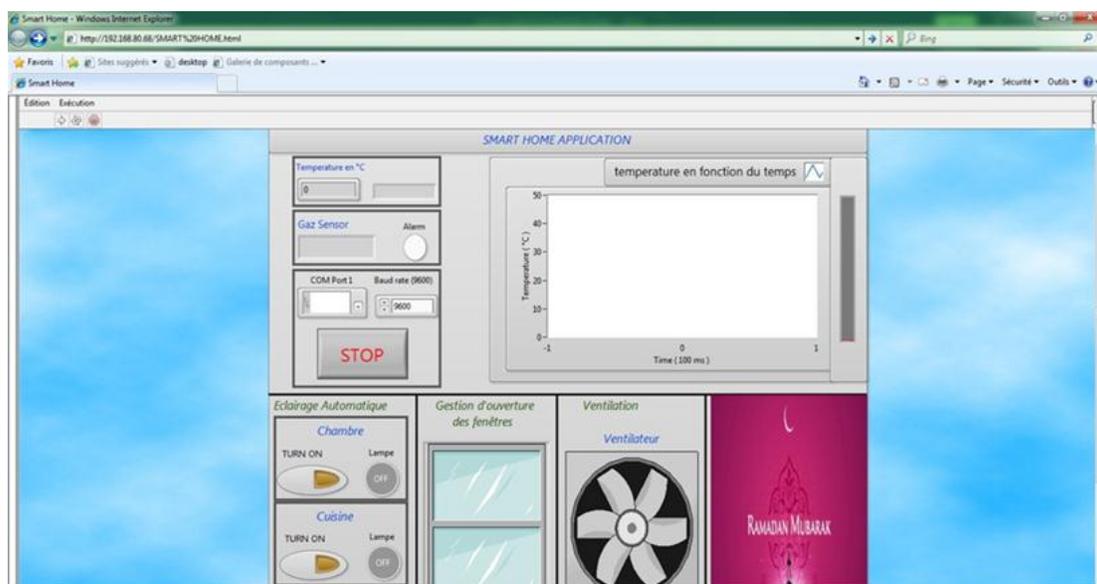


Figure 36 : page HTML de commande à distance via une adresse IP

- En développant une application Android de commande à distance via Bluetooth, (Figure37).

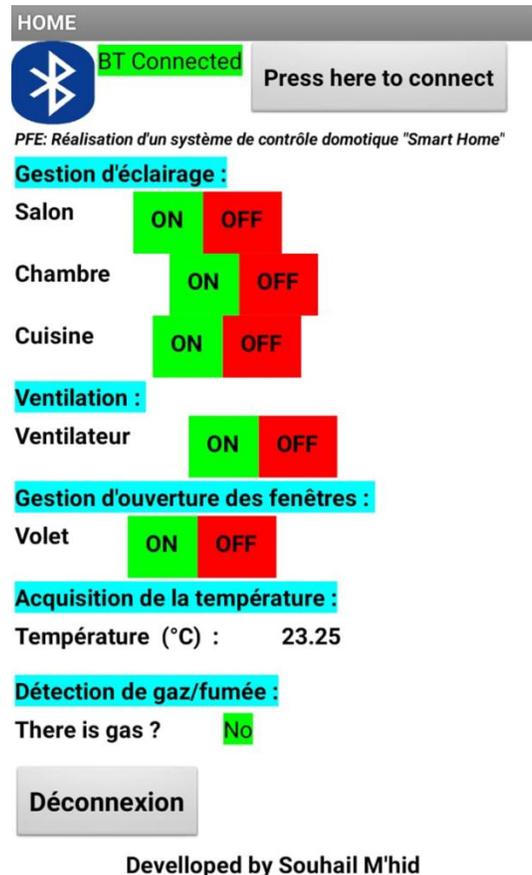


Figure 37 : application Android de commande à distance via Bluetooth

IV. Conclusion

Ce travail nous a permis d'affirmer qu'il est possible de faire de la domotique à moindre coût au prix de quelques efforts. Il conviendra, cependant, d'élargir le nombre de modules, en gardant à l'esprit qu'il y a des modules qui sont coûteux et gourmands en énergie. Les seules limitations restent l'imagination et les connaissances techniques.

Conclusion générale

Dans notre projet, nous avons mis en œuvre un système embarquée qui répond à des fonctions principales de la domotique à savoir la gestion d'éclairage, l'accès sécurisé à l'habitat, l'ouverture à distance des volets, l'acquisition de la température à l'intérieur de l'habitat, la détection d'une fuite de gaz et le contrôle de la ventilation.

Nous avons créé une application HTML sous LabVIEW et une autre avec Android en exploitant le protocole Bluetooth pour répondre aux besoins quotidiens de confort de la vie quotidienne, de sécurité et d'économie d'énergie.

Pour les perspectives, nous allons essayer de développer les points suivants :

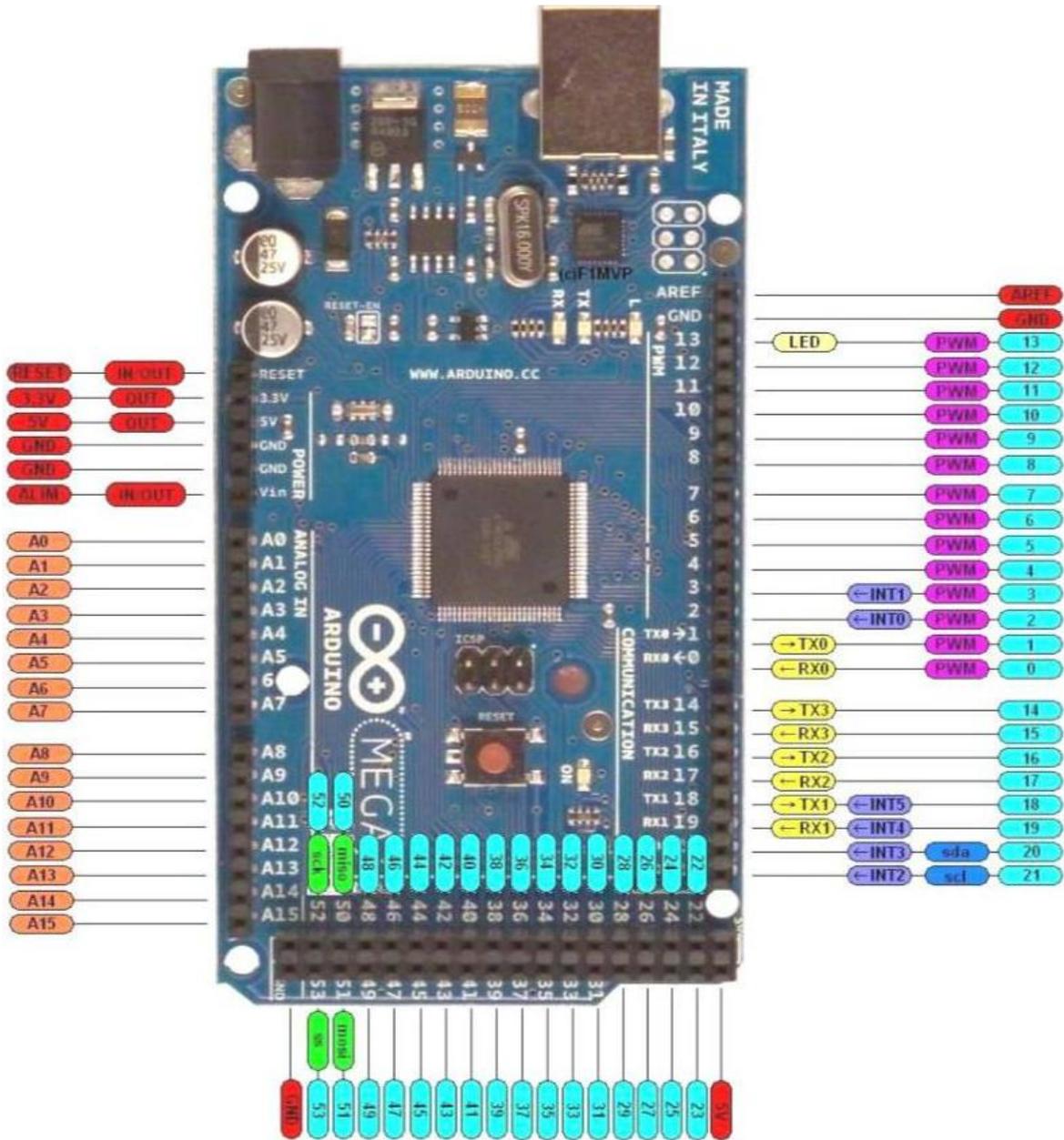
- Ajouter une fonction de détection d'intrusion tout en utilisant une caméra de surveillance avec un traitement d'images pour la détection facial des personnes.
- Réaliser le contrôle domotique en utilisant un module GSM qui permet d'augmenter la possibilité de surveiller l'habitat tout en communiquant via des commandes envoyées par SMS.

En termes de compétences acquis, ce projet m'a permis de :

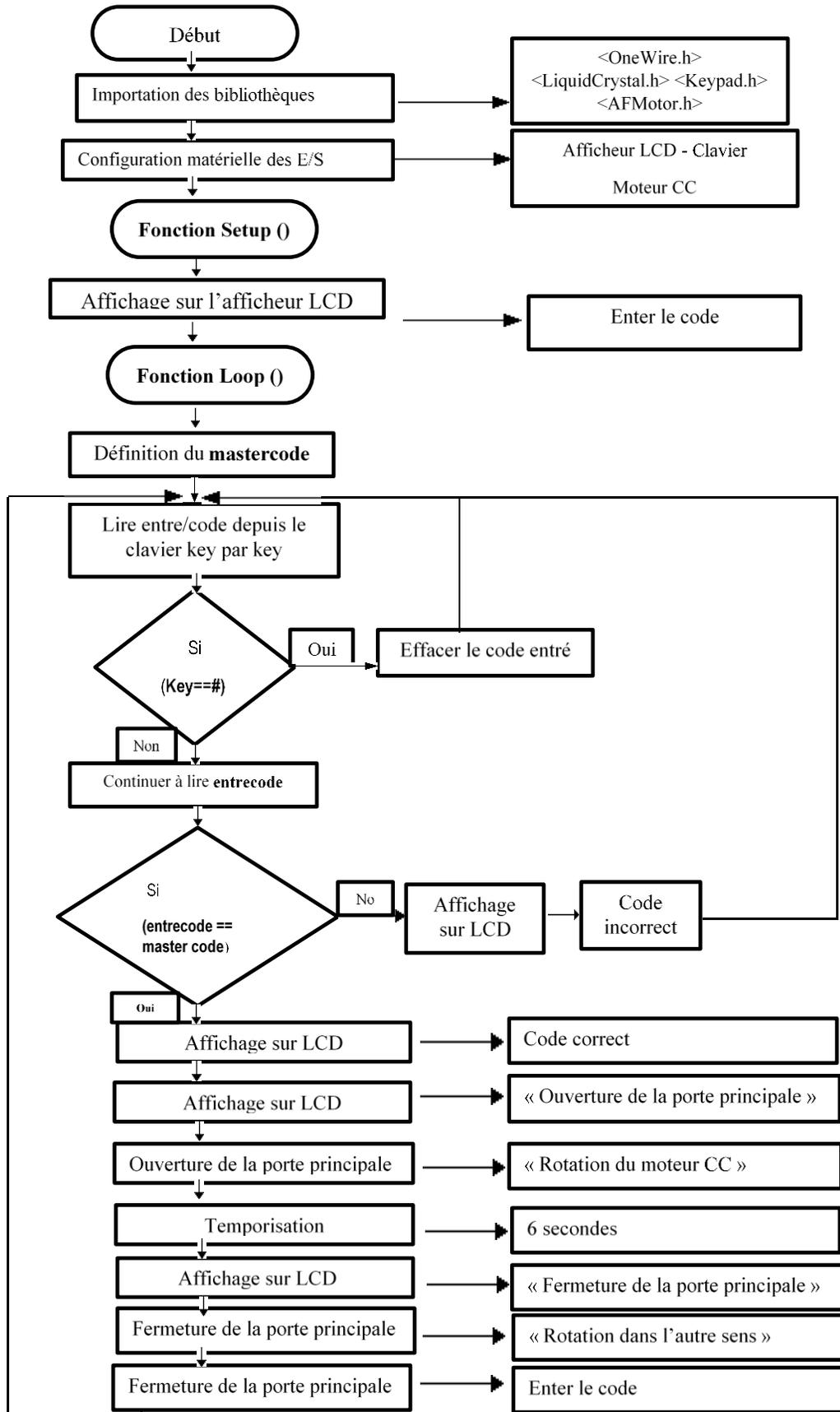
- Savoir programmer un microcontrôleur (Arduino).
- Connaître le logiciel LabVIEW et d'être capable de faire la liaison entre la carte Arduino et ce logiciel pour faire l'acquisition des données et aussi de commander en retour.
- Créer une page HTML en utilisant LabVIEW, qui est très répondeur dans l'acquisition des données, traitement d'image, simulation en 3D etc...., en utilisant le protocole WEB PUBLISHING TOOL
- Créer une application Android
- Faire face à des problèmes, des vrais problèmes pas uniquement en théorie, savoir les gérer et le plus important trouver des solutions fiables pour répondre à ces problématiques.

Annexes

Annexe 1 : Brochage de la carte Arduino Méga 2560



Annexe 2 : Organigramme de la fonction de sécurité



Annexe 3 : Code Arduino de la fonction d'accès sécurisé à l'habitat

```

accés_sécurisé
#include <LiquidCrystal.h> //Declaration de la bibliothèque de l'afficheur LCD
#include <Keypad.h> // Declaration de la bibliothèque du clavier matriciel
#include <AFMotor.h> // Declaration de la bibliothèque du moteur DC
char* password ="1234"; //création d'un mot de passe
int pozisyon = 0; // position du clavier
const byte rows = 4; //nombre de lignes du clavier
const byte cols = 4; //nombre de colonnes du clavier
char keyMap [rows] [cols] = { //définition des caracteres du clavier
    {'1', '2', '3', 'A'},
    {'4', '5', '6', 'B'},
    {'7', '8', '9', 'C'},
    {'*', '0', '#', 'D'}
};

byte rowPins [rows] = {53, 51, 49, 47}; //les pins du clavier
byte colPins [cols] = {45, 43, 41, 39};
Keypad myKeypad = Keypad( makeKeymap(keyMap), rowPins, colPins, rows, cols);
LiquidCrystal lcd (8, 9, 10, 11, 12, 13); // les pins de l'afficheur LCD. (RS, E, D4, D5, D6, D7)

AF_DCMotor motor(1); // déclaration du moteur qu'on va utiliser
int led = 20; // led qui va clignoter pendant l'ouverture de la porte
void setup(){
    lcd.begin(16, 2);
    pinMode(led, OUTPUT); //mettre led comme sortie
    setLocked (true); //état du mot de passe
    motor.setSpeed(155); // définition de la vitesse du mot de passe
}

void loop(){
    char whichKey = myKeypad.getKey(); //définir quel code a été pressé avec getKey
    lcd.setCursor(0, 0);
    lcd.print(" Welcome");
    lcd.setCursor(0, 1);
    lcd.print(" Enter Password");
    if(whichKey == '*' || whichKey == '#' || whichKey == 'A' || //définir un code incorrect
        whichKey == 'B' || whichKey == 'C' || whichKey == 'D')
    {
        pozisyon=0;
        setLocked (true);
        lcd.clear();
        lcd.setCursor(0, 0);
        lcd.print(" Invalid Key!");
        delay(1000);
        lcd.clear();
    }

    if(whichKey == password [pozisyon])
    {
        pozisyon ++;
    }
    if(pozisyon == 4){
        setLocked (false);
        lcd.clear();
        lcd.setCursor(0, 0);
        lcd.print("**** Verified ****");
        delay(2000);
        lcd.clear();
        lcd.setCursor(0, 0);
        lcd.print(" Porte ouverte");
        lcd.setCursor(0, 1);
        lcd.print("Smart Home");
        motor.run(FORWARD);
        delay(3000);
        motor.run(RELEASE);
        delay(4000);
        lcd.clear();
    }
    delay(100);
}

void setLocked(int locked)
{
    if(locked){
        digitalWrite(led, LOW);
    }
    else{
        digitalWrite(led, HIGH);
        delay(200);
        digitalWrite(led, LOW);
        delay(200);
    }
}

if(sensorgas==0)
{
    lcd.setCursor(0,1);
    lcd.print("Pas de fuite de Gaz :)");
    digitalWrite(led1,LOW);
    noTone(buzzer);
    delay(100);
}
}

```

Annexe 4 : Code d'acquisition de la température et de la détection de fuite de gaz

```
#include <LiquidCrystal.h> // Declaration de la bibliothèque de l'afficheur LCD
#include <OneWire.h> // Declaration de la bibliothèque de de liaison entre la sonde de temperature ds18b20 et l'Arduino
#include <DallasTemperature.h> // Declaration de la bibliothèque de la lecture de la temperature à partir de la sonde ds18b20
#define ONE_WIRE_BUS 8 // configurer le pin 8 pour recevoir les valeurs de la temperature
LiquidCrystal lcd(12, 11, 5, 4, 3, 2); // les pins de l'aficheur LCD. (RS, E, D4, D5, D6, D7)
OneWire oneWire(ONE_WIRE_BUS); // configurer le pin 8 pour recevoir les données sous forme oneWire
DallasTemperature sensors(&oneWire); // Passer notre référence oneWire au Dallas Temperature.
const int PinGaz = 7; // définir le pin 7 pour recevoir l'etat du de capteur de gaz
const int redled1 = 10; // allumer une led en cas de fuite de gaz
const int redled2 = 9; // allumer une led en cas de dépassement du la marge autorisée pour la temperature
const int buzzer = 15; // mettre le buzzer sur le pin 15, le buzzer c'est une alarme en cas de danger "fuite de gaz"

void setup(void)
{
  Serial.begin(9600); // définir la vitesse de transfert de données
  lcd.begin(16, 2);
  pinMode(PinGaz, INPUT); //mettre PinGaz comme entrée
  pinMode(redled1, OUTPUT); //mettre redled1 comme sortie
  pinMode(redled2, OUTPUT); //mettre redled2 comme sortie
  pinMode(buzzer, OUTPUT); // mettre buzzer comme sortie
  sensors.begin(); // commencer à recevoir les valeurs de la temperature
}

void loop(void)
{
  int sensorgaz= digitalRead(PinGaz);
  Serial.print("Requesting Temperatures from sensors: ");
  sensors.requestTemperatures(); // Send the command to get temperatures
  Serial.println("DONE");

  String sensor1 = String(sensors.getTempCByIndex(0),DEC); // convertir les valeurs numeriques en chaine de caracteres pour les envoyer au logiciel LabVIEW...
  String sensor2 = String(sensorgaz);
  Serial.println(temp1); // Envoyer les données au LabVIEW
  Serial.println(gaz);
  lcd.setCursor(0,0); // afficher la temperature et l'etat du gaz sur l'afficheur LCD
  lcd.println("Temp=");
  lcd.setCursor(5,0);
  lcd.print(sensors.getTempCByIndex(0),2);
  lcd.setCursor(10,0);
  lcd.println("C ");
  delay(100);
  if(sensors.getTempCByIndex(0)>30)
  { digitalWrite(redled2,HIGH);
    delay(100);
  }
  if(sensors.getTempCByIndex(0)<20)
  { digitalWrite(redled2,HIGH);
    delay(100);
  }
  else
  { digitalWrite(redled2,LOW);
    delay(100);
  }

  if(sensorgaz==1)
  {
    lcd.setCursor(0,1);
    lcd.print("Fuite de Gaz!!!!");
    digitalWrite(redled1,HIGH);
    tone(buzzer, 1000);
    delay(100);
  }
}
```

Annexe 5 : Code Arduino de l'éclairage automatique des 3 pièces aussi de l'éclairage manuel à partir de l'application HTML

```
const int salon = 13 ;
const int chambre = 12 ;
const int cuisine = 11 ;
const int motionPin0 = 7; // définir un variable motionPin pour recevoir les valeurs du capteur de mouvement
const int motionPin1 = 8;
const int motionPin2 = 9;
const int ldrPin0 = A0 ; // définir un variable ldrPin pour recevoir les valeurs du capteur de lumiere
const int ldrPin1 = A1 ;
const int ldrPin2 = A2 ;
int State0 = 0;
int State1 = 0;
int State2 = 0;
char income;
String inc;
void setup()
{
  pinMode(salon, OUTPUT);
  pinMode(cuisine, OUTPUT);
  pinMode(chambre, OUTPUT);
  pinMode(motionPin, INPUT);
  pinMode(ldrPin, INPUT);
  Serial.begin(9600);
}
void loop()
{
  int ldrStatus0 = analogRead(ldrPin0);
  int ldrStatus1 = analogRead(ldrPin1);
  int ldrStatus2 = analogRead(ldrPin2);
  State0=digitalRead(motionPin0);
  State1=digitalRead(motionPin1);
  State2=digitalRead(motionPin2);
  if(ldrStatus0 <= 400 && State0 == HIGH)
  {
    digitalWrite(salon, HIGH);
    delay(1000);
  }
  else
  {
    digitalWrite(salon, LOW);
    delay(1000);
  }
  if(ldrStatus1 <= 400 && State1 == HIGH)
  {
    digitalWrite(chambre, HIGH);
    delay(1000);
  }
  else
  {
    digitalWrite(chambre, LOW);
    delay(1000);
  }
  if(ldrStatus2 <= 400 && State2 == HIGH)
  {
    digitalWrite(cuisine, HIGH);
    delay(1000);
  }
  else
  {
    digitalWrite(cuisine, LOW);
    delay(1000);
  }
  // manuel
  if (Serial.available ()> 0)
  {
    inc="";
  }
  while(Serial.available()>0)
```

```
{
income=(byte) Serial.read();
if(income=='')
{
break;
}else
{
inc += income;
}
}
if(inc=="1")
{
digitalWrite(salon,HIGH);
delay(100);
}
else
{
//-----3-----
if(inc=="3")
{
digitalWrite(chambre,HIGH);
delay(100);
}
else
{
digitalWrite(chambre,LOW);
delay(100);
}
}
if(inc=="5")
{
digitalWrite(cuisine,HIGH);
delay(100);

}

else
{
digitalWrite(cuisine,LOW);
delay(100);
}
}
```

Annexe 6 : Code Arduino de commande de l'application Android

```
#include <SoftwareSerial.h>
#include <OneWire.h>
#include <DallasTemperature.h>
#include <AFMotor.h>
#define ONE_WIRE_BUS 8
OneWire oneWire(ONE_WIRE_BUS);
DallasTemperature sensors(&oneWire); // Pass our oneWire reference to Dallas Temperature.

int chambre = 7 ;
int cuisine = 6 ;
int salon = 5 ;
int ventilateur = 4;
int gaz = 2;
int state = 0;
AF_DCMotor motor(2);
const int buzzer = 9;

void setup()
{
  pinMode(chambre, OUTPUT);
  pinMode(cuisine, OUTPUT);
  pinMode(salon, OUTPUT);
  pinMode(gaz, INPUT);
  pinMode(ventilateur, OUTPUT);
  motor.setSpeed(155);
  pinMode(buzzer, OUTPUT);
  Serial2.begin(9600);
}

void loop()
{
  BT();
}

void BT()
{
  Serial.print("Requesting Temperatures from sensors: ");
  sensors.requestTemperatures(); // Send the command to get temperatures
  Serial.println("DONE");

  if(Serial2.available()>0)
  {state=Serial2.read();
  }
  if(state == '1')
  { digitalWrite(salon,HIGH);
  }
  if(state == '2')
  { digitalWrite(salon,LOW);
  }
  if(state == '3')
  { digitalWrite(chambre,HIGH);
  }
  if(state == '4')
  { digitalWrite(chambre,LOW);
  }
  if(state == '5')
  { digitalWrite(cuisine,HIGH);
  }
  if(state == '6')
  { digitalWrite(cuisine,LOW);
  }
  if(state == '7')
  { digitalWrite(ventilateur,HIGH);
  }
  if(state == '8')
  { digitalWrite(ventilateur,LOW);
  }

  if(state == '9')
  { motor2.run(FORWARD);
    delay(3000);
    motor2.run(RELEASE);
    delay(2000);
  }
  if(state == '10')
  { motor.run(BACKWARD);
    delay(3000);
    motor.run(RELEASE);
    delay(2000);
  }

  Serial2.print(sensors.getTempCByIndex(0));
  Serial2.print("|");
  if(gaz==LOW)
  {Serial2.print("Yes");}
  tone(buzzer, 1000);
}
else
{Serial2.print("No");
noTone(buzzer);
}
}
delay(500);
}
```

Annexe 7 : Code Arduino de contrôle à distance via la télécommande infrarouge

```
#include "IRremote.h"
#include <AFMotor.h>
int chambre = 7 ;
int cuisine = 6 ;
int salon = 5 ;
int ventilateur = 4;
int gaz = 2;
int state = 0;
AF_DCMotor motor(2);
int receiver = 11; // Signal Pin of IR receiver to Arduino Digital Pin 11
IRrecv irrecv(receiver); // create instance of 'irrecv'
decode_results results; // create instance of 'decode_results'
void setup()
{
  Serial.begin(9600);
  Serial.println("IR Receiver Button Decode");
  irrecv.enableIRin(); // Start the receiver
}

void loop() /*----( LOOP: RUNS CONSTANTLY )----*/
{
  if (irrecv.decode(&results)) // have we received an IR signal?
  {
    translateIR();
    irrecv.resume(); // receive the next value
  }
}

void translateIR() // takes action based on IR code received
{
  switch(results.value)
  {
    case 0xFF6897: state=1; break;
    case 0xFF30CF: state=4; break;
    case 0xFF18E7: state=5; break;
    case 0xFF7A85: state=6; break;
    case 0xFF10EF: state=7; break;
    case 0xFF38C7: state=8; break;
    case 0xFF5AA5: state=9; break;
    case 0xFF42BD: state=10; break;
    default:
      state=0;
  }
  if(state == '1')
  { digitalWrite(salon,HIGH);
  }
  if(state == '2')
  { digitalWrite(salon,LOW);
  }
  if(state == '9')
  { motor2.run(FORWARD);
    delay(3000);
    motor2.run(RELEASE);
    delay(2000);
  }
  if(state == '10')
  { motor.run(BACKWARD);
    delay(3000);
    motor.run(RELEASE);
    delay(2000);
  }
  delay(500);

  { digitalWrite(ventilateur,LOW);
  }
  { motor2.run(FORWARD);
    delay(3000);
    motor2.run(RELEASE);
    delay(2000);
  }
  if(state == '10')
  { motor.run(BACKWARD);
    delay(3000);
    motor.run(RELEASE);
    delay(2000);
  }

  Serial2.print(sensors.getTempCByIndex(0));
  Serial2.print("!");
  if(gaz==LOW)
  {Serial2.print("Yes");}
  tone(buzzer, 1000);

}
else
{Serial2.print("No");
noTone(buzzer);

}
delay(500);
}
```

Bibliographie

- <http://air.imag.fr/index.php/Arduino>
- http://air.imag.fr/index.php/PIR_Motion_Sensor
- http://air.imag.fr/index.php/135038_Arduino_Flame_Detection_Sensor_Module
- http://air.imag.fr/index.php/Gas_Sensors
- <http://www.ajolly.fr/fr/conseil-et-deloppement-mainmenu-27/19-LabVIEW.html>
- <https://knowledge.ni.com/KnowledgeArticleDetails?id=kA00Z0000019LsaSAE>
- <http://fablab.ensimag.fr/index.php/PILBI-2013-Team2>
- <https://www.arduino.cc/en/Guide/ArduinoMega2560>
- <http://automacile.fr/ds18b20-capteur-de-temperature-one-wire-arduino/>
- <https://www.carnetdumaker.net/articles/mesurer-une-temperature-avec-un-capteur-1-wire-ds18b20-et-une-carte-arduino-genuino/>
- <https://learn.adafruit.com/pir-passive-infrared-proximity-motion-sensor>
- <https://www.carnetdumaker.net/articles/mesurer-la-luminosite-ambiante-avec-une-photoresistance-et-une-carte-arduino-genuino/#quest-ce-quune-photoresistance>
- https://www.electronics-tutorials.ws/io/io_4.html
- http://nalhossri.free.fr/Librairies%20pour%20CODE%20WARRIOR/clavier_matriciel/clavier_matriciel/decodage_clavier_matriciel.html
- http://wiki.seeedstudio.com/Grove-Gas_Sensor-MQ2/
- <https://eskimon.fr/tuto-arduino-601-le-moteur-%C3%A0-courant-continu>
- https://wiki.eprolabs.com/index.php?title=Bluetooth_Module_HC-05
- <https://wiki.eprolabs.com/index.php?title=LCD>
- <http://roboticus.org/articles/15-la-telecommande-infrarouge-theorie>
- <https://electronics.howstuffworks.com/remote-control2.htm>
- <http://teen-code.com/2016/06/03/apprendre-a-coder-avec-app-inventor/>