

République Algérienne Démocratique et Populaire
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique
Université Abderrahmane Mira Bejaïa
Faculté de Technologie
Département de Génie Electrique



Mémoire de fin d'études

En vue de l'obtention du diplôme de Master en Automatique

Thème

Gestion et commande d'une maison intelligente

Encadré et dirigé par :

M^r. HADJI Slimane

Réalisé par :

ACHOURI Sara

KHIMA Wissem

Devant le jury composé de :

M^{me}. GAGAOUA

M^r. YAHIAOUI

Soutenu le :

Année Universitaire : 2019/2020

Remerciements

En premier lieu, nous remercions le bon Dieu tout puissant de nous avoir donné la force et la volonté durant nos études et pendant la réalisation de ce projet.

Nous voudrions remercier aussi notre encadreur Mr HADJI pour son encadrement.

Comme nous tenons à remercier les membres du jury d'avoir accepté d'examiner et juger notre travail.

Un remerciement particulier à nos familles pour nous avoir soutenus, accompagnés et nous avoir permis d'en arriver là.

Enfin, nous remercions tous nos amis pour leur aide et leurs encouragements, en particulier Nanou, Koukou et Sid. Et tous ceux qui ont contribué de près ou de loin à la réalisation de ce projet.

Dédicaces

A la mémoire de mon père puisse-t-il reposer en paix.

A ma mère et mon frère qui m'ont soutenu et encouragé durant mes études.

A mes amis qui ont toujours été présents.

Sara.

Dédicaces

Je dédie ce travail

A ma très chère mère et mon très cher père pour tout leur soutien, patience,
sacrifice et leur amour.

A mes frères pour leurs encouragements.

A tous mes amis pour leur aide et leur présence.

Wissem.

Sommaire

Introduction générale	1
1 Chapitre 1 : Domotique	4
1.1 Introduction	5
1.2 Définition de la domotique	5
1.2.1 Maison intelligente	5
1.3 Domaine d'application de la domotique.....	6
1.3.1 Economie d'énergie.....	6
1.3.2 Sécurité et protection.....	7
1.3.3 Santé et bien-être.....	7
1.4 Fonctionnement de la domotique.....	7
1.4.1 Technologie Bus filaire	7
1.4.2 Technologie de courant porteur de ligne (CPL).....	8
1.4.3 Technologie sans fil ou radiofréquence	8
1.5 Conclusion.....	8
2 Chapitre 2 : Généralités sur l'Arduino	9
2.1 Introduction.....	10
2.2 Architecture et caractéristiques.....	10
2.2.1 Microcontrôleur.....	10
2.2.2 Architecture	11
2.2.3 Caractéristiques des différentes cartes Arduino.....	12
2.2.3.1 La carte LEONARDO	12
2.2.3.2 La carte NANO	12
2.2.3.3 La carte UNO.....	13
2.2.3.4 La carte MEGA 2560.....	13
2.3 Modules et Shields	13
2.3.1 Shields	14
2.3.2 Modules	14
2.3.3 La différence entre module et shield	15

2.4	Langages de programmation.....	17
2.5	Conclusion.....	21
3	Chapitre 3 : Conception et réalisation.....	22
3.1	Introduction.....	23
3.2	Fonctions intégrées dans le système	23
3.2.1	Serrure codée.....	23
3.2.1.1	Fonctionnement de la serrure.....	24
3.2.1.2	Organigramme de la serrure codée	25
3.2.2	Garage automatique.....	25
3.2.2.1	Fonctionnement du garage automatique	25
3.2.2.2	Organigramme du garage automatique.....	26
3.2.3	Eclairage intérieur	26
3.2.3.1	Fonctionnement de l'éclairage intérieur	27
3.2.3.2	Organigramme de l'éclairage intérieur	28
3.2.4	Eclairage extérieur.....	28
3.2.4.1	Fonctionnement de l'éclairage extérieur.....	29
3.2.4.2	Organigramme de l'éclairage extérieur	29
3.2.5	Alarme incendie	29
3.2.5.1	Fonctionnement de l'alarme	29
3.2.5.2	L'organigramme de l'alarme	30
3.3	Matériel utilisé	31
3.3.1	Serrure codée.....	31
3.3.2	Garage automatique.....	33
3.3.3	Eclairage intérieur	34
3.3.4	Eclairage extérieur.....	35
3.3.5	Alarme incendie	36
3.4	Schémas synoptique.....	38
3.5	Branchement des applications	39
3.5.1	Serrure codée.....	39
3.5.2	Garage automatique.....	40
3.5.3	Eclairage intérieur	41

Sommaire

3.5.4	Eclairage extérieur.....	42
3.5.5	Alarme incendie	43
3.6	Réalisation de la maquette	44
3.7	Application Android (App Inventor)	44
3.8	Conclusion.....	47
	Conclusion générale.....	48
	Références	50
	Annexes.....	53

Liste des figures

Fig.1. 1. Maison intelligente.....	6
Fig.2. 1. Circuits interne de la carte MEGA.....	11
Fig.2. 2. Carte Arduino MEGA2560.....	12
Fig.2. 3. Exemple d'un shield/module.....	14
Fig.2. 4. Capteur de température.....	16
Fig.2. 5. Schéma électrique d'un capteur de température.....	16
Fig.2. 6. Programme et moniteur série.....	17
Fig.2. 7. Interface générale de l'Arduino.....	18
Fig.2. 8. Schéma électrique de la LED.....	20
Fig.2. 9. Programme Arduino.....	21
Fig.3. 1. Organigramme de la serrure codée.....	25
Fig.3. 2. Organigramme de garage.....	26
Fig.3. 3. Organigramme de l'éclairage intérieur.....	28
Fig.3. 4. Organigramme de l'éclairage extérieur.....	29
Fig.3. 5. Organigramme de l'alarme.....	30
Fig.3. 6. Ecran LCD 16x2.....	31
Fig.3. 7. Module Bluetooth.....	32
Fig.3. 8. Le servomoteur.....	32
Fig.3. 9. Télécommande.....	33
Fig.3. 10. Capteur infrarouge.....	33
Fig.3. 11. Schéma du bouton poussoir.....	34
Fig.3. 12. Capteur de mouvement.....	34
Fig.3. 13. Relais 5V.....	35
Fig.3. 14. Photorésistance.....	36
Fig.3. 15. LED.....	36
Fig.3. 16. Capteur de gaz et fumée.....	37
Fig.3. 17. Buzzer.....	37
Fig.3. 18. Schéma synoptique de la carte UNO.....	38
Fig.3. 19. Schéma synoptique de la carte MEGA.....	38
Fig.3. 20. Schéma électrique de la serrure.....	39
Fig.3. 21. Schéma électrique du garage automatique.....	40
Fig.3. 22. Schéma électrique de l'éclairage intérieur.....	41
Fig.3. 23. Schéma électrique de l'éclairage extérieur.....	42
Fig.3. 24. Schéma électrique de l'alarme incendie.....	43
Fig.3. 25. La maquette vue d'ensemble.....	44
Fig.3. 26. La maquette alimentée.....	44
Fig.3. 27. Interface de design d'App inventor.....	45
Fig.3. 28. Interface de programmation.....	46
Fig.3. 29. Fenêtre d'identification.....	46
Fig.3. 30. Fenêtre de déverrouillage.....	47

Liste des tableaux

Tableau 3. 1. Branchement de la serrure.....	39
Tableau 3. 2. Branchement du garage automatique.....	40
Tableau 3. 3. Branchement de l'éclairage intérieur.....	41
Tableau 3. 4. Branchement de l'éclairage extérieur.....	42
Tableau 3. 5. Branchement de l'alarme incendie.....	43

Introduction générale

Introduction générale.

Le progrès et l'évolution dans le monde de l'électronique ainsi que dans le domaine de la technologie ont donné naissance à l'internet des objets (Internet Of Things IoT), ces derniers sont capables de communiquer entre eux, d'apprendre et s'adapter à leur environnement d'utilisation, et cela toujours dans l'optique d'améliorer le niveau de vie et de satisfaire les besoins du quotidien.

Cette avancée technologique que l'on retrouve dans de multiples domaines résulte de la propagation de moyens de communications, avec des composants de plus en plus petits et faciles à utiliser et grâce aux systèmes embarqués à faible coût, ainsi que la disponibilité des plateformes open source, ce qui a attiré l'attention des experts et passionnés, a titillé la curiosité des amateurs et les a poussés à créer des projets, développer des idées et mettre au point des langages de programmations pour offrir une large diversité .

Parmi ces domaines, on peut citer la domotique ; ces nouvelles habitations dites intelligentes, qui ont pour principes d'établir une liaison entre les appareils et permettre la communication de ces derniers avec l'utilisateur, dans le but de rendre la vie facile aux occupants de ces logements, et cela dans différents aspects. Ce genre de maisons profitent beaucoup plus aux personnes âgées et à mobilité réduite pour tous les avantages qu'elles proposent à savoir : l'assistance, la sécurité des habitants et des biens, la réduction de la consommation d'énergie et enfin le confort et la tranquillité. En utilisant une télécommande ou bien un smartphone/tablette on arrive à contrôler l'ouverture et fermeture des fenêtres/volets, verrouiller les portes, allumer ou éteindre les lumières ou la télévision, augmenter ou baisser la température, tous cela en un seul clic.

En outre on peut utiliser des capteurs qui activent des appareils spécifiques une fois qu'ils ont détecté des changements ou une défaillance, à titre d'exemple un capteur de gaz et fumée qui déclenche une alarme s'il y'a une fuite de gaz ou un incendie. De nos jours, il existe même des kits complets que l'on trouve sous le nom de box domotique ou bien centrale domotique qui relie les différents capteurs et appareil d'une maison par un réseau internet.

Dans le cadre de notre formation de Master en Automatique, notre projet a pour but la réalisation d'une maison intelligente réduite basée sur la carte Arduino MEGA 2560 et la carte UNO. La maquette comptera cinq différents scénarios : une serrure codée avec écran LCD, un garage automatique, un éclairage intérieur avec un capteur de mouvement PIR et un éclairage extérieur, enfin un détecteur de fumée et de gaz MQ-2 ainsi que la conception d'une application Android pour déverrouiller la serrure.

Introduction générale.

Notre mémoire se présente en trois chapitres, dans le premier nous définirons la domotique ainsi que la maison intelligente, avec les domaines d'utilisation et les différentes technologies pour concevoir ces maisons.

Le deuxième chapitre traitera des généralités sur l'Arduino, nous verrons les différents types de cartes et leurs caractéristiques et les nombreux langages et interfaces de programmation.

Quant au troisième chapitre, nous allons expliquer les fonctions intégrées qui constitueront notre maison intelligente : fonctionnements et schémas représentatifs. Nous décrirons en détail le matériel utilisé avec les différents circuits et branchements nécessaires à la réalisation des scénarios proposés, de la conception de la maquette et également nous verrons le développement de l'application Android sur smartphone.

Enfin, nous terminerons sur une conclusion générale dans laquelle on parlera de nos impressions et le savoir que nous aurons acquis au terme de ce projet.

Chapitre 1 : Domotique

1.1 Introduction

Depuis la nuit des temps, l'Homme n'a cessé d'innover et chercher ce qu'il y'a de mieux toujours dans le but de faciliter son quotidien et d'améliorer son confort ; cela va s'en dire que le foyer dans lequel il vit, lui aussi a connu des évolutions à travers le temps donnant ainsi naissance aux maisons intelligentes.

Le développement de l'automatisme, la télécommunication, l'informatique et l'électronique ont permis l'apparition de maisons communicantes, qui offrent plusieurs fonctions pour différents usages, mais qui ont toujours pour but de faciliter la vie de tous les jours, réduire nos consommations d'énergies et apporter un sentiment de sécurité. [1]

Dans ce chapitre, nous aborderons quelques notions sur la domotique et la maison intelligente, les domaines d'application ainsi que le fonctionnement de la domotique.

1.2 Définition de la domotique

La domotique est l'intégration des techniques modernes telles que l'automatisation dans les maisons, d'où l'étymologie du nom « Domus » du latin qui signifie maison et le suffixe « tique » pour les techniques utilisées.

Appelée aussi maison intelligente, elle a pour principe de faire communiquer les équipements électriques d'un même domaine [2], et cela en utilisant différents protocoles tel que des câblages de contrôle ou bien des signaux intégrés dans le circuit électrique existant déjà dans la maison ou alors des signaux radiofréquence (RF), dans certains cas c'est la combinaison de plusieurs méthodes, avec toutes les tâches de contrôles sont faites par un microprocesseur. [3]

Parce que la maison intelligente regroupe un nombre important de disciplines en plus des travaux de bâtiments traditionnels, plusieurs personnes l'associent comme maison de luxe alors que de nos jours ce domaine a énormément évolué et donc on peut trouver des solutions accessibles pour tous budgets. [4]

1.2.1 Maison intelligente

Une maison intelligente est une habitation dont plusieurs éléments tels que le chauffage, l'éclairage et le système de sécurité, sont contrôlés par un ensemble de boutons et de télécommandes depuis des applications mobiles, disponibles sur smartphone, tablette ou sur ordinateur.

Elle a la capacité d'augmenter le confort de l'habitant et sa sécurité surtout quand il s'agit des personnes avec une capacité limitée de mouvement. Mais pour cela il faut rassembler et intégrer un ensemble de techniques de l'électronique, de l'informatique d'automatisme et de télécommunication afin d'avoir une surveillance en continue de la maison et pouvoir piloter de près ou de loin. [5] [6]

Dans la figure (Fig.1.1) ci-dessous nous avons illustré une maison intelligente.



Fig.1. 1. Maison intelligente.

1.3 Domaine d'application de la domotique

Parmi les différents services que peut offrir la domotique il y'a principalement trois critères qu'on retrouve le plus souvent dans les maisons intelligentes :

1.3.1 Economie d'énergie

La domotique a pour but de promouvoir la consommation plus modérée d'énergie, afin de réduire le montant de la facture mais pas que ; il est tout aussi important de diminuer cette consommation d'un point de vue écologique. De ce fait l'économie d'énergie s'applique au quotidien en maîtrisant la consommation d'électricité et cela grâce aux volets électriques qui s'ouvrent et se ferment afin de laisser pénétrer la lumière extérieure ou bien l'éclairage automatique, ainsi on aura plus à se soucier de laisser la lumière allumée en quittant une

pièce. La gestion du chauffage se fait avec des capteurs de température qui règlent le thermostat du chauffage avec précision ou l'ouverture et fermeture des fenêtres selon la météo, on retrouve aussi l'économie d'eau notamment dans les jardins qui utilisent des systèmes d'arrosage automatique.

1.3.2 Sécurité et protection

Afin d'éviter tout accident malencontreux, la sécurité est devenue l'un des facteurs les plus importants dans la domotique, et parmi les dangers auxquels on peut faire face, il y'a le risque d'intrusion dans la maison, en d'autres termes un cambriolage et pour éviter ce genre de situation des moyens ont été mis en place tel que les caméras de sécurités, des détecteurs de mouvement, des alarmes, serrures codées, etc.

On trouve aussi des risques techniques tel que des pannes ou des défaillances c'est-à-dire fuite de gaz, incendie ou bien inondation c'est pour cela que des dispositifs peuvent être installé comme les détecteurs et alarmes pour avertir les propriétaires [1].

1.3.3 Santé et bien-être

Le sentiment de sécurité que proposent les maisons intelligentes ainsi que cette notion d'utilisation d'énergie réglée selon le besoin et l'envie offrent un niveau de confort optimal aux habitants de ces maisons.

Plus encore elles facilitent le quotidien aux personnes avec des besoins particuliers ou une assistance tel que les personnes âgées, handicapées ou encore atteintes de maladies neurodégénératives comme l'Alzheimer, ces derniers peuvent rester chez eux en toute sécurité une fois la maison programmée et adaptée à leurs besoins. [1]

1.4 Fonctionnement de la domotique

Le fonctionnement de la domotique se résume au transfert d'informations entre la partie commande de la domotique et sa partie opérative. Cela se fait grâce à des capteurs qui captent des informations et d'actionner par la suite les commandes grâce a des actionneurs. Pour cela trois différentes technologies de liaison sont possibles : [7]

1.4.1 Technologie Bus filaire

C'est une technologie basée sur le câbles qui permet la communication entre les composants afin d'échanger et de traiter les informations. Celle-ci est donc envoyée par l'entrée vers des récepteurs de sortie qui exécutent la tâche.

L'installation de ce dispositif est composée d'un réseau bus filaire qui relie les capteurs aux actionneurs, et d'un réseau d'alimentation qui relie les actionneurs au courant fort.

Cette technologie est très fiable, et permet de mettre plusieurs modules sur un seul bus, c'est pourquoi nous avons choisis de réaliser notre projet à l'aide cette dernière. [6] [7]

1.4.2 Technologie de courant porteur de ligne (CPL)

Cette technologie, utilise l'installation électrique de la maison pour l'échange d'informations entre l'émetteur et le récepteur. L'avantage de cette technologie est qu'elle ne nécessite pas d'endroit précis pour l'installation et permet de communiquer sur de très longues distances.

La fiabilité de la technologie CPL reste critiquable. De plus, Les protocoles utilisant les courants porteurs sont très rapidement perturbés par l'environnement et leur équipement est très coûteux. [7]

1.4.3 Technologie sans fil ou radiofréquence

Avec cette technologie, l'information est échangée sans fil entre l'émetteur et le récepteur. Elle utilise plusieurs supports technologiques, comme la radiofréquence, infrarouge, ou Bluetooth et wifi qui sont des technologies radio. La mise en œuvre de cette technologie est facile et il est conseillé de ne pas mixer le sans-fil avec une autre technologie car cela peut endommager l'installation ou la qualité de communication.

Les protocoles sans fil sont appréciés pour la liberté de placement des capteurs et interrupteurs sans avoir à câbler. Seulement, le principal défaut de ces protocoles reste la durée de vie. [6] [7]

1.5 Conclusion

Dans ce chapitre, nous avons pu définir la domotique ainsi que les commodités qu'elle offre dans la vie quotidienne et l'assistance qu'elle propose à une catégorie de personnes dans le besoin.

Par la suite, nous avons mentionné les domaines de la domotique et expliqué les trois types de fonctionnements avec lesquels sont réalisées ces maisons.

Chapitre 2 : Généralités sur l'Arduino

2.1 Introduction

Arduino est une carte électronique dans laquelle se trouve un microcontrôleur que nous pouvons programmer grâce au langage Arduino IDE pour analyser et produire des signaux électriques de manière assez simple afin d'effectuer diverse tâches. [8]

C'est une plateforme basée sur une interface entrée/sortie (analogique et numérique) qui permet de traiter l'information provenant des capteurs pour commander des actionneurs et peut être utilisée pour construire des objets interactifs indépendants, ou bien peut être connectée à un ordinateur pour communiquer avec des logiciels. [8]

Dans ce chapitre nous nous intéresserons à l'architecture ainsi qu'au différentes caractéristiques de plusieurs modèles de carte Arduino ainsi que les langages de programmations qui peuvent être utilisée, nous aborderons aussi les différents types de shields et modules pour mieux comprendre la différence entre les deux, tout cela accompagné de quelques exemples pratiques.

2.2 Architecture et caractéristiques

2.2.1 Microcontrôleur

Il s'agit d'un élément indispensable considéré comme un organe vital dans l'informatique embarquée, grâce auquel les cartes électroniques logiques ont été simplifiées et ont moins de composants.

Le microcontrôleur ATmega 2560 appartient à la famille 8 bits qui sont connus pour leur puissance de calcul, on les retrouve dans l'audio et vidéo, l'électroménager, l'automobile et la domotique.

Ce dernier est constitué d'un microprocesseur qui lui est responsable du traitement des informations et l'envoi des ordres.

Doté d'une mémoire vive (RAM) dans laquelle sont stockées les données temporaires (mémoire de données) en d'autres termes volatile. On compte notamment une mémoire morte (ROM) dite mémoire programmable, celle-ci contient les instructions du programme pilotant le microcontrôleur qui est en général toujours le même donc dans ce cas n'est pas volatile et/ou EEPROM, EPROM et FLASH. [1]

Enfin, les ressources auxiliaires ou encore périphériques, qui correspondent aux entrées/sorties, timers ou bien les convertisseurs CNA/CAN. [11]

2.2.2 Architecture

comme le montre la figure (Fig.2.2), les cartes Arduino sont généralement constituées de quelques éléments fondamentaux que l'on peut retrouver dans la plupart d'entre elles, tel que le composant de base le **microcontrôleur** qui dans certains cas est préprogrammé, ce qui simplifie le chargement des programmes mémoire flash incorporé dans la carte, il permet également de stocker le programme transféré grâce au **câble USB** et de l'exécuter par les capteurs et les actionneurs branchés aux **entrées/sorties**, lors du transfert de données des **LED de signalisations** (*Tx* et *Rx*) clignotent, ainsi qu'une LED qui elle, indique que la carte est alimentée par le câble USB ou bien par une pile, batterie ou adaptateur secteur qui sont branchés à la **prise jack**. On retrouve aussi le **port ICSP** (in-circuit serial programming) qui est une interface de programmation des microcontrôleurs. Enfin la carte compte un **bouton reset** (remise à zéro) pour réinitialiser cette dernière. [9]

La figure suivante (Fig.2.1) représente l'architecture interne de la carte MEGA2560.

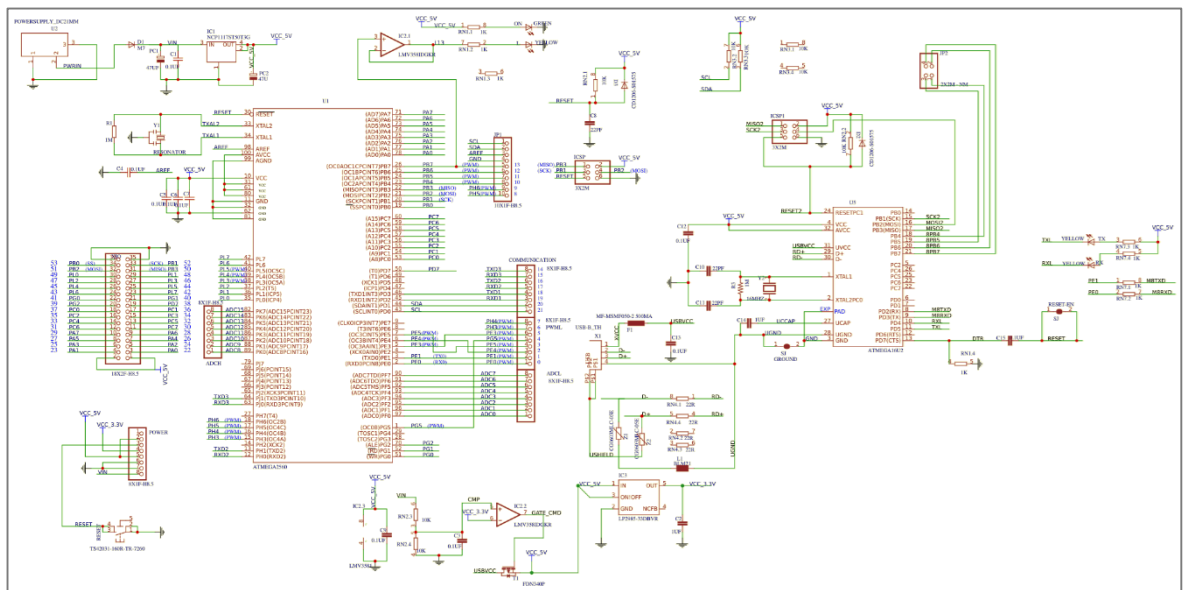


Fig.2. 1. Circuits interne de la carte MEGA2560.

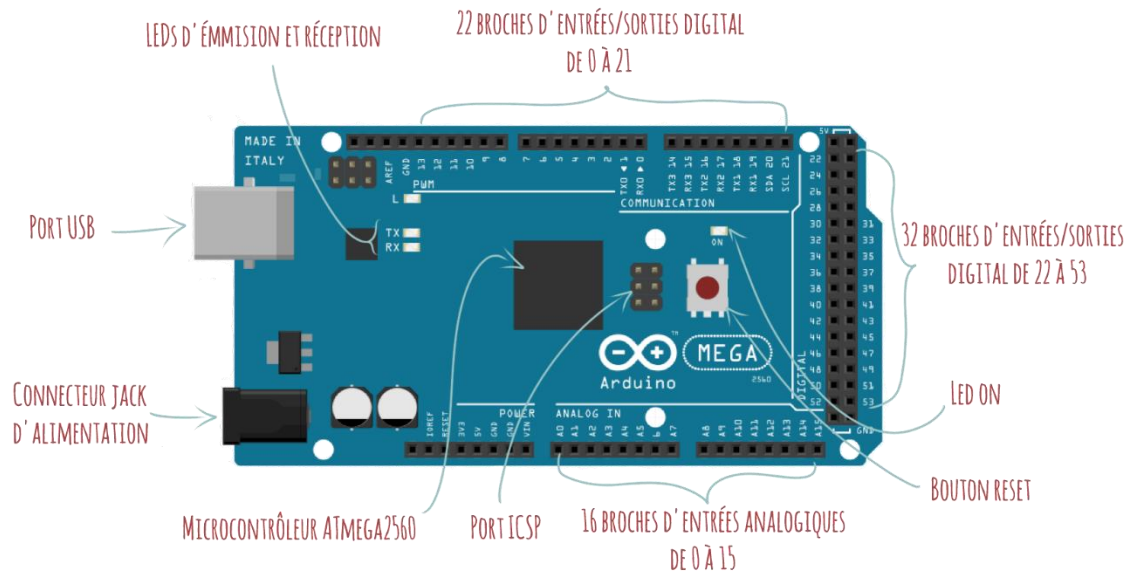


Fig.2. 2. Carte Arduino MEGA2560.

2.2.3 Caractéristiques des différentes cartes Arduino

Il existe dans le marché plusieurs types de cartes, celles-ci différent par leur taille, puissance de microcontrôleur et leur consommation. L'avantage avec ses cartes c'est qu'elles sont toutes des open source c'est-à-dire disponibles gratuitement au public avec possibilité de modification du programme. Voici quelques-unes :

2.2.3.1 La carte LEONARDO

La carte Arduino LEONARDO est basée sur un microcontrôleur ATmega32u4 cadencé à 16 MHz permettant la gestion du port USB par un seul processeur Avec 32 ko de mémoire flash, 2,5 ko de mémoire SRAM, 1 ko de mémoire EEPROM et 20 broches d'entrées/sorties dont 7 sont des sorties PWM ainsi que 12 entrées analogiques. C'est la carte qui est prévue de succéder à la carte Arduino UNO en présentant des caractéristiques équivalentes mais avec une ergonomie revue et une stabilité plus éprouvée. [12]

2.2.3.2 La carte NANO

C'est l'une des plus petites cartes de microcontrôleur de la gamme Arduino. Sa taille et son poids réduit lui permettent d'être utilisée des espaces petits. Elle possède un microcontrôleur ATmega328 à 16 MHz, comprend 32 Ko de mémoire flash 1 Ko d'EEPROM, 2 Ko de RAM, 14 entrées/sorties numériques, 6 entrées analogiques et une tension de fonctionnement de 5V. [10]

2.2.3.3 La carte UNO

Il s'agit du modèle le plus répandu. Livré avec une puce ATmega328 qui convient aux tâches relativement simples en raison de sa mémoire flash et SRAM et également un nombre d'entrées/sorties limitées qui est 14 les entrées/sorties numériques dont 6 PWM et 6 entrées analogiques. L'un des avantages de ce microcontrôleur, est sa facilité d'utilisation. La carte se connecte facilement à un ordinateur via un câble USB fourni. Elle fonctionne à une fréquence de 16 MHz, elle comprend 32 Ko de mémoire programme, 2 Ko de RAM, 1 Ko d'EEPROM et un rail d'alimentation de 5V et 3,3V.

L'Arduino UNO diffère d'Arduino MEGA uniquement par le nombre de broches et la tailles des mémoires. [10] [11]

2.2.3.4 La carte MEGA 2560

La carte MEGA2560 est la plus utilisée après la UNO, grâce au nombre important d'entrées/sorties qu'elle propose, estimées à 54 broches numériques dont 14 peuvent être utilisées comme sorties PWM et 16 entrées analogiques, elle offre une configuration plus complète que le modèle UNO et permet une importante marge de manœuvres et des montages plus poussées. Et sa connexion à l'ordinateur via port USB, permet une programmation complète et facile par l'interface du logiciel.

Elle fonctionne grâce à un microcontrôleur ATmega2560, avec une fréquence d'horloge de l'oscillateur à quartz égale à 16 MHz et à une tension de 5V, sa tension d'entrée recommandée est comprise entre 7 et 12V et la tension limite est entre 6V et 20V.

Dotée d'une mémoire flash de 256 Ko pour stocker les codes dont 4 Ko sont utilisée par bootloader, d'une SRAM de 8 Ko ce qui la rend relativement rapide et une EEPROM de 4 Ko. Toutes ces caractéristiques lui permettent d'exploiter des algorithmes plus complexes. [7] [10] [13].

2.3 Modules et Shields

Ce sont des extensions qu'on ajoute aux cartes Arduino qui nous permettent d'obtenir plus de fonctionnalités que la carte elle-même ne possède pas. [11]

La figure (*Fig.2.3*) montre l'exemple d'un shield Wifi et d'un module capteur d'obstacle.

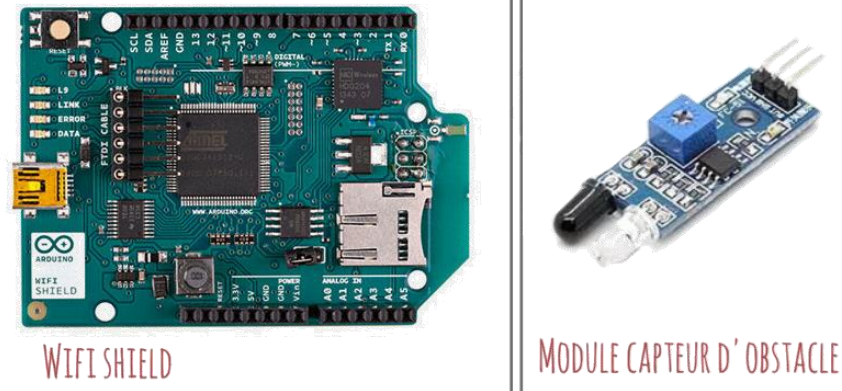


Fig.2. 3. Exemple d'un shield/module.

2.3.1 Shields

Les shields sont des cartes qu'on place au-dessus de la carte Arduino pour former un seul bloc, les cartes UNO, MEGA et la LEONARDO utilisent les mêmes shields, contrairement à la carte nano qui elle, est plus petite donc cette dernière a des shields différents conçus spécialement pour elle. On distingue plusieurs shields : [11]

- Shield Ethernet :

Cela nous permet de connecter la carte Arduino au réseau Internet, pour cela il suffit juste de brancher le shield sur la carte.

- Shield GSM :

Ce shield nous permet de connecter notre carte Arduino à un réseau de téléphone mobile ainsi que de composer un numéro de téléphone ou encore envoyer des messages.

- Shield moteur :

Celui-ci fait en sorte d'avoir un contrôle plus facile et interagit plus précisément avec les moteurs.

- Shield capteur :

Ce shield nous permet de lire plus facilement les informations de différents capteurs.

2.3.2 Modules

Tous comme les shields, les modules offrent des fonctions supplémentaires qui se connectent aux cartes Arduino avec des fils conducteurs ce qui les rend plus pratiques, bien plus petits et souvent beaucoup moins chers. On peut les trouver sous forme de capteurs ou d'actionneurs, voici quelques exemples de modules : [14] [15]

- Capteur d'obstacle :

Souvent utilisé dans la robotique, il sert à détecter les objets à courte distance (entre 2 à 30cm et un angle de 35°) et cela se fait en émettant un signal comme sortie numérique et l'indicateur (LED verte) s'allume.

- Capteur microphone :

C'est un détecteur de son à deux sorties ; analogique et numérique, cette dernière envoie un signal élevé lorsque l'intensité sonore atteint un certain seuil.

- Capteur de pression barométrique numérique :

Ce module convertit la pression mesurée en altitude, grâce à ce capteur on peut déterminer la hauteur d'un robot ou d'un projectile, il est aussi utilisé dans la mesure de température et la détection d'humidité.

- Capteur de vitesse :

Un tachymètre est un module assez simple néanmoins qui détecte la vitesse d'un moteur, le nombre d'impulsion, la limite de position, etc.

2.3.3 La différence entre module et shield

L'avantage d'utiliser un shield est que ce dernier n'a pas besoin de câblage, il se branche directement à la carte contrairement au module où on doit s'assurer que les broches sont correctement connectées et mises en place, il se peut aussi qu'on ait besoin d'ajouter 3.3V pour supporter la demande du module car la broche de la carte ne le peut pas. D'un autre côté, les modules sont plus petits, moins chers et nous offrent une certaine liberté lorsqu'il est question de les connecter à la carte vu que l'on ne nous impose pas des broches spécifiques comme on le fait pour les shields, ce qui peut poser problème si on utilise ces mêmes broches pour autre choses. [15]

Exemple d'un capteur de température LM35

Ce module est un capteur de température linéaire dont la sortie est une tension proportionnelle à la valeur de la température c'est-à-dire qu'il produit une augmentation de température de $10mV$ par degré Celsius, par exemple si la tension de sortie est de $345mV$, cela veut dire que la valeur de la température est $34,5^{\circ}C$.

Le capteur se présente avec trois broches comme le montre la figure (fig.2.4) :

Chapitre 2 : Généralités sur l'Arduino.

On déclare le **capteur** en tant qu'entrée et on initialise la communication série à 9600, ensuite la valeur reçue à partir du capteur est lue par la variable **tension** grâce à l'instruction **analogRead**.

Enfin, on utilise l'instruction **serial.print** pour visualiser les valeurs lues dans le moniteur série : température en degré Celsius, température en Fahrenheit et tension, le délai d'une seconde représente la fréquence d'échantillonnage de l'entrée.

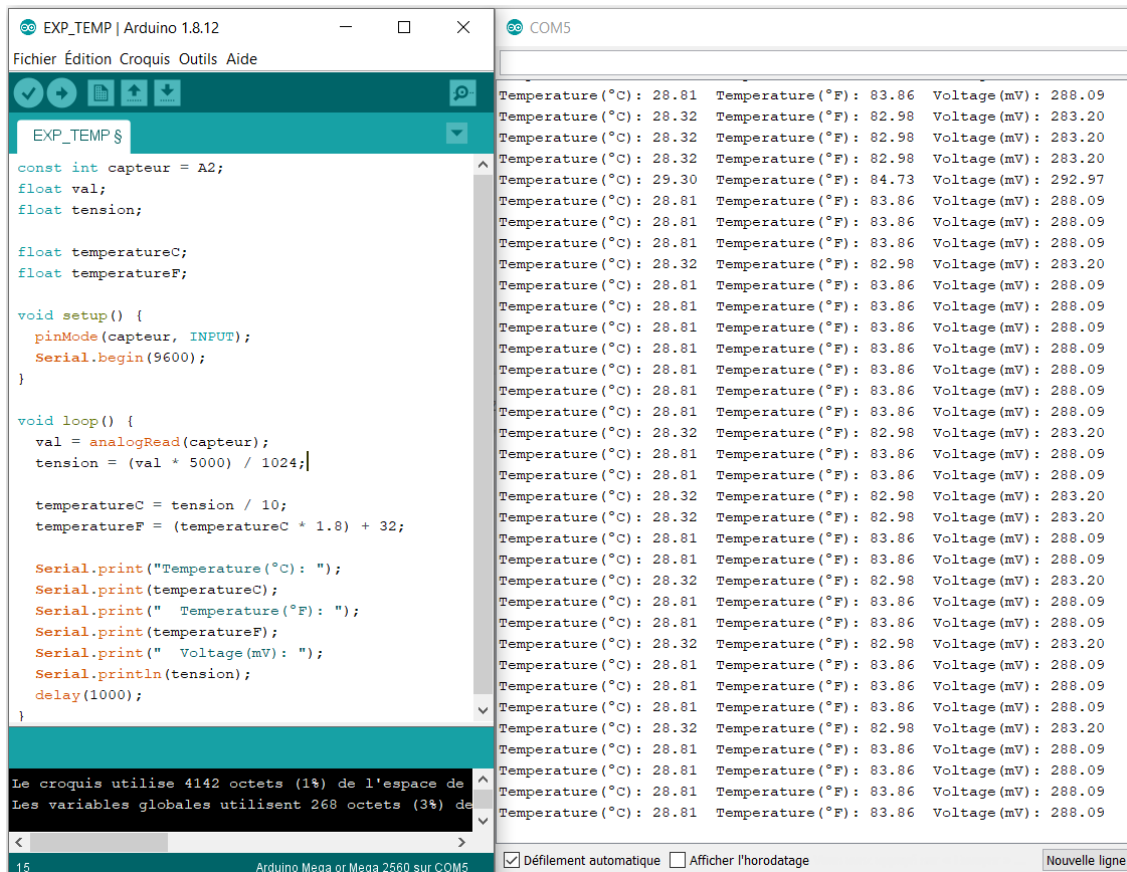


Fig.2. 6. Programme et moniteur série.

2.4 Langages de programmation

Généralement, le langage Arduino est le langage de programmation le plus utilisé lorsqu'il est question de programmer les cartes Arduino. En raison de sa facilité d'utilisation et ses bibliothèques complètes qui sont prêtes à être utilisées, ce dernier est disponible sur plusieurs systèmes d'exploitation tels que Windows, Mac OS et Linux,

Le langage étant simple à manier permet aux débutants de se lancer sur la programmation et le développement du projet sans être obligé de suivre une formation avancée, pour cela

Chapitre 2 : Généralités sur l'Arduino.

nous avons juste à écrire le code, le compiler (vérifier) ensuite téléverser le programme vers la carte.

Dans la (figure 2.7) ci-dessous on peut voir à quoi ressemble l'interface générale de l'IDE. Celle-ci est constituée de plusieurs éléments ; dans la fenêtre édition de programme on remarque deux fonctions obligatoires **loop ()** et **setup ()**, la première est la partie principale qui constitue une boucle qui s'exécute sans fin, en d'autres termes c'est elle qui se charge du traitement du programme et ceci grâce à une suite d'instructions. [11]

Concernant la deuxième fonction, celle-ci a pour rôle d'initialiser les entrées et sorties. Généralement cette action se déroule une seule fois au début lors du démarrage [11]. Il existe aussi une troisième partie facultative qui se trouve le plus souvent en haut de l'éditeur et qui a pour but de déclarer les variables et le leurs types, on peut également ajouter des fonctions selon le besoin pour mieux organiser notre programme surtout si celui-ci est très long.



Fig.2. 7. Interface générale de l'Arduino.

Néanmoins, il existe plusieurs autres langages si on n'est pas satisfait par l'IDE standard, voici quelques logiciels alternatifs :

- **Visual Studio Code**

Conçu par Microsoft, c'est un outil entièrement gratuit qui existe en plusieurs versions pour Linux et OSX. L'avantage de cet outil est qu'il possède plusieurs nouvelles extensions qui ajoutent de nouvelles fonctionnalités et donc le rendent plus facile à utiliser, parmi ses extensions on retrouve la **PlatformIO** qui peut être téléchargée directement dans Visual Studio Code, qui permet la programmation de cartes Arduino et Raspberry. [16]

- **KTechLab**

Il s'agit d'un logiciel pour les microcontrôleurs, il peut simuler ce dernier et son circuit, cette interface nous permet aussi de visualiser la tension et le courant qui circulent dans le circuit.

C'est une plateforme open source qui fonctionne sur Windows et Linux. [17]

- **Codebender**

C'est un IDE cloud c'est-à-dire il fonctionne en ligne, ce qui est un avantage vu qu'on peut l'utiliser et y accéder partout sans devoir l'installer ce qui nous permet d'avoir toute les fonctionnalités qu'offre le langage Arduino.

C'est une plateforme polyvalente dans la mesure où il prend en charge plusieurs cartes téléchargeables. Tout comme les autres logiciels alternatifs celui-ci est aussi open source. [18]

- **Atmel Studio 7**

C'est une plateforme intégrée professionnelle, qui permet d'écrire des programmes, les déboguer. Elle prend en charge les microcontrôleurs SAM et AVR, et ne se limite pas qu'à l'Arduino mais également au développement d'applications. [19]

Exemple de clignotement d'une LED

On réalise un simple exemple de clignotement d'une LED à partir d'une carte UNO programmée avec le langage Arduino, afin de voir les différentes étapes pour mieux comprendre l'architecture de la carte lors de la réalisation d'un projet :

Chapitre 2 : Généralités sur l'Arduino.

La première étape consiste à réaliser un montage électrique, nous aurons besoin d'une carte UNO ainsi qu'une tablette d'essai, une LED et des fils afin de connecter les composants à la carte.

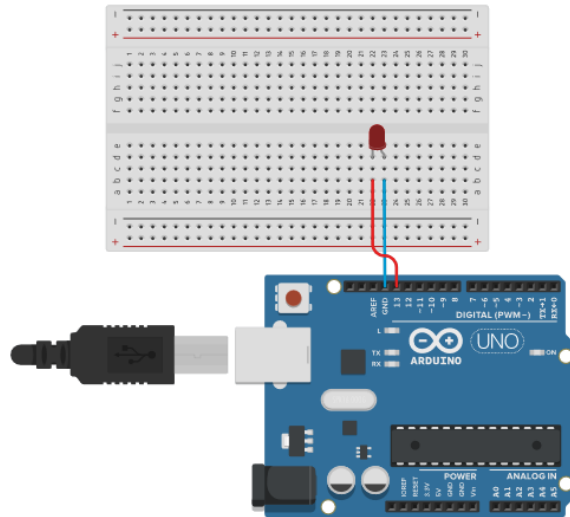


Fig.2. 8. Schéma électrique de la LED.

La LED est branchée à la plaque d'essai son anode représente la sortie qui est reliée à la broche 13, et la cathode est relié à la masse, comme on le voit dans la figure (Fig.2.8).

L'étape suivante est celle de l'écriture du programme dans L'IDE, le langage est relativement simple avec coloration syntaxique qui aide à mieux visualiser les différentes instructions.

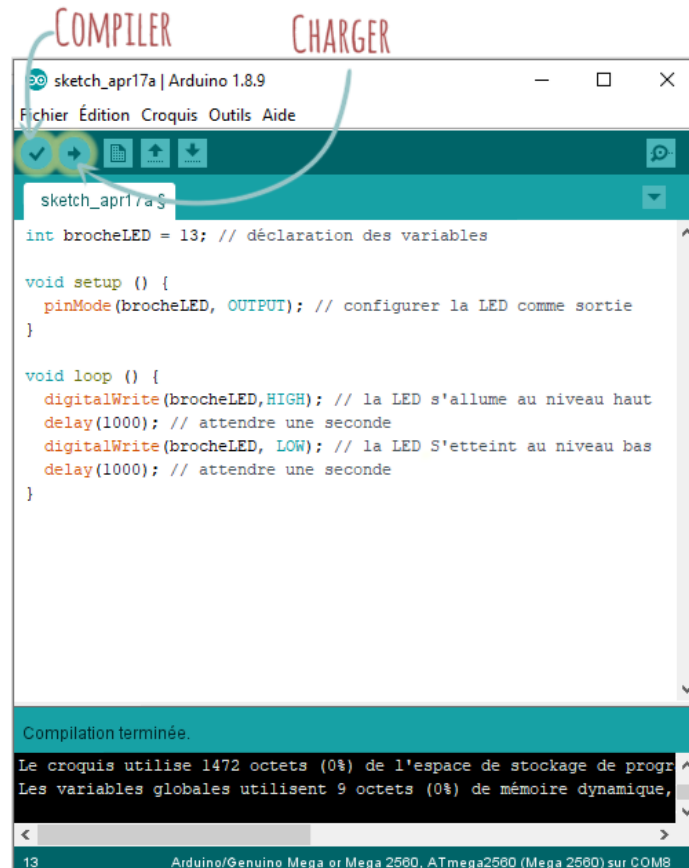


Fig.2. 9. Programme Arduino.

Une fois le programme écrit on vérifie s'il n'y a pas d'erreur en le compilant.

Enfin si le code est correct, on choisit le port de communication adéquate qui se trouve dans « outils », on branche la carte à l'ordinateur via câble USB et on charge le programme, comme on le voit dans la figure au-dessus (Fig.2.9).

2.5 Conclusion

Au terme de ce chapitre, nous avons pu voir les généralités sur l'Arduino d'un point de vu hardware où nous nous sommes intéressés à l'architecture générale des cartes Arduino ainsi qu'aux caractéristiques de plusieurs types de cartes que propose la compagnie. Nous avons cité quelques modules et shields pour comprendre leurs utilités et la différence entre les deux, accompagnés d'un exemple afin de voir les fonctionnalités qu'ils nous offrent.

Ensuite, nous avons abordé la partie software avec le langage Arduino ainsi que les autres IDE et plateformes équivalentes pour la programmation des cartes, avec un exemple pour voir les étapes à suivre lors de la conception du programme.

Chapitre 3 : Conception et réalisation

3.1 Introduction

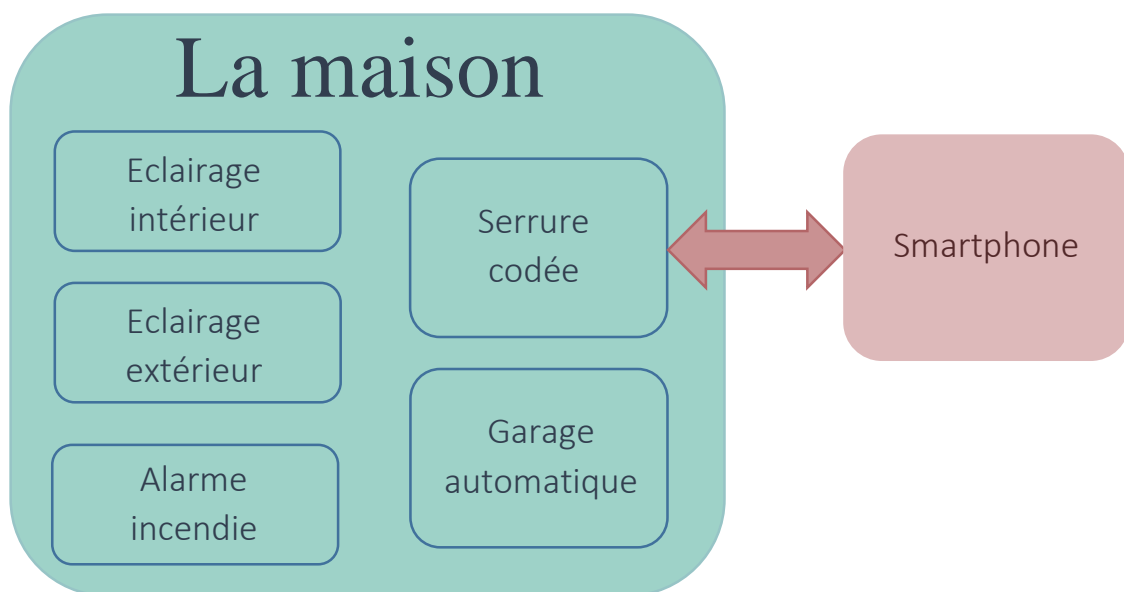
Dans ce chapitre nous exposerons le processus de réalisation de notre box domotique est dotée d'un Bluetooth.

Dans un premier lieu, nous allons présenter le matériel nécessaire à la conception de notre projet ; nous expliquerons les différents capteurs et composants qui seront utilisés ainsi que leur mode de fonctionnement, ensuite nous allons procéder au branchement qui sera placé dans la maquette.

Enfin nous allons réaliser une application Android sur smartphone pour déverrouiller la serrure.

3.2 Fonctions intégrées dans le système

Dans notre projet nous allons nous lancer dans la réalisation des cinq scénarios suivants : Serrure codée, éclairage intérieur, éclairage extérieur, garage automatique et enfin alarme incendie, Dans ce qui suit le fonctionnement et organigramme de chaque application.



3.2.1 Serrure codée

La serrure est l'élément clé qui procure le sentiment de sécurité dans une maison, avec l'évolution et l'automatisation de plusieurs objets celle-ci est devenue plus élaborée, avec un code à taper ou bien reconnaissance faciale ou encore empreinte digitale, tous les moyens sont bons pour garantir la meilleure sécurité et protection des biens.

3.2.1.1 Fonctionnement de la serrure

Afin de déverrouiller la serrure nous avons le choix entre saisir le code par le clavier ou à l'aide d'une application sur smartphone.

En utilisant le clavier nous avons droit à trois essais, si durant l'une de ces tentatives le code est correcte, la serrure se déverrouille et la porte s'ouvre après avoir validé le mot de passe en appuyant sur la touche (A), si on introduit un mot de passe faux un message « code incorrecte » s'affichera. Au-delà du troisième essai, l'écran affichera le message suivant « serrure bloquée », pour débloquer cette dernière il faudra saisir le code sur téléphone. Le changement de code se fait via l'application.

Dans notre réalisation la porte possède que la serrure codée étant donné que la maquette est de taille réduite, mais dans une véritable habitation la porte sera également dotée d'une serrure classique qui se déverrouille avec une clé en cas de panne.

3.2.1.2 Organigramme de la serrure codée

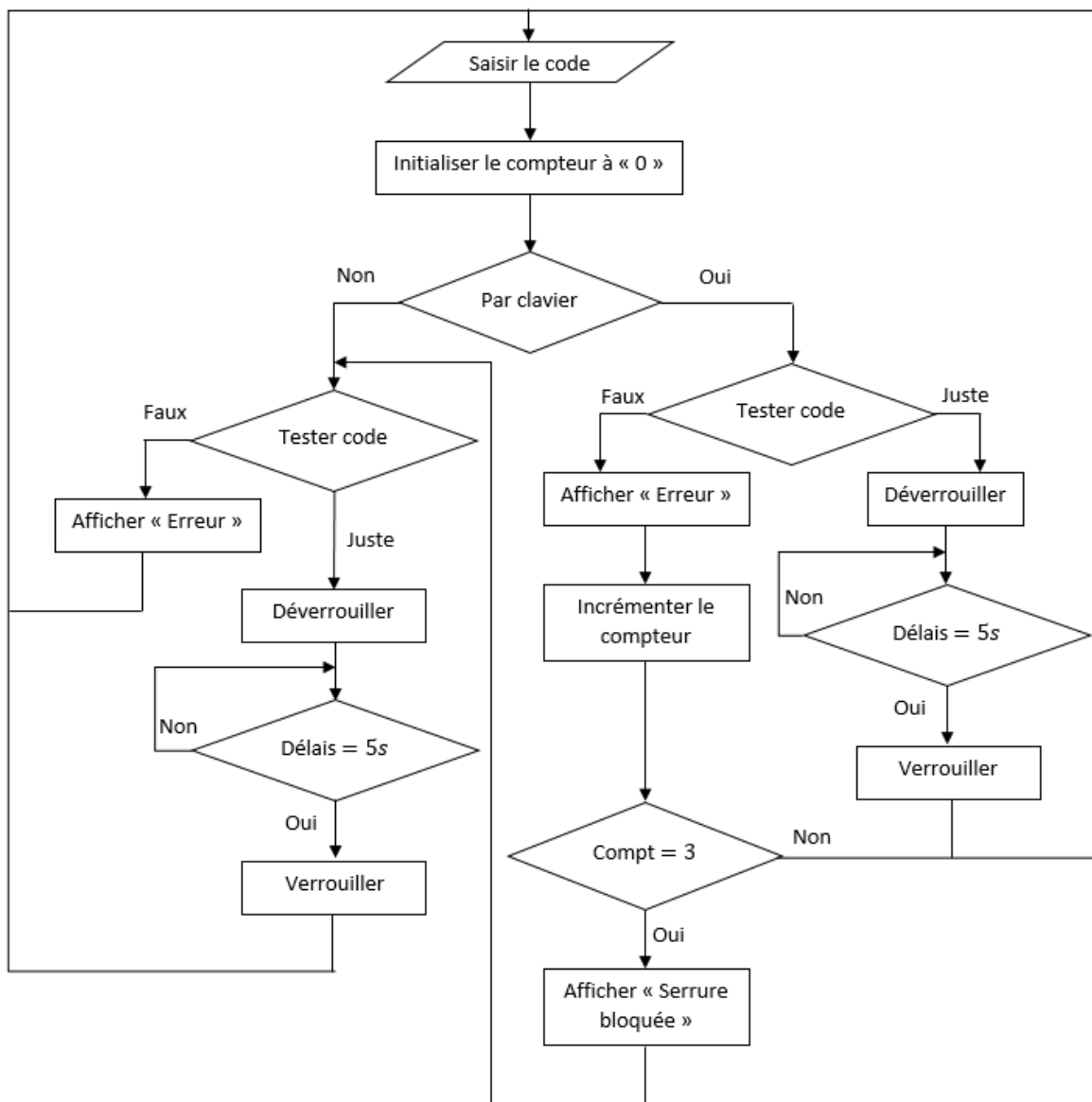


Fig.3. 1. Organigramme de la serrure codée.

3.2.2 Garage automatique

Les maisons dotées de garages pour y garder sa voiture et de ce fait la protéger des intempéries et de voleurs, offrent un sentiment de sécurité aux propriétaires et si en plus la porte fonctionne de manière automatique on ajoute à la sécurité le critère du confort.

3.2.2.1 Fonctionnement du garage automatique

Le garage est commandé à distance à l'aide d'une télécommande pour l'ouvrir ou le fermer de l'extérieur. En plus de la télécommande un bouton permet de fermer et ouvrir le garage depuis l'intérieur.

3.2.2.2 Organigramme du garage automatique

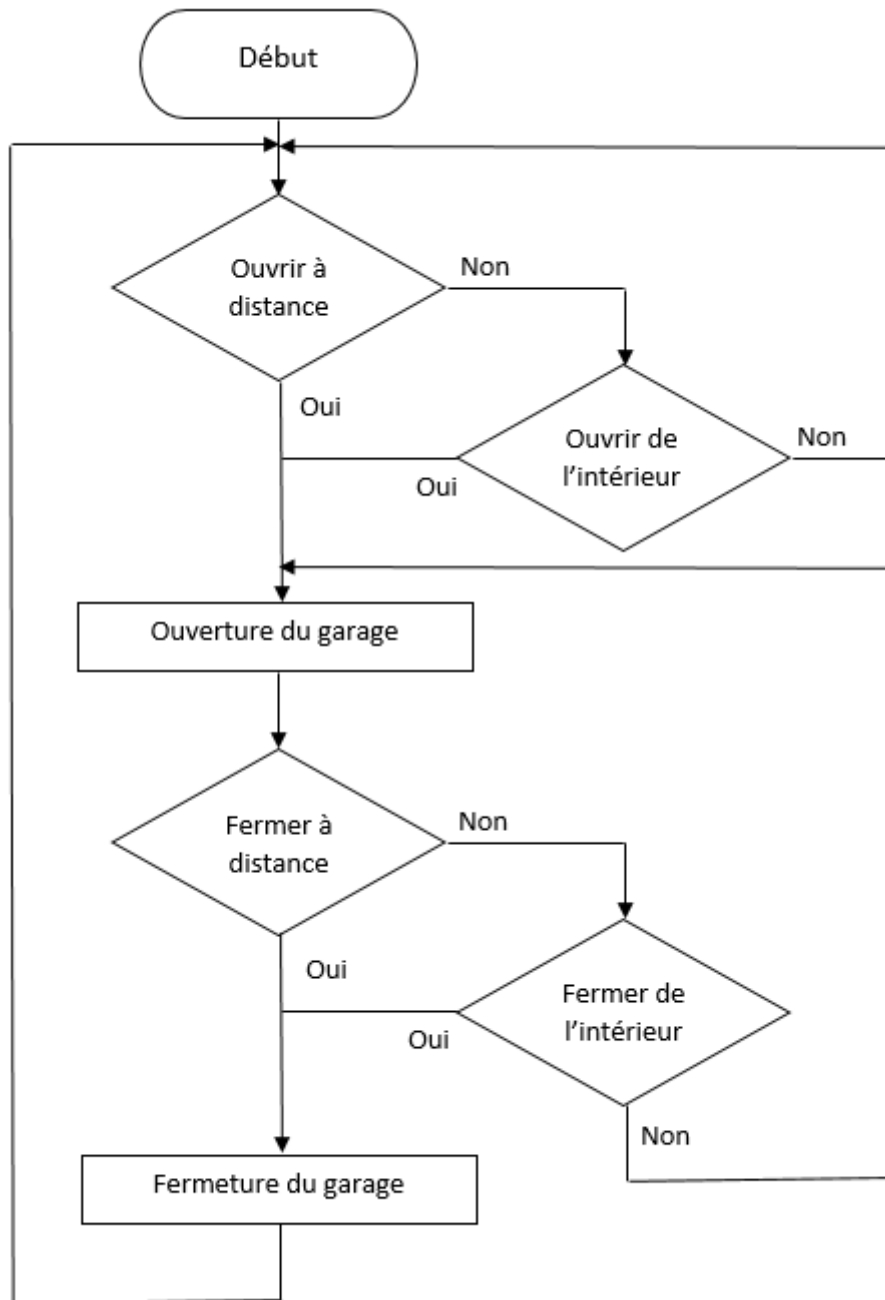


Fig.3. 2. Organigramme de garage.

3.2.3 Eclairage intérieur

Ce système permet l'économie d'énergie et la réduction d'une consommation inutile en électricité. Quel que soit le capteur utilisé : détecteur de mouvement ou détecteurs de présence, le principe est le même ; à savoir allumer la lumière d'une manière automatique.

3.2.3.1 Fonctionnement de l'éclairage intérieur

La lampe s'allume lorsqu'un mouvement est détecté par le capteur et si la pièce n'est pas suffisamment éclairée ou bien plongée dans le noir.

La lampe reste allumée pendant 30s, si entre temps un autre mouvement est détecté, le compteur se remet à zéro.

On peut également éteindre la lumière grâce à un bouton poussoir désactivant ainsi le système ce bouton nous permet aussi d'éteindre la lumière. On doit appuyer une seconde fois sur le bouton poussoir pour réactiver le système.

3.2.3.2 Organigramme de l'éclairage intérieur

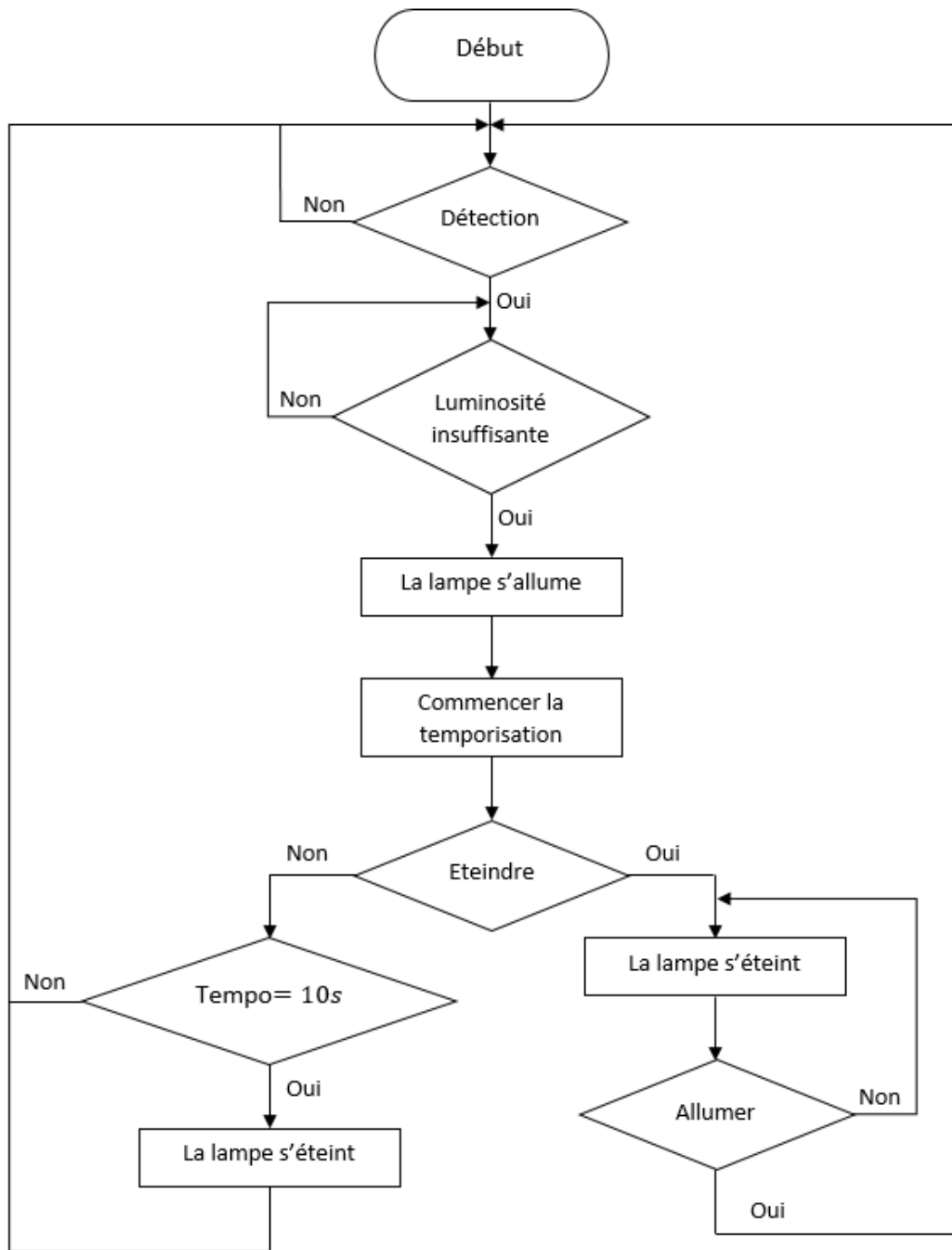


Fig.3. 3. Organigramme de l'éclairage intérieur.

3.2.4 Eclairage extérieur

Ces dernières années, l'éclairage extérieur nocturne a pris de l'importance en matière de confort et sécurité, en effet on peut l'utiliser dans un but fonctionnel c'est-à-dire faciliter les déplacements lorsqu'il fait nuit, éviter les chutes ou encore dissuader des voleurs de s'introduire dans la propriété.

3.2.4.1 Fonctionnement de l'éclairage extérieur

Dans notre projet nous allons utiliser six LED bleues, reliées à une photorésistance afin de détecter la lumière extérieure. Lorsque celle-ci est trop faible, les LED s'allument, à partir du crépuscule jusqu'au lever du jour.

Un bouton poussoir est ajouté pour activer ou désactiver le fonctionnement automatique de l'éclairage.

3.2.4.2 Organigramme de l'éclairage extérieur

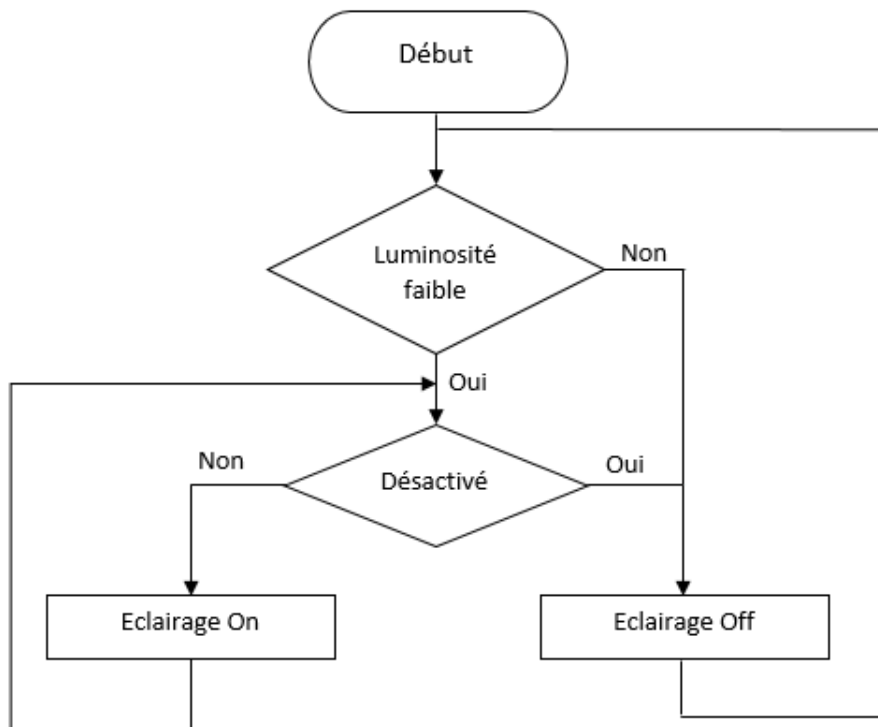


Fig.3. 4. Organigramme de l'éclairage extérieur.

3.2.5 Alarme incendie

L'un des éléments les plus importants dans une maison intelligente est le système d'alarme incendie, étant donné que ce dernier protège à la fois les habitants et la propriété, ce qui procure un sentiment de sécurité aux propriétaires. Ce type d'installation peut couvrir l'ensemble de la maison et donc offre une protection totale, comme il peut être placé que dans certains endroits spécifiques, par exemple dans la cuisine.

3.2.5.1 Fonctionnement de l'alarme

Le système d'alarme incendie a pour fonction de détecter la fumée qui est généralement causée par un feu ou un incendie ainsi que les fuites de gaz, et de donner l'alerte sous forme d'alarme qu'on peut arrêter manuellement grâce à un bouton poussoir. Si après 10s le

capteur continue de détecter alors l'alarme se déclenchera à nouveau, si le capteur ne détecte plus de gaz ou fumée même sans appuyer sur le bouton l'alarme s'arrête toute seule.

3.2.5.2 L'organigramme de l'alarme

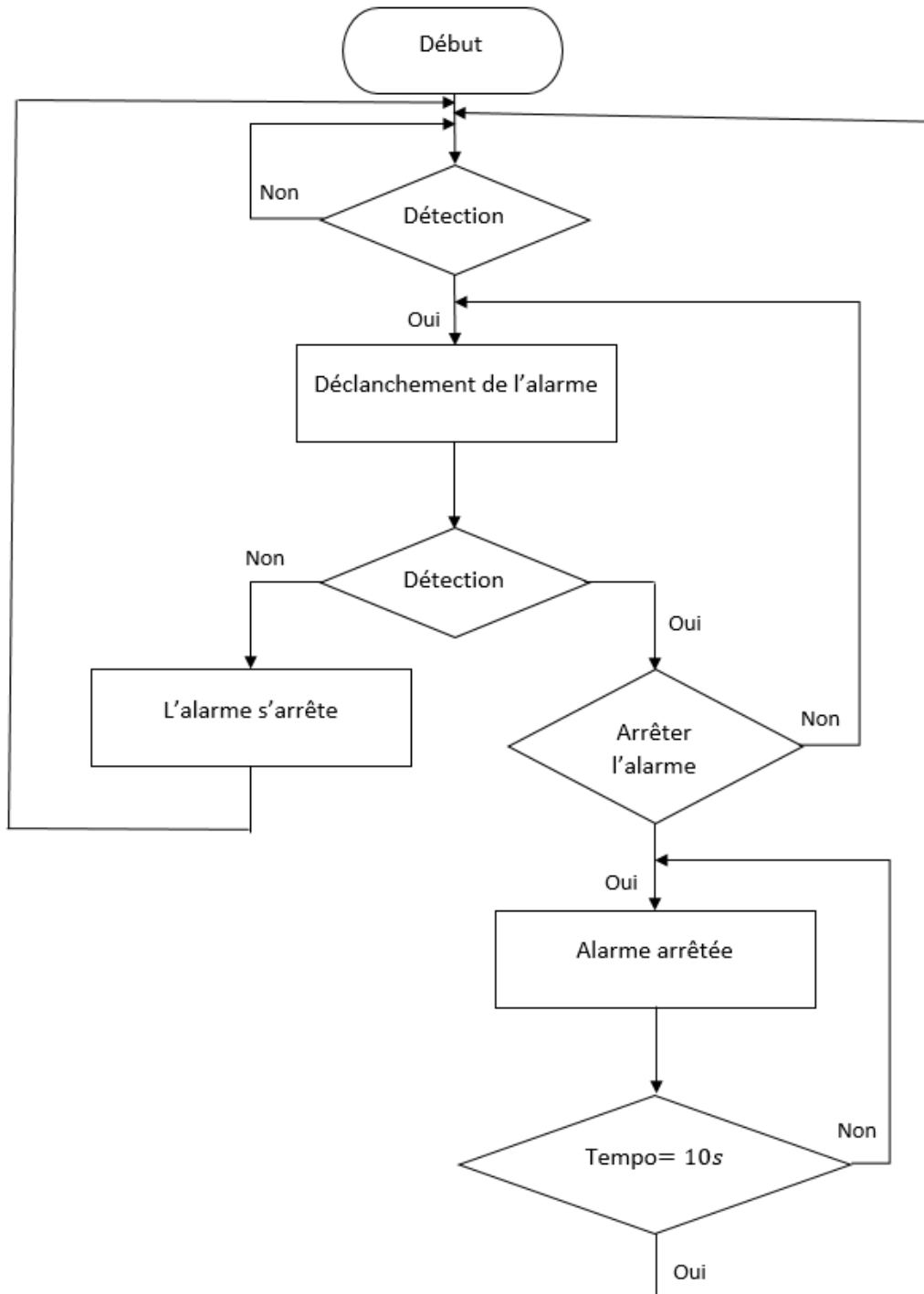


Fig.3. 5. Organigramme de l'alarme.

3.3 Matériel utilisé

Les cartes Arduino MEGA2560 et Arduino UNO seront présentes lors des réalisations qui seront effectuées ainsi qu'une plaque d'essai.

Afin de protéger les composants nous utiliserons aussi des résistances de 180Ω .

3.3.1 Serrure codée

Pour cette réalisation, nous allons utiliser la carte Arduino UNO avec certains composants qui sont les suivants :

- **L'écran LCD**

Pour notre cas nous avons utilisé un afficheur LCD alphanumérique 16x2 avec rétro-éclairage pour qu'il soit lisible de jour comme de nuit.

Le module dispose de 16 broches que nous pouvons connecter à 8 bits en utilisant toutes les broches ou bien dans notre cas une connexion parallèle à 4 bits ou on utilise 12 broches ou seul les pins **DB4** à **DB7** sont utilisé comme sortie, les broches **E** et **RS** pour la commande, nous avons également relié à l'écran un potentiomètre pour régler le contraste, et étant donné que nous faisons uniquement l'écriture nous avons relié la broche **W/R** à la masse. [20]

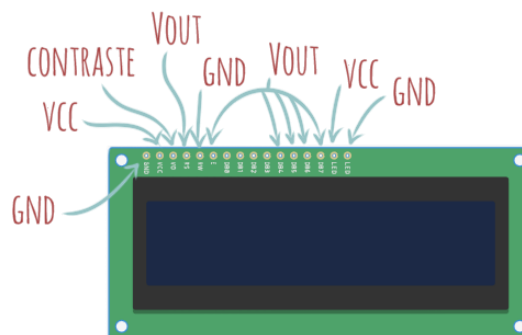


Fig.3. 6. Ecran LCD 16x2.

- **Bluetooth**

Le Bluetooth est un outil de communication sans fil bidirectionnelle, avec un fonctionnement simple, celui-ci relie plusieurs périphériques à un hôte ou plus.

Le module que nous avons choisi pour notre réalisation est le modèle HC-05, il utilise un mode de communication maître/esclave. Avec un débit limité et une portée comprise entre 15 à 20m. [2]

Chapitre 3 : Conception et Réalisation.

Il est doté de six broches : celles de la masse et l'alimentation, ainsi que deux autres broches **TX** et **RX** pour respectivement établir la transmission et la réception, pour ce qui est des deux broches externes nous ne les avons pas utilisées : la broche **state** retourne 1 lorsque le module est connecté, tant dit que la broche **EN** ou bien **KEY** est connectée lorsqu'on veut configurer le module, et doit être alimenter à 3V non-associé, durant la configuration la LED du Bluetooth clignote.

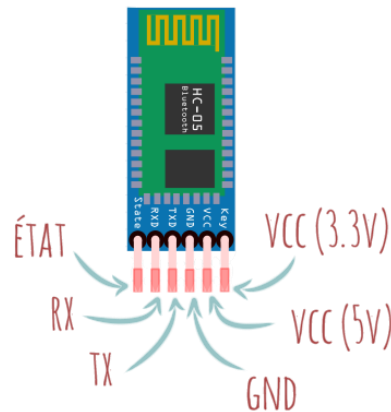


Fig.3. 7. Module Bluetooth.

- **Servomoteur**

Il s'agit d'un actionneur capable de maintenir une position statique, avec un débattement de 180 degrés.

Il est constitué d'un potentiomètre, d'un moteur électrique à courant continu, d'un réducteur pour augmenter le couple et de palonnier qui se trouve en haut, c'est ce dernier qui effectue les rotations.

Doté de trois fils marron, rouge et jaune qui représentent respectivement la masse, l'alimentation et la sortie PWM.

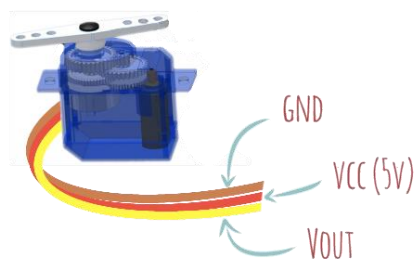


Fig.3. 8. Le servomoteur.

3.3.2 Garage automatique

En plus de la carte Arduino MEGA, le matériel utilisé pour réaliser le montage du garage automatique est le suivant

- **La télécommande infrarouge**

Souvent utilisée avec les télévisions ainsi que pour les portes de garages, entre autre tout appareil commandable par infrarouge.

La télécommande est dotée d'une diode électroluminescente, qui émet des rayons infrarouges qui sont interceptés par le récepteur qui est une photodiode capable de transformer un signal lumineux en un signal électrique. [20]

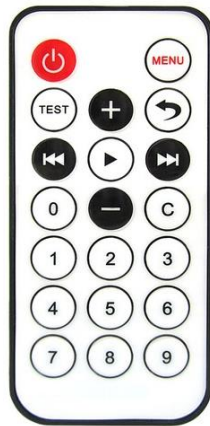


Fig.3. 9. Télécommande.

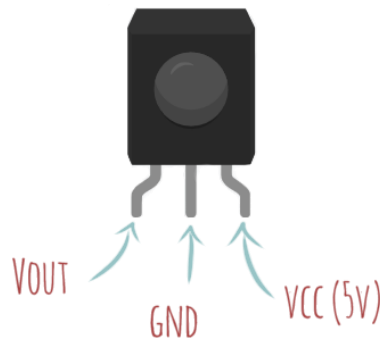


Fig.3. 10. Capteur infrarouge.

- **Bouton poussoir**

C'est un interrupteur avec deux positions :

La première est lorsque le bouton est relâché c'est-à-dire le contacteur est relié à la masse et donc ne laisse pas passer le courant alors le circuit est déconnecté ou bien ouvert. La seconde position est lorsque le contacteur est relié à l'alimentation et permet au courant de passer, et cela en appuyant sur le bouton, le circuit est donc fermé.

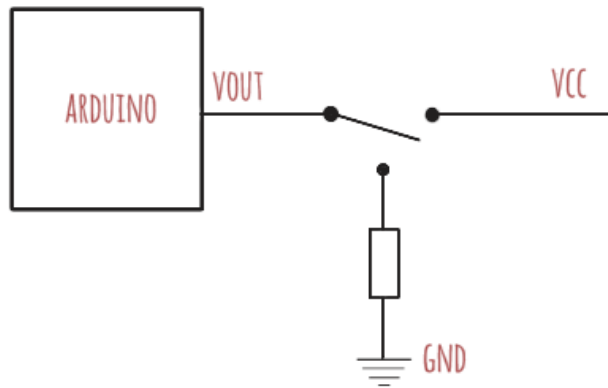


Fig.3. 11. Schéma du bouton poussoir.

3.3.3 Eclairage intérieur

Afin de réaliser ce montage, nous avons utilisé en plus de la carte MEGA, les composants et capteur suivant

- **Capteur PIR**

Il s'agit d'un capteur de mouvement infrarouge passif (passive infrared), le terme « passif » dans ce cas signifie que le capteur n'utilise pas d'énergie lors de la détection, mais reçoit uniquement celle dégagée sous forme d'énergie thermique par un corps humain ou animal lorsque ces derniers se trouvent dans le champ de détection.

Le module se compose d'une lentille sous forme de dôme qui focalise les signaux infrarouges sur le capteur. Il compte aussi trois broches (la masse, l'alimentation 5V et la sortie). Il possède également deux potentiomètres celui de droite est pour régler la sensibilité du capteur (la distance), celui de gauche pour ajuster le temps pendant lequel l'objet est détecté. [14]

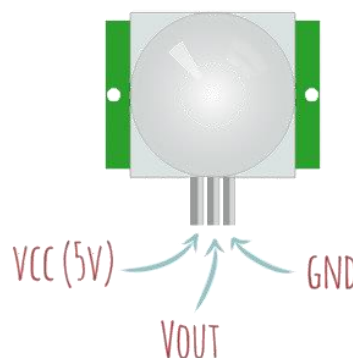


Fig.3. 12. Capteur de mouvement.

- **Le relais**

Il s'agit d'un interrupteur mécanique, qui possède un électroaimant qui tire une plaque métallique afin d'activer le circuit à contrôler. L'avantage d'un relais est qu'il assure l'isolation complète des deux circuits. [21]

Nous avons choisi un relais a un seul canal qui utilise une tension Arduino de 5V pour commander un circuit d'une tension de 220V avec un courant de 10A.

Doté de trois broches qu'on doit connecter à l'Arduino : la masse, l'alimentation et la sortie, il possède également trois ports pour le branchement de haute tension : le **NC** (normalement fermé), **NO** (normalement ouvert) et enfin le **C** (commun).

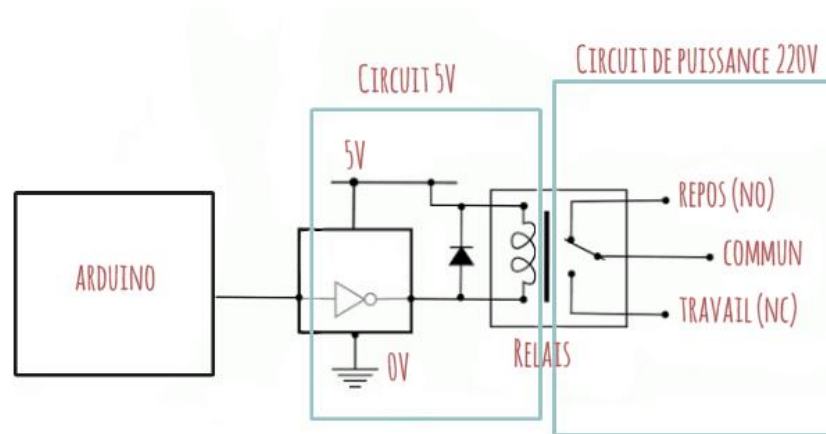


Fig.3. 13. Relais 5V

3.3.4 Eclairage extérieur

Le montage suivant est réalisé à partir d'une carte Arduino MEGA et du matériel suivant

- **La photorésistance**

Il s'agit d'une résistance qui varie selon l'intensité lumineuse, en d'autres termes plus la lumière est élevée plus la résistance diminue et vice versa.

La photorésistance est un composant électronique transducteur par conséquent peu précis, qui est préférable de l'utiliser en tant que détecteur plutôt qu'élément de mesure.

Il est constitué de matériau photoconducteur et d'électrodes.

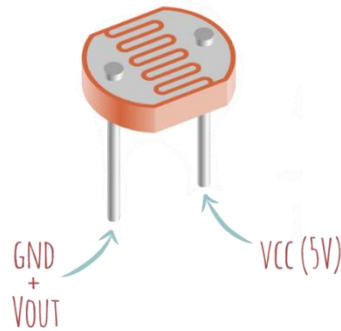


Fig.3. 14. Photorésistance.

- **Les LED**

Les LED sont des diodes lumineuses qui laissent passer le courant dans un seul sens lors de la polarisation directe ; de l'anode vers la cathode. Bloque le courant lorsqu'il vient du sens inverse.

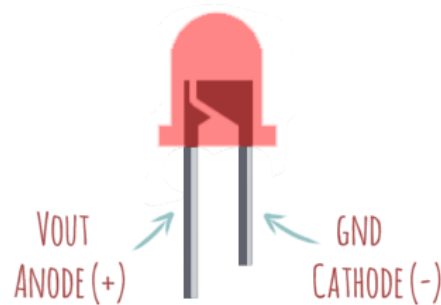


Fig.3. 15. LED.

3.3.5 Alarme incendie

Pour réaliser le montage d'un système d'alarme incendie, nous aurons besoin d'une carte Arduino MEGA et du matériel suivant

- **Un capteur de gaz/fumée :**

Dans le cadre de notre projet nous avons opté pour un détecteur de gaz et fumée pour une protection plus optimale des biens et des habitants.

Le capteur en question est le modèle **MQ-2**, ce dernier peut détecter six types de gaz : le butane, propane, méthane, alcool, l'hydrogène et enfin le GPL (gaz de pétrole liquéfié), en plus de la fumée qui généralement est une conséquence due au déclenchement d'un incendie.

La tension de sortie du capteur change selon la valeur du gaz ou de fumée détectée et cela de façon proportionnelle. Celui-ci est aussi doté d'un potentiomètre qui permet d'ajuster

Chapitre 3 : Conception et Réalisation.

la valeur du seuil de la sortie numérique et c'est ce seuil en question qui définit la valeur à laquelle les gaz/ fumée sont détectés. [22]

Il compte deux LED la première pour indiquer la mise sous tension en rouge et la seconde en vert lorsqu'il capte de la fumée ou bien du gaz, il possède également quatre broches : la masse (**GND**), l'alimentation (**VCC à 5V**) et enfin deux sorties ; analogique ou bien numérique (**DOUT** et **AOUT**).

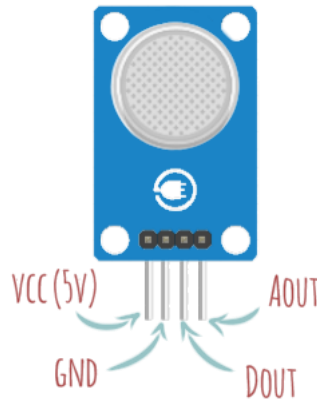


Fig.3. 16. Capteur de gaz et fumée.

- **Le Buzzer**

Il existe deux types de buzzer : le buzzer passif qui agit plus comme un mini haut-parleur et le buzzer actif, qui est un dispositif sonore, qui fonctionne principalement en transformant l'énergie électrique en vibration en d'autres termes : un son qui change de tonalité selon la fréquence qu'on lui applique et c'est ce dernier que nous utilisons pour notre projet.

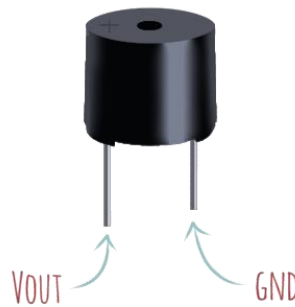


Fig.3. 17. Buzzer.

3.4 Schémas synoptique

Les schémas suivants représentent respectivement la représentation synoptique de la structure générale de la carte UNO (Fig.3.18) ensuite celle de la carte MEGA (Fig.3.19).



Fig.3. 18. Schéma synoptique de la carte UNO.

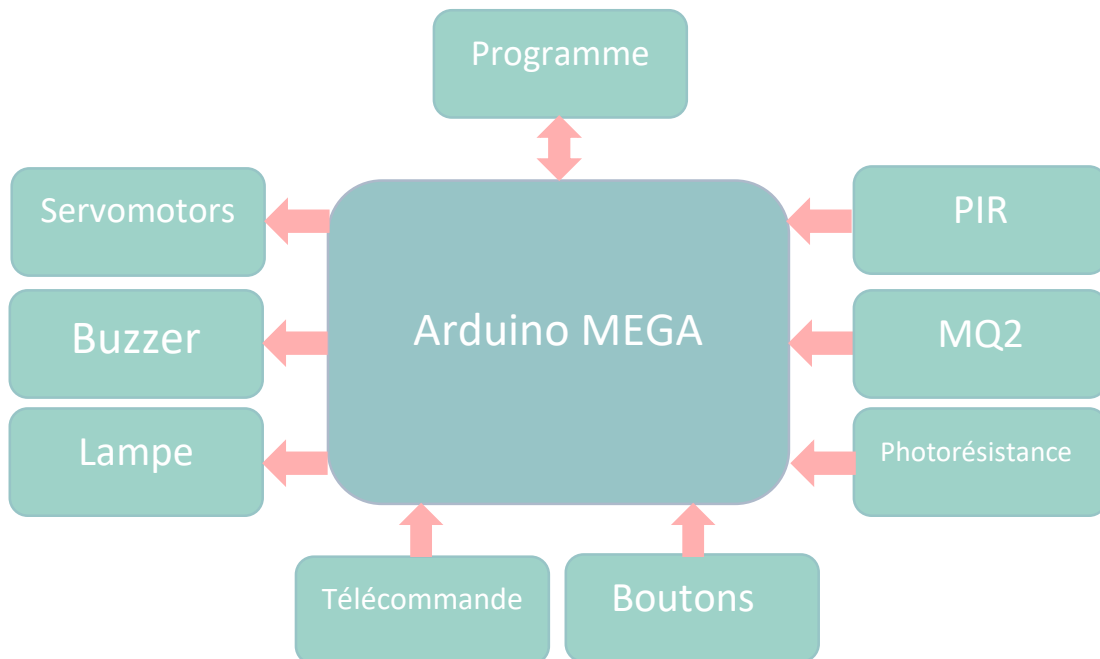


Fig.3. 19. Schéma synoptique de la carte MEGA.

3.5 Branchement des applications

Dans cette partie nous allons décrire les branchements réalisés pour les différentes applications.

3.5.1 Serrure codée

La figure suivante représente le schéma du circuit réalisé pour la serrure codée.

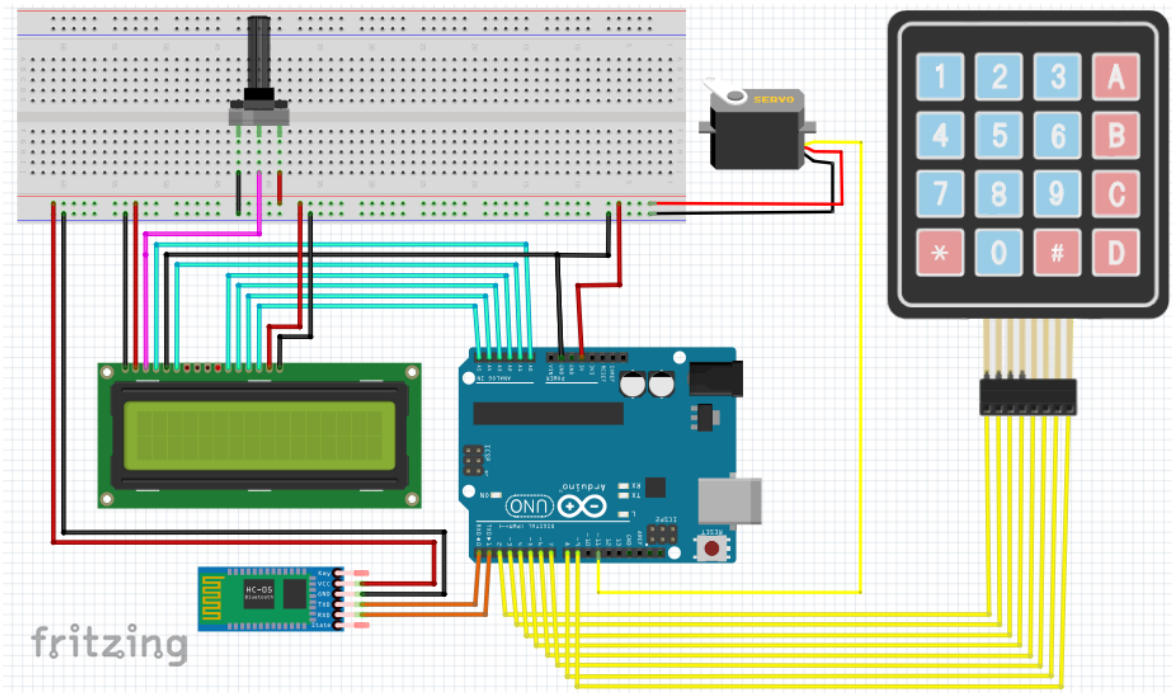


Fig.3. 20. Schéma électrique de la serrure.

Le tableau 3.1 représente l'affectation des broches de différents composants utilisés pour la serrure.

Tableau 3. 1. Branchement de la serrure.

Composants	LCD	Clavier	Bluetooth	Servomoteur
Branchement	<ul style="list-style-type: none"> - VSS, W/R et K vers la masse. - VDD et A vers le VCC. - VO vers le potentiomètre. - DB4 à DB7 sont reliés de A2 à A5. - E et RS à A0 et A1. 	Les broches sont connectées aux pins allant de 2 à 9.	TX vers RX. RX vers TX.	La broche PULSE vers la pin 11.

3.5.2 Garage automatique

Le schéma ci-dessus représente les branchements du garage automatique.

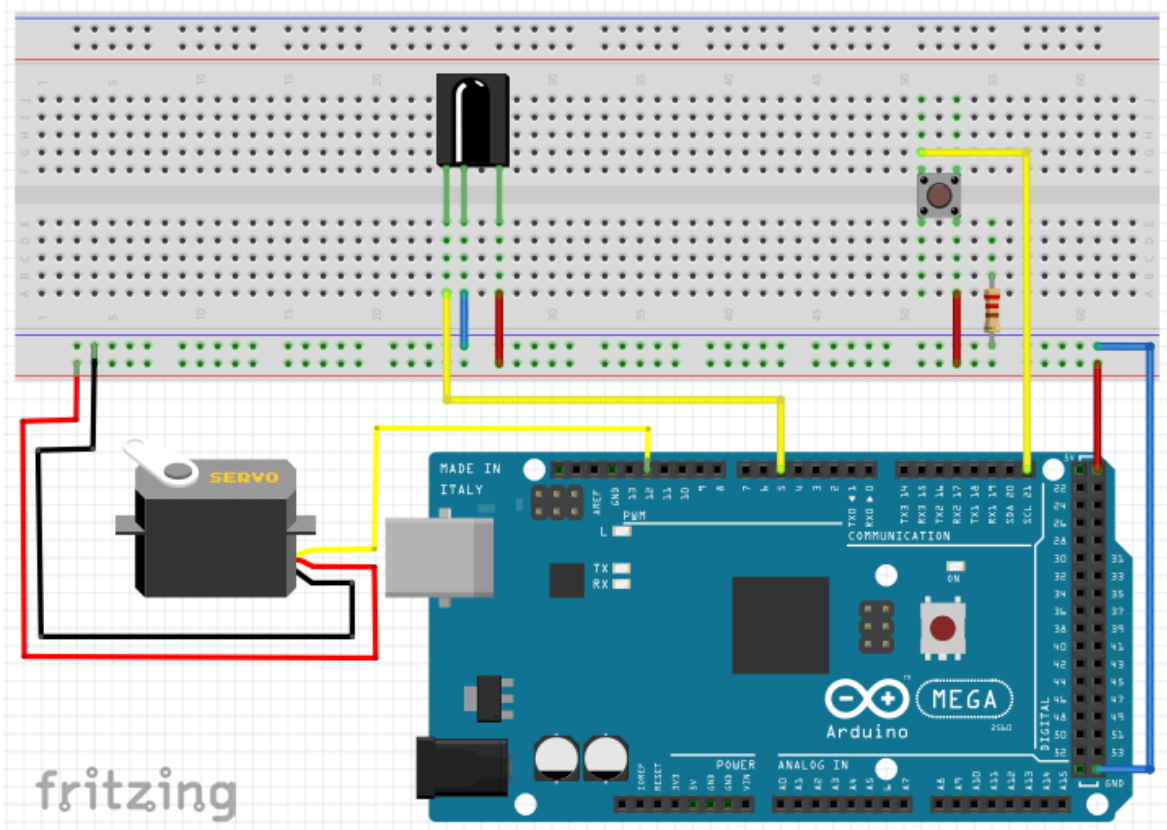


Fig.3. 21. Schéma électrique du garage automatique.

L'affectation des pins pour le garage est citée dans le tableau 3.2 :

Tableau 3. 2. Branchement du garage automatique.

Composants	Récepteur infrarouge	Bouton d'arrêt	Servomoteur
Branchement	La sortie vers la pin 5	Connecté à la pin 21	connecté à la pin 12

3.5.3 Eclairage intérieur

Le schéma de l'éclairage intérieur est présenté par la figure suivante

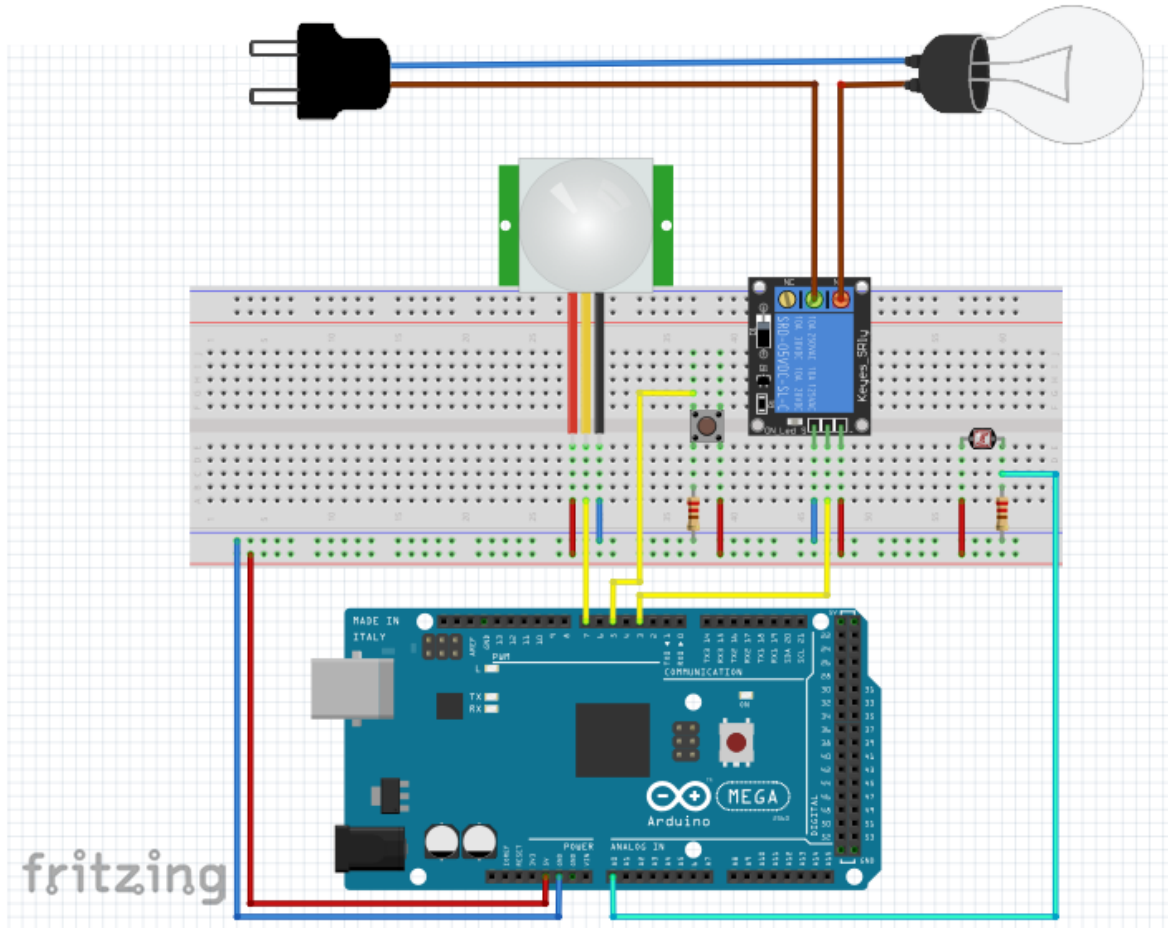


Fig.3. 22. Schéma électrique de l'éclairage intérieur.

L'affectation des pins pour les différents composants utilisés pour l'éclairage figure dans le tableau 3.3.

Tableau 3. 3. Branchement de l'éclairage intérieur.

Composants	Relais	Capteur PIR	Bouton	Photorésistance
Branchement	La sortie est reliée à la pin 3.	La sortie est reliée à la pin 7.	Connecté à la pin 5.	Connectée à la pin analogique A0.

3.5.4 Eclairage extérieur

La figure suivante représente le schéma de l'éclairage extérieur.

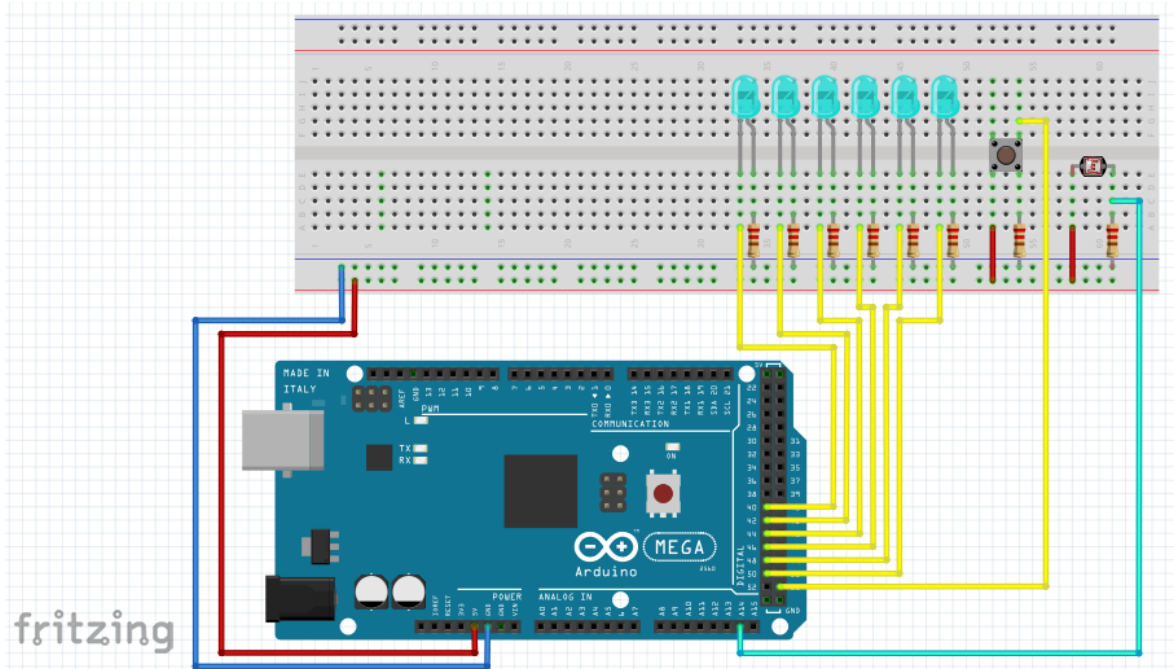


Fig.3. 23. Schéma électrique de l'éclairage extérieur.

Dans le tableau 3.4 on retrouve les pins affectés aux différentes broches.

Tableau 3. 4. Branchement de l'éclairage extérieur.

Composants	LED	Bouton	Photorésistance
Branchement	Sont branchées aux pins 40, 42, 44, 46, 48 et 50.	Connecté à la pin 53.	Connecté à la broche analogique A14.

3.5.5 Alarme incendie

Pour l'alarme incendie le schéma se présente comme suit.

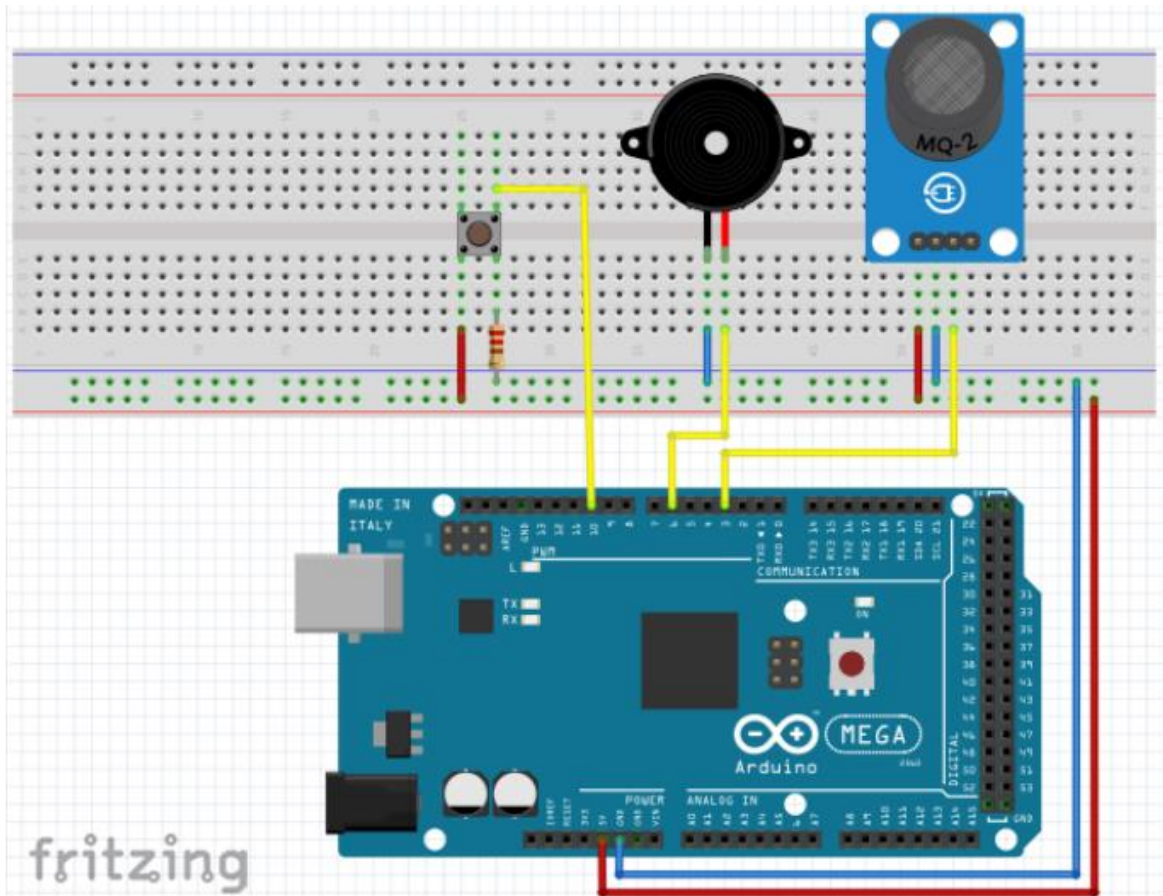


Fig.3. 24. Schéma électrique de l'alarme incendie.

Les différentes broches sont affectées comme suit dans le tableau 3.5.

Tableau 3. 5. Branchement de l'alarme incendie.

Composants	Capteur de gaz	Buzzer	Bouton
Branchement	La sortie est branchée à la pin 3.	est connecté à la pin 6.	Est relié à la pin 10.

3.6 Réalisation de la maquette

Dans la figure (Fig.3.25) on a une vue d'ensemble sur la façade extérieure et sur l'intérieur de la maquette que nous avons conçu et dans la figure (Fig.3.26) on peut voir la maquette alimentée.



Fig.3. 25. La maquette vue d'ensemble.



Fig.3. 26. La maquette alimentée.

3.7 Application Android (App Inventor)

Toujours dans l'optique de garantir le confort et la sécurité des habitants, la conception d'une application sur smartphone ou tablette est préférable.

Chapitre 3 : Conception et Réalisation.

Nous avons choisi App Inventor qui est un logiciel visuel open source en ligne, pour sa facilité d'utilisation. Il s'agit d'un logiciel qui fonctionne avec des blocs à imbriquer comme un puzzle, ces derniers sont préprogrammés.

Le logiciel possède deux interfaces. La première est celle du Design où on construit l'interface graphique de l'application, celle-ci se divise en quatre parties comme montré dans la figure (Fig.3.27) :

Le « view » : représente un téléphone et donc dessus nous pouvons voir à quoi ressemblera l'application, il regroupe les différents composants et écrans utilisés.

La palette : qui contient les composants pré-codés et préinstallés, ces derniers peuvent être visibles comme invisibles sur l'écran.

Les composants : dans cette partie on retrouve les composants sélectionnés pour notre application dans la partie précédente.

Propriétés : cette partie nous permet de modifier et personnaliser les composants utilisés.

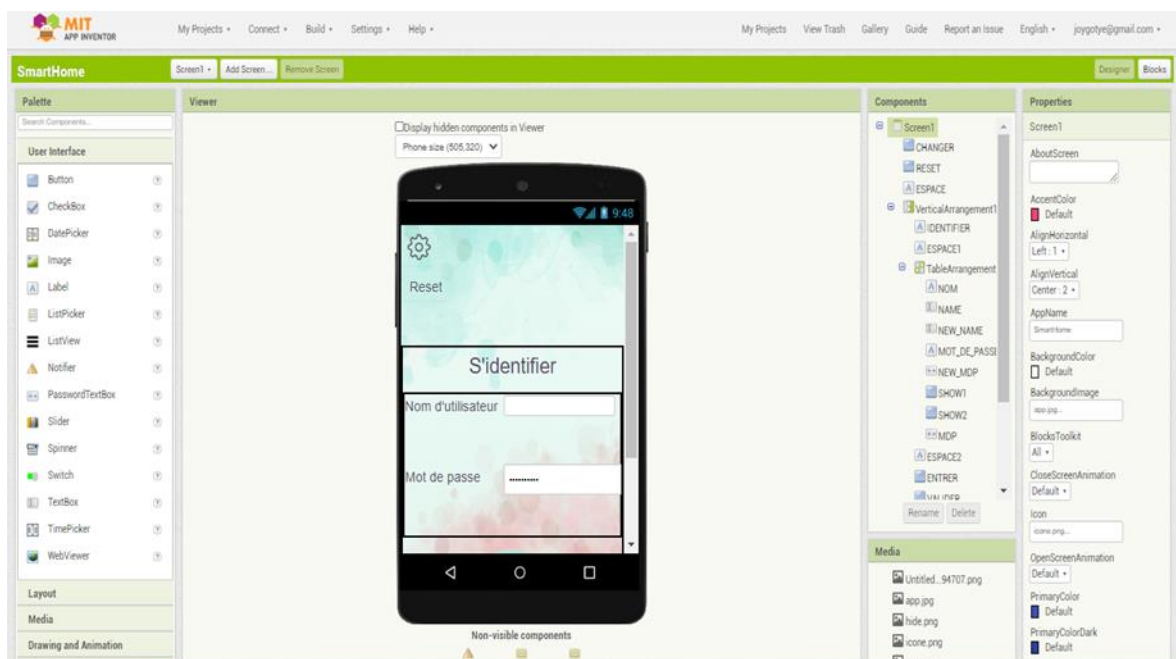


Fig.3. 27. Interface de design d'App inventor.

Chapitre 3 : Conception et Réalisation.

La seconde interface (Fig.3.28) est celle de la programmation ou nous allons pouvoir programmer les différents éléments choisis. Et cela grâce des blocs codé au préalables ou bien des blocs associés aux composants que nous avons choisi.

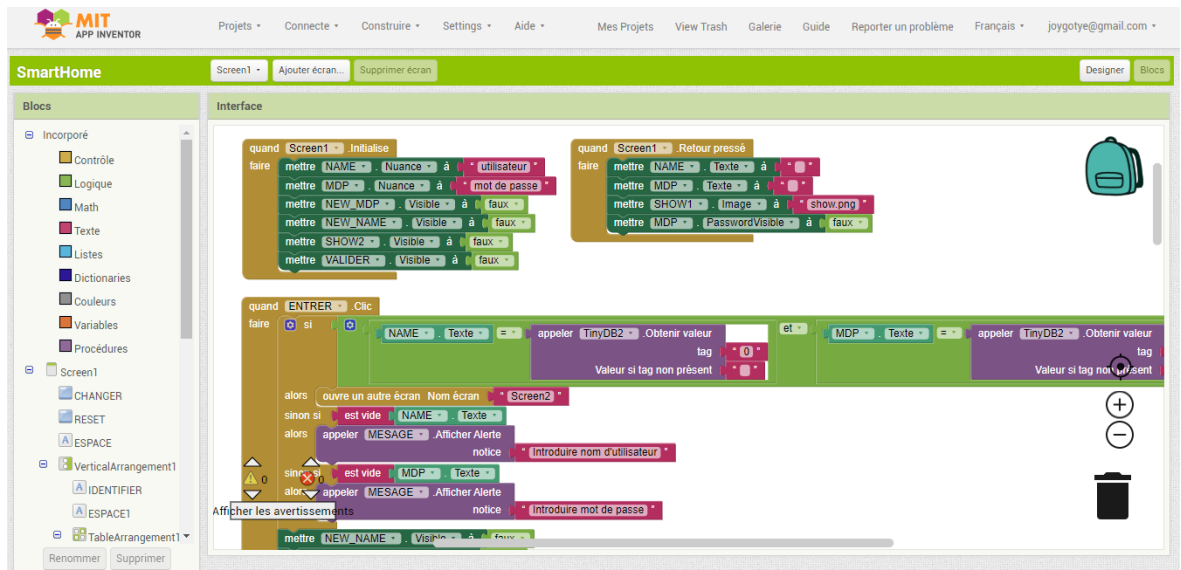


Fig.3. 28. Interface de programmation.

L'application que nous avons conçue se connecte à un module Bluetooth pour déverrouiller la serrure, avec identification c'est-à-dire que dès qu'on lance l'application dans la première fenêtre qui apparait on remarque deux champs de textes un pour le nom de l'utilisateur, l'autre pour le mot de passe avec possibilité de les changer comme le montre la figure suivante (Fig.3.29).

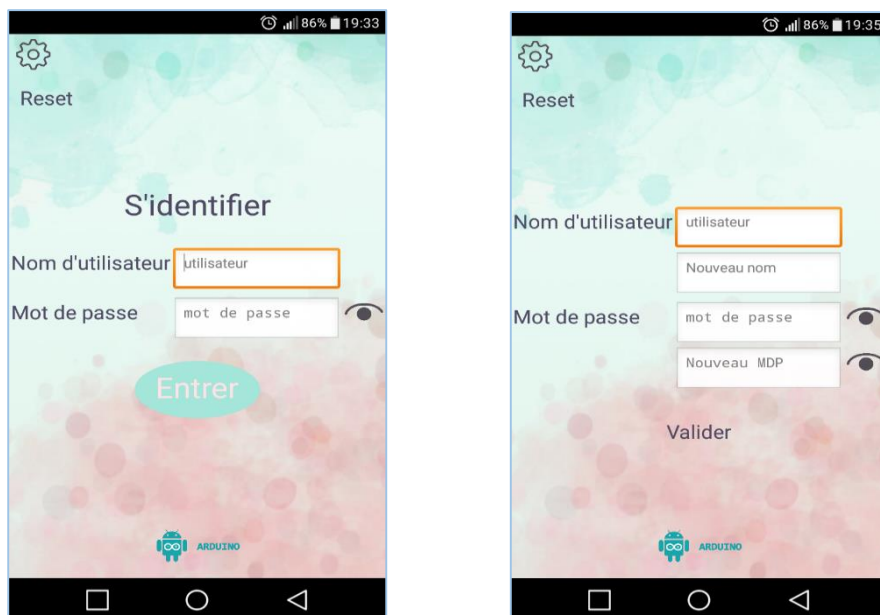


Fig.3. 29. Fenêtre d'identification.

Chapitre 3 : Conception et Réalisation.

Une fois l'identification terminée et après avoir appuyé sur le bouton « entrer », on se retrouve dans la deuxième fenêtre où un message apparaît qui nous demande d'activer le Bluetooth ensuite de se connecter à ce dernier afin de déverrouiller la serrure en introduisant le code. Dans cette même fenêtre il est possible de changer le code si nécessaire comme on le voit dans la figure (Fig.3.30).

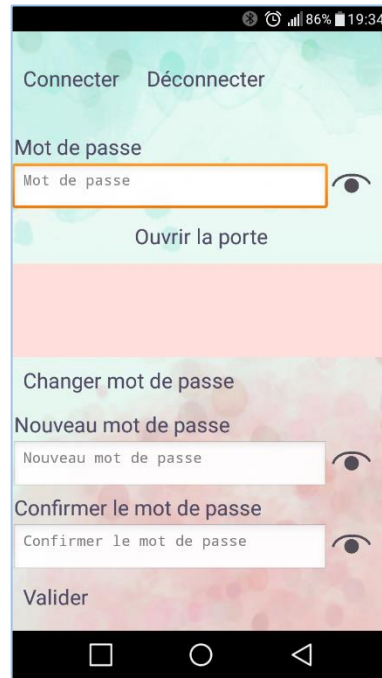


Fig.3. 30. Fenêtre de déverrouillage.

3.8 Conclusion

Dans ce chapitre nous avons présenté les cinq scénarios que nous avons réalisés pour notre maison en expliquant leur fonctionnement accompagné d'une représentation schématique.

Ensuite, nous avons cité le matériel utilisé pour chaque réalisation ainsi que les branchements des circuits.

Enfin, nous avons également présenté brièvement le logiciel utilisé pour la création de l'application Android utilisée pour déverrouiller la serrure.

Conclusion générale

Conclusion générale.

Ces dernières années, l'informatique, l'électronique et la technologie de communication ont connu un large développement, et ont été appliquées dans la conception de maisons intelligentes, qui a pour but d'assister l'habitant dans diverses situations domestiques, lui garantir le confort, améliorer les conditions de vie et le sentiment de sécurité et permet l'économie de l'énergie.

Dans ce cadre, nous avons essayé de développer un système domotique en traitant le concept de l'intelligence dans un espace réduit celui de la maison. Une maison qui permet de contrôler des dispositifs domestiques localement ou à distance. Pour cela, nous avons considéré les deux cartes Arduino utilisées UNO et MEGA comme les deux cerveaux du système, nous avons également exploité le protocole de communication Bluetooth, qui assure la liaison entre le système et l'application Android que nous avons créé.

Notre projet avait pour but de faire fonctionner certains appareils domestiques automatiquement l'éclairage à l'intérieur de contrôler à l'extérieur, de prévenir en cas d'incendie ou de fuite de gaz, et qui permet l'ouverture du garage à distance ainsi que de déverrouiller une serrure codée soit avec clavier soit en utilisant son smartphone pour plus de sécurité.

Au cours de la réalisation de ce projet, nous avons rencontré certaines difficultés au niveau dans la programmation pour intégrer toutes les réalisations dans un seul et même programme et au niveau de la disponibilité du matériel, mais malgré cela, nous avons pu atteindre les objectifs assignés à ce projet.

L'opportunité qui nous a été offerte pour travailler sur ce projet de fin d'étude, nous a permis de renforcer notre capacité à travailler en groupe et d'enrichir nos connaissances acquises durant le cursus universitaire ainsi que l'ouverture à un domaine très prometteur, celui de la domotique et nous familiariser avec ses différentes applications, ce qui pourrait nous être fortement utile dans notre vie.

Pour ce qui est des perspectives d'amélioration, plusieurs voies peuvent être envisagées pour plus de confort et de sécurité. Nous pouvons développer d'autres fonctionnalités et les ajouter à notre système domotique tel que la gestion des multimédia, la surveillance interne et externe avec des caméras, un système d'arrosage automatique. Comme nous pouvons utiliser internet pour contrôler les fonctions domotiques.

Références

- [1] : *T.MEKHALFIA* et *T.GHEDBANE*. « Etude et réalisation d'un système de commande à distance des installations électriques pour la domotique ». Thèse de Master académique. Université de M'sila, 2018.
- [2] : *M.INGUEL*. « Conception et réalisation d'un système Domotique par GSM ». Thèse de Master académique. Université de Tizi Ouzou, 2017.
- [3] : *N.ALMALI*, *K.SALIH BAHIR* et *Ö.ATAN*. « ARDUINO BASED SMART HOME AUTOMATION SYSTEM ». International Journal of Scientific Research in Information Systems and Engineering. 2016, VOLUME 2, ISSUE 2, p. 1-5.
- [4] : *C.LOCQUENEUX*, « Le guide de la maison et des objets connectés », Edition Eyrolles,2016.
- [5] : *A.KAOUBI*. « Maison intelligente ». Thèse de Master professionnel. Université virtuelle de Tunis Année, 2018.
- [6] : *A.YAHI* et *L.KOURI*. « Contrôle et suivie d'une maison intelligente via internet ». Thèse de Master, Université Akli Mohand Oulhadj-bouira, 2018.
- [7] : *A.ELHAMMOUMI* et *M.SLIMANI*. « Conception et réalisation d'un prototype d'une maison domotique intelligente, My Smart Home ». Mémoire de Master. Maroc, Université Hassan 1^{er} settat, 2016.
- [8] : *S.LANDRAULT* et *H.WEISSLINGER*. « Arduino : premier pas en informatique embarquée ». Blog d'Eskimon, 2014.
- [9] : *M.VERMA*. « Working, operatin and types of arduino microcontroller ». International journal of engineering sciences & research technology. June 2017.
- [10] : *K.HEBBACH* et *Y.OUALI* « Serrure codée à base d'arduino ». Mémoire de Licence. Université de Bejaia, 2018.
- [11] : *J.Lechalupé*. « Cours d'initiation a Arduino ». Cours. Université Paul Sabatier, 2014.
- [12] : *S.MONK*. « The tab book of arduino projects. ». MC graw hill education, 2014.
- [13] : <https://www.arduino-france.com/review/arduino-mega-utilisation-et-fonctionnement/> Consulté le (07/2020).

Références.

- [14] : *R.Santos* et *S.Santos*. « Ultimate guide for Arduino sensors/modules ». 1ere édition.
- [15] : <https://forum.arduino.cc/index.php?topic=460167.0> . Consulté le (05/2020).
- [16] : <https://code.visualstudio.com/> . Consulté le (05/2020).
- [17] : <https://github.com/ktechlab/ktechlab/wiki> . Consulté le (05/2020).
- [18] : <https://codebender.cc/> . Consulté le (05/2020).
- [19] : <http://www.microchip.com/mplab/avr-support/atmel-studio-7> . Consulté le (05/2020).
- [20] : *S.M'HID* et *A.MEBROUK*. « Conception et réalisation d'une maison intelligente. ». Thèse de Master. Maroc, université SIDI Mohamed Ben Abdellah, 2018.
- [21] : *A.KRAMA* et *A.GOUGUI*. « Etude et réalisation d'une carte de contrôle par Arduino via le système Android ». Thèse de Master académique. Université Kasdi Merbah Ouargla, 2015.
- [22] : *M.A.ZERROUKI* et *R.NESNAS*. « Conception et réalisation d'un système de commande d'une habitation ». Thèse de Master académique. Université de Mouloud Mammeri de Tizi Ouzou, 2018.

Annexes

Le programme de la serrure

```

#include <Keypad.h>           // bibliothèque pour utiliser le clavier
#include<LiquidCrystal.h>    // bibliothèque pour utiliser l'écran
LCD
#include<EEPROM.h>           // bibliothèque pour utiliser la memoire
EEPROM
#include <Servo.h>           // bibliothèque pour utiliser le
servomoteur

LiquidCrystal lcd(A0,A1,A2,A3,A4,A5); // relier l'écran aux pins
A0,A1,A2,A3,A4,A5
Servo servo;                 // déclarer le servomoteur

String inputString = "";
String commande = "";
String valeur = "";
String mot_de_passe = "1234"; // déclarer le mot de
passe

boolean stringComplet = false;
long maintenant;

int str_len = mot_de_passe.length()+1;
int Position = 0;           // déclarer la position du
servomoteur
int compte= 3;             // déclarer nombre d'essais
possible
char code[4];              // le code doit contenir 4 caractères
char touche_appuyee;
int i=0;

const byte lignes = 4;
const byte colonnes = 4;
char Keys[lignes][colonnes] = // déclarer le clavier sous forme
matricielle
{
  {'1','2','3','A'},
  {'4','5','6','B'},
  {'7','8','9','C'},
  {'*','0','#','D'}
};

byte L_pins[lignes] = {2,3,4,5}; // les pins connectées aux lignes
byte C_pins[colonnes] = {6,7,8,9}; // les pins qui représentent les
colonnes
Keypad keypad_key = Keypad( makeKeymap(Keys), L_pins, C_pins, lignes,
colonnes); // construire le clavier

void setup()
{
  lcd.begin(16,2);
  lcd.setCursor(0,0);
  lcd.print("Entrez le code");
  lcd.setCursor(0,1);

```

```
servo.attach(13);
servo.write(110);

Serial.begin(9600);
inputString.reserve(50);
commande.reserve(50);
valeur.reserve(50);
boolean stringOK = false;
}

void loop()
{
    ////////// Déverrouiller la serrure avec APPLICATION //////////

    char password[str_len];
    mot_de_passe.toCharArray(password, str_len);

    if (stringComplet) {
        delay(100);
        int Position = inputString.indexOf('=');
        if (Position > -1) {

            commande = inputString.substring(0, Position);
            valeur = inputString.substring(Position+1, inputString.length()-
1);

            if(!mot_de_passe.compareTo(valeur) && (commande ==
"OUVRIR")) // conditions d'ouverture de la porte: - le code entré =
mot de passe

/
- Le bouton OUVRIR sur application /
est appuyé
    {    maintenant=millis();

        PorteOuverte(); // faire appel a la fonction
PorteOuverte

        lcd.clear();
        lcd.setCursor(0,0);
        lcd.print("    Acces    ");
        lcd.setCursor(0,1);
        lcd.print("    Autorise ");
        compte=3;

    }
    else
    {
        lcd.clear();
        lcd.setCursor(0,0);
        lcd.print(" code incorrect ");
    }

    inputString = "";
    stringComplet = false;
}
}

if(millis()-maintenant==5000)
```

```
{ PorteFermee();
  lcd.clear();
  lcd.print(" Entrez le code ");
  lcd.setCursor(0,1);
  i=0;
}

if(commande=="VALIDER")
{
  mot_de_passe=valeur;
}
//////////Déverrouiller la serrure avec CLAVIER//////////

touche_appuyee = keypad_key.getKey();

  if (touche_appuyee) // si une touche est
appuyée
  {
    code[i++]=touche_appuyee; // taper le mot de
passe
    lcd.print('*'); // afficher (*)
    // Serial.print(code);
  }

  if((touche_appuyee=='A') && (compte > 0)) // si a touche A est
appuyé et il reste au moins un essai
  {
    if(!(strcmp(code,password,4)) ) // si le code tapé = le
mot de passe
    {
      lcd.clear();
      lcd.setCursor(0,0);
      lcd.print("    Acces    ");
      lcd.setCursor(0,1);
      lcd.print("    Autorise  ");

      delay(100);

      servo.write(20); // ouverture
      delay(5000);

      servo.write(110); // fermeture
      compte=3;
      lcd.clear();

      lcd.clear();
      lcd.print(" Entrer le code ");
      lcd.setCursor(0,1);
      i=0;
    }
  }

else

{
  compte=compte-1;
  lcd.clear();
  lcd.print(" code incorrect ");
  delay(1000);

  lcd.clear();
  lcd.print(" Entrez le code ");
}
```



```
    lcd.setCursor(0,1);
    i=0;

    if(compte<1)
    {
    lcd.clear();
    lcd.print(" La serrure est ");
    lcd.setCursor(0,1);
    lcd.print("   bloquee   ");
    delay(1000);

//    lcd.clear();
//    lcd.print("  Debloquez la  ");
//    lcd.setCursor(0,1);
//    lcd.print(" avec telephone ");
    }
    }
}

void serialEvent()
{
  while (Serial.available())
  {
    char inChar = (char)Serial.read();
    inputString += inChar;
    if (inChar == '\n' || inChar == '\r')
    {
      stringComplet = true;
    }
  }
}

void PorteOuverte()
{
  servo.write(20);
  delay(100);
}

void PorteFermee()
{
  servo.write(110);
  delay(100);
}
```

Le programme du reste des applications

```
//VARIABLES DES INTERRUPTEURS DE LUMIERE/////
int interrupteur_garage=20;
int interrupteur_cuisine= 21;
int lumiere_garage=22;
int lumiere_cuisine=23;
int etat_garage=0;
int etat_cuisine=0;

////////VARIABLE DU GARAGE////////
#include <IRremote.h> //librairie pour la communication infrarouge
#include <Servo.h> //on inclut une librairie pour utiliser le
servomoteur
Servo monServo; //on déclare l'utilisation d'un servomoteur nommé
"monServo"
#define pin_recepteur_infra 51 //variable contenant le numéro du pin ou
est cooncté le recepteur infrarouge
IRrecv monRecepteur_infra(pin_recepteur_infra); //création d'un nom
pour le recepteur
decode_results message_recu; //variable contenant le message reçu par
le recepteur infrarouge
int bouton_garage=2;
bool ETAT=0;

///// VARIABLES D ECLAIRAGE////////
int pir = 7;
int lampe = 3;
int photoR = A0;
int boutonInt = 5;
int valeur = 0;
int app = 0;
int lumiere = 0;
bool act = 1;
int leDelay = 10000;
long instant = 0;

///////// VARIABLES D ALARME //////////
int capteurGaz = 12;
int boutonAlarme = 11;
#define buzzer 9
bool activation = 1;
int etat = 0;
long moment = 0;
int monDelay = 10000;

/////////VARIABLE ECLAIRAGE EXT /////
int boutonExt = 45;
int val=0; // état du bouton
int lumiereON = 0; // état de la lumière
int appuyer = 0; // état d'appuie
bool active = 1;
const byte led[] = {31,33,35,37,39,41};
int photo_R=A15;

void setup() {
  Serial.begin(9600);

  ////INERRUPTEURS////////
```

```
pinMode(interrupteur_garage, INPUT);
pinMode(interrupteur_cuisine, INPUT);
pinMode(lumiere_garage, OUTPUT);
pinMode(lumiere_cuisine, OUTPUT);

//// ECLAAIRAGE INT/////
pinMode(lampe, OUTPUT);
pinMode(pir, INPUT);
pinMode(photoR, INPUT);
pinMode(boutonInt, INPUT);

////alarme////////
pinMode(buzzer, OUTPUT);
pinMode(capteurGaz, INPUT);
pinMode(boutonAlarme, INPUT);
etat = digitalRead(boutonAlarme);

////ECLAIRAGE EXT/////
pinMode(boutonExt, INPUT);
pinMode(photo_R, INPUT);
for (int i=0; i<5; i++)
pinMode(led[i], OUTPUT);

//////////GARAGE////////
monServo.attach(3); //on déclare la broche de connection du
servo(digitale 11 PWM)
monServo.write(90); //onmet le servomoteur en arrêt
monRecepteur_infra.enableIRIn(); //commande pour activer le module
infrarouge
monRecepteur_infra.blink13(true); //active une led lors de la
reception des données

pinMode(bouton_garage, INPUT);
attachInterrupt(digitalPinToInterrupt(2), garage, RISING);
}

void loop()
{
    //INTERRUPTEUR
    if(digitalRead(interrupteur_garage)==HIGH)
    etat_garage= 1-etat_garage;
    if(digitalRead(interrupteur_cuisine)==HIGH)
    etat_cuisine=1-etat_cuisine;
    interrupteur();

    //ECLAIRAGE INT
    valeur = digitalRead(boutonInt);
    eclairage();

    //ALARME
    alarme();

    //ECLAIRAGE EXT
    eclairageExt();

    //GARAGE
    if(monRecepteur_infra.decode(&message_recu))
    {
        Serial.println(message_recu.value, HEX);
        delay(500);
    }
}
```

Annexe.

```
    monRecepteur_infra.resume();    //permet au recepteur de recevoir de
nouveaux messages

    if (message_recu.value == 0xFFC23D)
    {    monServo.write(180);
      ETAT=1;    }
    if (message_recu.value == 0xFF22DD)
    {    monServo.write(90);
      ETAT=0;    }
  }
}

void eclairage()
{
  if (digitalRead(pir) == HIGH && analogRead(photoR) < 100 && act == 1)
  {
    digitalWrite(lampe, HIGH );
    instant = millis();
  }

  {
    if (millis() - instant > leDelay)
      digitalWrite(lampe, LOW);
  }
  if (valeur == HIGH && lumiere == LOW)
  {
    app = 1 - app;
    delay(100);
  }
  lumiere = valeur;
  if (app == HIGH)
  {
    act = 0;
    digitalWrite(lampe, LOW);
  }
  else
  {
    act = 1;
  }
}

void alarme()
{
  if (!digitalRead(capteurGaz) && activation == 1)
  {
    analogWrite (buzzer, 3000);
  }
  else
  {
    analogWrite (buzzer, 0);
  }
  if (digitalRead(boutonAlarme) != etat)
  {
    moment = millis();
    activation = 0;
  }
  if (millis() - moment > monDelay)
  {
    activation = 1;
  }
}
```

```
void eclaireExt ()
{
    val=digitalRead(boutonExt);
    int lumiere = analogRead(photo_R);

    if (lumiere <80 && active ==1)
    {
        for (int i=0; i<5; i++)
            digitalWrite(led[i],HIGH);
    }
    else
    {
        for (int i=0; i<5; i++)
            digitalWrite(led[i],LOW);
    }

    if(val==HIGH && lumiereON==LOW)
    {
        appuyer=1-appuyer;    // mémoriser l'état qu'il vient de detecter
        delay(100);
    }
    lumiereON=val;           //mettre a jour
    if (appuyer==HIGH)
    {
        active=0;
    }
    else
    {
        active=1;
    }
}

void garage ()
{
    if (ETAT==0)
    { monServo.write(180);
      ETAT=1; }
    else if (ETAT==1)
    { monServo.write(90);
      ETAT=0; }
}

void interrupteur ()
{
    if (etat_garage=HIGH)
        digitalWrite(lumiere_garage,HIGH);
    else
        digitalWrite(lumiere_garage,LOW);

    if (etat_cuisine,HIGH)
        digitalWrite(lumiere_cuisine,HIGH);
    else
        digitalWrite(lumiere_cuisine,LOW);
}
```

Résumé

Les maisons intelligentes œuvrent à faciliter la vie de tous les jours des utilisateurs cela de différentes manières et de la rendre plus confortable grâce à l'utilisation de la technologie.

Ce travail consiste à réaliser une interface de contrôle d'une maison à base de cartes Arduino MEGA et UNO. Le programme est écrit en langage Arduino et communique avec ces cartes via un port série (USB), afin de faciliter les tâches répétitives du quotidien tel qu'ouvrir ou fermer la porte par clavier ou par application sur smartphone conçu avec APP inventor et le garage avec un récepteur infrarouge plus un transmetteur (télécommande), contrôle de l'éclairage et enfin une alarme incendie.

Abstract

Intelligent home environments, attempt to ease the user's life in different ways and make it more comfortable using the technology.

This project presents the principle of realizing a home control interface based on an Arduino MEGA and UNO. The program is written in Arduino language communicate with the boards via the serial port (USB), thus facilitating repetitive tasks such as opening and closing the door with either the keypad or a smartphone application made with APP inventor and the garage using an IR receiver and a remote IR transmitter, a lighting control and a fire/gas alarm.

Mots clé : ARDUIONO, carte MEGA, maison_intelligente, smartHome, home_automation