



FACULTE DES SCIENCES D'ELJADIDA

Formation de Licence fondamentale en Sciences de la matière Physique

Option : Electronique

MEMOIRE DE FIN D'ETUDES

**Réalisation d'un système de sécurité et d'alarme à
base d'Arduino**

Préparé par :

➤ EL YAMINI Ahmed

Et

➤ EL ISMAILI Ghizlane

Sous la Direction de **EL FAJRI** , *Professeur à la Faculté Des Sciences d'El-Jadida, Université Chouaib Doukkali*

Soutenu devant le jury :

- ◆ Prof. EL FAJRI : FSJ
- ◆ Prof. AGNAOU: FSJ
- ◆ Prof. RMAILY : FSJ

REMERCIEMENTS

Nous remercions d'abord le Dieu qui nous a donné le courage, la patience, la santé et la volonté d'arriver à la fin de ce travail.

Nous tenons à remercier profondément **Pr. EL FAJRI**, Professeur à l'Université Chouaib Doukkali, faculté des sciences d'El Jadida, qui nous avoir confié ce travail ainsi que pour son encadrement et ses orientations avisées tout au long de ce travail, et pour ses remarques constructives, et pour sa disponibilité et ses précieux conseils qui nous avons permis d'améliorer ce travail

Nos vifs remerciements vont également aux membres du jury pour l'intérêt qu'ils ont porté à notre recherche en acceptant d'examiner ce travail et de l'enrichir par leurs propositions.

Nous adressons également nos remerciements à Kamal El IRAOUI, un étudiant diplômé de l'Université Chouaib Doukkali, faculté des sciences d'El Jadida, pour leur aide, et leurs précieux partages d'expérience.

Nous remercions très profondément nos chers parents, pour nous avoir soutenu constamment durant toute notre vie et pour avoir fait de nous ce que nous somme aujourd'hui, et pour soutien leurs financier et pour le financement de notre projet.

RESUME

Le concept de base de ce projet est la conception et réalisation d'un système de sécurité et d'alarme connectée à base d'une carte Arduino UNO. L'objectif de ce système est de pouvoir sécuriser une maison, une entreprise ou tout autre espace en détectant les intrusions ou les mouvements suspects grâce au capteur PIR. La mise en marche du système commence après l'activation par un code saisi par un clavier numérique, ce dernier permet également de changer le mot de passe et de désactiver l'alarme. Avec un afficheur LCD on visualise les informations relatives au fonctionnement du système. Une fois l'intrusion détectée, le l'utilisateur va s'alerter par une sonore.

Mots clés : Arduino, Système d'alarme, PIR, ISIS Proteus.

Table des matières

REMERCIEMENTS	1
RESUME.....	2
TABLE DES MATIERES.....	3
LISTE DES FIGURES	6
INTRODUCTION GENERALE.....	8
CHAPITRE 1 : GENERALITES SUR LA CARTE ARDUINO.....	9
I.1. INTRODUCTION :.....	10
I.2. DEFINITION :	10
I.3. HISTORIQUE DU PROJET ARDUINO :.....	11
I.4. LES TYPES DE LA CARTE :	11
a. La carte arduino mega :	12
b. La carte arduino nano :	12
c. La carte arduino due :	13
d. La carte arduino MKR :.....	13
I.5. LA CARTE ARDUINO UNO :	14
I.5.1. Les composantes d'Arduino Uno :.....	15
I.5.1.1. Le microcontrôleur Atmel :	16
I.5.1.2. entrées analogiques :.....	17
I.5.1.3. Broches PWM :.....	17
I.5.1.4. entrée/sortie numériques :.....	17
I.5.1.5. connecteur USB :.....	17
I.5.1.6. prise jack :.....	17
I.5.1.7. Reset :	17
I.5.1.8. Horloge :	18
I.5.1.9. LED :.....	18
I.5.2. L'alimentation de l'Arduino :	18
I.5.3. Les caractéristiques techniques de l'Arduino Uno :.....	19
I.6. CONCLUSION :	19
CHAPITRE 2 : MATERIELS ET COMPOSANTS	20
II.1. INTRODUCTION :	21
II.2. CAPTEUR A INFRAROUGE :	21

II.2.1. Capteur à infrarouge HC-SR501 :	21
II.2.2. Caractéristiques du capteur PIR :	22
II.2.3. Connexion d'un Capteur à infrarouge à L'Arduino :	23
II.3. ÉCRAN LCD 16X2 :	23
II.3.1. Broches d'écran LCD 16x2 :	24
Connexion d'un LCD à L'Arduino :	24
I.6.2.	25
II.4. LE BUZZER :	25
II.4.1. Définition :	25
II.4.2. Connexion d'un Buzzer à un microcontrôleur Arduino :	25
II.5. CLAVIER NUMERIQUE 4X4 :	26
II.5.1. Principe de fonctionnement :	27
II.5.2. Connexion d'un clavier numérique à un microcontrôleur Arduino :	27
II.6. PLAQUE D'ESSAI : (BREAD BOARD).....	28
II.7. POTENTIOMETRE ROTATIF :	28
II.8. CONCLUSION :	29
CHAPITRE 3 : PRESENTATION DES LOGICIELS UTILISES	30
III.1. L'IDE ARDUINO :	31
III.1.1. Introduction :	31
III.1.1. Présentation de l'interface de l'IDE Arduino :	31
III.1.2. Choisir de la carte arduino :	32
III.1.3. Ouvrir l'exemple BLINK :	33
III.1.4. Bibliothèques électroniques :	33
III.1.5. La structure d'un programme :	33
III.2. PRESENTATION DU LOGICIEL PROTEUS :	34
III.2.1. Introduction :	34
III.2.2. Présentation de l'interface Proteus 8 :	35
III.2.3. Création d'un nouveau projet :	36
III.3. CONCLUSION :	36
CHAPITRE 4 : SIMULATION ET REALISATION DU SYSTEME DE SECURITE ET D'ALARME.....	37
IV.1. INTRODUCTION :	38
IV.2. DESCRIPTION DE SYSTEME :	38
IV.3. ORGANIGRAMME DU SYSTEME DE SECURITE ARDUINO :	38
IV.4. SIMULATION DU SYSTEME D'ALARME :	39

IV.5. LA REALISATION FINALE DE PROJET :	41
IV.5.1. Téléversement du code vers Arduino :.....	41
IV.5.2. Mise en marche du système :	41
IV.5.2.1. Activation du système de sécurité et d'alarme :	42
IV.5.2.2. Changement du mot de passe pour le système de sécurité :.....	43
IV.6. CONCLUSION :	44
CONCLUSION GENERALE :	45
BIBLIOGRAPHIE :	47
ANNEXE :	48

LISTE DES FIGURES

Figure I-1 la carte arduino Mega	12
Figure I-2 la carte arduino nano	12
Figure I-3 la carte Arduino due	13
Figure I-4 la carte Arduino MKR 131	14
Figure I-5 Arduino Uno	15
Figure I-6 Arduino Uno SMD.....	15
Figure I-7 Présentation de la carte ARDUINO-UNO	15
Figure I-8 les LED	18
Figure II-2 principe général des capteurs	21
Figure II-2 Capteur de mouvement	22
Figure II-3 Schéma de câblage de PIR avec Arduino.....	23
Figure II-4 Schéma de câblage pour LCD avec Arduino.....	25
Figure II-5 Buzzer	25
Figure II-6 Schéma de câblage pour Buzzer avec Arduino	26
Figure II-7 Clavier numérique 4X4	26
Figure II-8 Disposition des touches dans un clavier numérique 4*4.....	27
Figure II-9 Schéma de câblage pour clavier numérique avec Arduino.....	28
Figure II-10 Plaque d'essai.....	28
Figure II-9 Potentiomètre rotatif	29
Figure III -3 Présentation de l'interface graphique de l'IDE Arduino.....	31
Figure III -2 Interface de sélection de la carte Arduino dans l'IDE.....	32
Figure III -3 Exemple de code : utilisation de la fonction Blink.....	33
Figure III -4 Utilisation des bibliothèques dans l'IDE Arduino	33

Figure III -5 Exemple Blink	34
Figure III -6 Proteus 8.....	35
Figure III -7 Interface de Proteus 8	35
Figure III -8 Créer un nouveau projet sur ISIS	36
Figure IV-4 Schéma du système de sécurité et d'alarme	38
Figure IV-2 Organigramme de fonctionnement général de notre système.....	39
Figure IV-3 Schéma du circuit électrique du système sur Proteus	40
Figure IV-4 la communication entre le capteur PIR, l'écran LCD et Arduino dans la simulation	40
Figure IV-5 Vue d'ensemble du matériel utilisé.....	41
Figure IV-6 Procédure de téléversement du code vers l'Arduino.....	41
Figure IV-7 Réalisation pratique du système de sécurité et d'alarme	42
Figure IV-8 Compte à rebours avant l'activation du système	42
Figure IV-9 Affichage de l'état du système sur l'écran LCD après l'activation.....	43
Figure IV-10 Déclenchement de l'alarme	43
Figure IV-11 Changement du mot de passe sur l'écran LCD du système de sécurité.....	44

INTRODUCTION GENERALE

En vue que les cambriolages et les vols sont des incidents fréquents et très préjudiciables qui peuvent causer des pertes matérielles importantes, ainsi que des dommages psychologiques irréparables aux occupants des lieux, ce qui est devenu une source d'anxiété pour de nombreuses personnes, faisant de la sécurité une priorité absolue. Le besoin de surveiller et sécuriser la vie humaine et les biens est devenue une préoccupation majeure de la plupart des gens. D'où l'importance de mettre en place un système d'alarme efficace qui permettrait une détection rapide et une réponse appropriée en cas d'intrusion dans une habitation pour qu'il soit la première étape consiste à renforcer la sécurité d'un domicile ou d'un établissement, pour prévenir les intrusions.

Il est essentiel de prendre toutes les mesures nécessaires en installant un système de sécurité sonore, ce qui peut aider à réduire la possibilité d'intrusion dans une résidence

Tout système d'alarme doit être capable de détecter les événements mettant en danger les personnes ou les biens, tels que les cambriolages, les agressions et les incendies avec des détecteurs. Au-delà de cela, un système d'alarme devrait comporter un panneau de contrôle programmable et des dispositifs d'avertissement pour signaler la survenue d'un événement.

Dans ce travail, nous proposons la conception d'un système de sécurité et d'alarme basé sur la technologie d'Arduino. Ce système vise à prévenir les intrusions dans une maison ou un établissement en détectant les mouvements suspects à l'aide d'un capteur et en guidant l'utilisateur pour assurer une utilisation efficace de ce système.

Le travail décrit dans ce mémoire est conçu en quatre chapitres : Le premier chapitre pour la présentation de la carte Arduino d'une manière générale et d'une manière précise la carte Arduino UNO. Le second chapitre est consacré à la présentation des composants matériels adopté pour la réalisation du système d'alerte. Le troisième chapitre est consacré à la présentation du logiciel IDE adopté pour programmer le microcontrôleur du carte Arduino, et du logiciel Proteus ISIS adopté pour simuler le fonctionnement du circuit avant de l'exécuter en pratique. Le dernier chapitre est consacré à la simulation et à la réalisation de notre système de sécurité et d'alarme.

CHAPITRE I

GENERALITES SUR LA CARTE ARDUINO

I.1. Introduction :

Les systèmes embarqués sont des systèmes électroniques intégrés dans la fabrication de nombreux appareils électriques qui nous entourent. Ils sont autonomes sur le plan électronique et informatique, intégrant du matériel et des logiciels conçus ensemble pour réaliser en temps réel une tâche précise au sein de l'appareil auquel ils sont intégrés. Ils fonctionnent grâce à des microprocesseurs ou des microcontrôleurs et présentent deux caractéristiques principales : une taille limitée et une consommation énergétique assez faible.

Arduino est la carte électronique la plus populaire pour développer et programmer des systèmes embarqués. Dans ce chapitre, nous allons présenter la carte Arduino de manière générale, la carte Arduino Uno en détail, les différents types de cartes Arduino.

I.2. Définition :

L'Arduino est une plate-forme de prototypage d'objets électronique open source. Elle est constituée d'une carte électronique programmable et d'un logiciel multiplateforme. La carte Arduino repose sur un circuit intégré basé sur un microcontrôleur, elle permet l'accès simple à l'informatique embarquée.

La carte Arduino est équipée d'entrées et de sorties qui permettent à l'utilisateur de brancher différents types d'éléments externes :

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.

Sorties : des actionneurs qui agissent sur le monde physique telle qu'une lampe ou un buzzer, etc.

Ce projet comprend trois éléments principaux : une carte électronique, un environnement de programmation et une communauté qui échange. Arduino est un projet open source publié sous licence libre, qui permet à l'utilisateur de réaliser ses propres projets, comme la fabrication d'un robot ou la télécommande d'un appareil, par l'expérimentation directe et sans avoir besoin d'une formation en électronique. La carte électronique Arduino est basée sur un microcontrôleur, qui est un petit ordinateur intégré dans un système de petite dimension. Le microcontrôleur est l'élément principal de l'Arduino. Il est programmable et dispose de plusieurs entrées et sorties permettant de lire et de générer des signaux électriques.

Il est vrai qu'il existe plusieurs plateformes qui ont le même rôle que l'Arduino, comme le Raspberry Pi. Cependant, il y a plusieurs raisons pour lesquelles on choisit l'Arduino plutôt que d'autres plateformes. L'un des avantages de l'Arduino est sa facilité d'accès et son utilisation simple. En effet, l'Arduino est conçu pour être facile à utiliser et à programmer, même pour les débutants en électronique. De plus, les cartes Arduino sont relativement peu coûteuses et qu'elles sont donc un choix populaire pour les débutants en électronique. En effet, le prix est souvent un facteur important pour les débutants qui cherchent à se lancer dans l'électronique et l'informatique. L'Arduino est très abordable et peut être utilisé pour une grande variété de projets. Il est également très flexible et peut être utilisé avec une grande variété de capteurs et d'autres composants électroniques. Enfin, l'Arduino dispose d'une grande communauté de développeurs qui partagent leurs projets et leurs connaissances en ligne.

I.3. Historique du projet arduino :

En 2003, les étudiants de l'École de Design d'Interaction d'Ivrea ont été confrontés à un défi majeur : les outils disponibles pour créer des projets d'interactivité étaient coûteux et complexes à utiliser. Ces outils de prototypage étaient alors principalement réservés à l'ingénierie et à la robotique, et leur processus de développement était long et difficile, ce qui les rendait peu accessibles aux débutants. Pour répondre à ce besoin, une équipe composée d'enseignants et d'étudiants a entrepris de créer une plateforme de prototypage électronique abordable et facile à utiliser, destinée aux étudiants, artistes et bricoleurs. Cette plateforme a été nommée Arduino en hommage au bar où l'équipe se réunissait pour discuter de son développement. En 2005, la première carte Arduino, nommée "Arduino NG" a été lancée. Elle était basée sur un microcontrôleur Atmel AVR et permettait une programmation et une communication avec un ordinateur via un port USB. Grâce à sa simplicité et à son accessibilité, la plateforme Arduino a rapidement gagné en popularité auprès des artistes et designers du monde entier. [1]

I.4. Les types de la carte :

La famille Arduino contient une grande variété de cartes, chacune possédant ses avantages et ses inconvénients. Les cartes diffèrent en termes de :

- Nombre d'entrées et de sorties pour chaque type.
- Microcontrôleur : la capacité de stockage et la rapidité d'exécution des informations.

Dans ce qui suit, nous présentons un ensemble des cartes Arduino les plus populaires.

a. La carte arduino mega :

Est une carte microcontrôleur basée sur l'ATmega2560. Elle dispose de 54 broches d'entrée/sortie numériques (dont 15 peuvent être utilisées comme sorties PWM), de 16 entrées analogiques, de 4 ports série UART, d'un oscillateur à cristal de 16 MHz, d'une connexion USB, d'une prise d'alimentation, d'un connecteur ICSP et d'un bouton de réinitialisation.

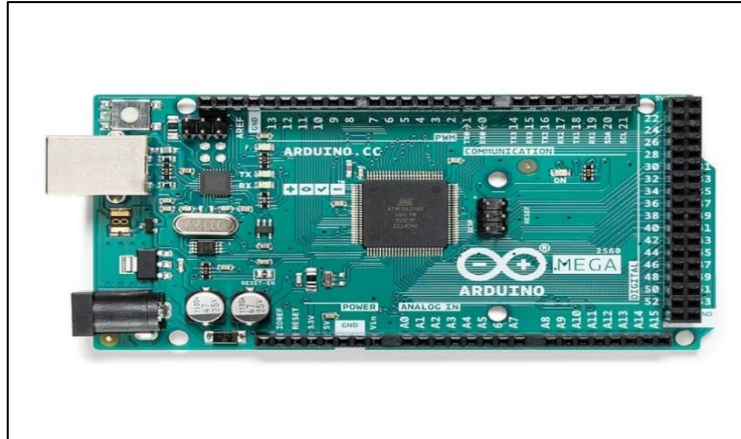


Figure I-1 la carte arduino mega

b. La carte arduino nano :

Est une carte microcontrôleur complète et compacte, compatible avec une breadboard, basée sur l'ATmega328P (pour les versions Nano jusqu'à 3.x). Elle dispose de 22 broches d'entrée/sortie, dont 14 peuvent être utilisées comme sorties PWM et 8 comme entrées analogiques. La carte Nano est très populaire pour les projets nécessitant une petite carte microcontrôleur facile à utiliser et peu encombrante. Elle ne dispose pas de prise d'alimentation DC, mais fonctionne avec un câble USB mini-B. Il existe également une version plus récente appelée "Nano Every" qui utilise un microcontrôleur plus puissant, l'ATmega4809.

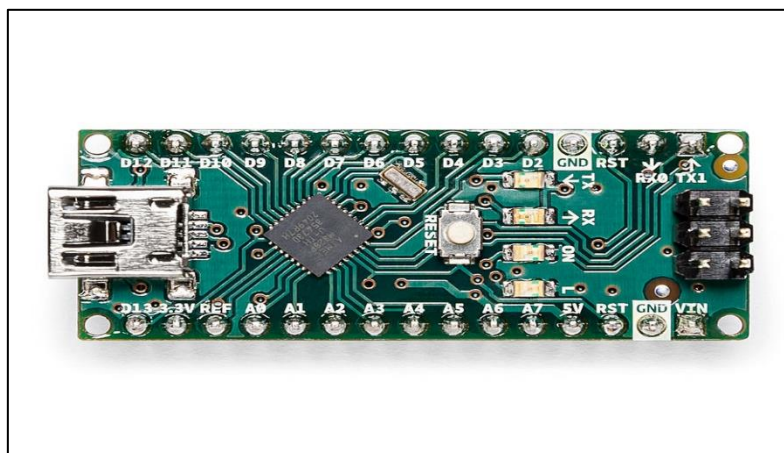


Figure I-2 la carte arduino nano

c. La carte arduino due :

Est une carte de développement basée sur un microcontrôleur ARM Cortex-M3 32 bits Atmel SAM3X8E. Cette carte possède 54 broches d'entrée/sortie numériques, 12 entrées analogiques, une fréquence d'horloge de 84 MHz, une connectivité USB, une prise Ethernet et une interface de programmation intégrée (ICSP). La carte est conçue pour offrir une plus grande puissance de traitement que les cartes Arduino plus anciennes, comme l'Arduino Uno ou l'Arduino Mega. La carte Arduino Due est compatible avec la plupart des Shields Arduino, mais elle est souvent utilisée pour des projets qui nécessitent une plus grande puissance de traitement, tels que des systèmes de contrôle de mouvement, des projets audio ou vidéo, ou des projets qui utilisent des algorithmes de traitement de signal numérique (DSP). Il est important de noter que la carte Arduino Due fonctionne à 3,3 V plutôt qu'à 5 V comme la plupart des autres cartes

Arduino. Cela signifie que certains composants électroniques conçus pour fonctionner à 5 V peuvent ne pas être compatibles avec la carte Arduino Due.

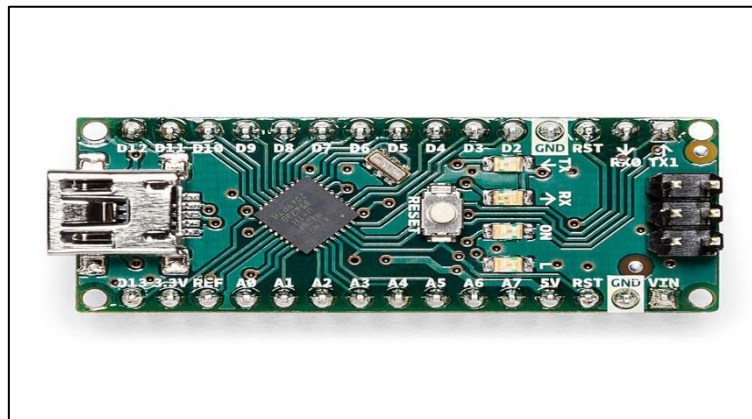


Figure I-3 la carte Arduino due

d. La carte arduino MKR :

Est une série de cartes de développement conçues pour les projets IoT (Internet des objets) et les projets à faible consommation d'énergie. Cette série comprend plusieurs modèles, tels que la MKR1000, la MKR Wifi 1010, la MKR GSM 1400, la MKR NB 1500 et la MKR FOX 1200. Les cartes Arduino MKR sont basées sur des microcontrôleurs ARM Cortex-M0+ 32 bits et disposent d'une faible consommation d'énergie, de connectivité sans fil (Wi-Fi, GSM, NB-IoT, Sigfox), d'un connecteur d'antenne et d'un connecteur pour batterie Li-Po. Elles sont également compatibles avec les Shields Arduino et disposent d'une interface de programmation intégrée (ICSP) pour la programmation et le débogage. La carte Arduino MKR1000, par exemple, dispose d'une connectivité Wi-Fi intégrée, d'un port USB, de 8 entrées/sorties

analogiques et de 12 entrées/sorties numériques. Elle est souvent utilisée pour les projets IoT, les capteurs connectés, les contrôleurs de maison intelligente, etc. La série Arduino MKR est idéale pour les projets nécessitant une faible consommation d'énergie, une connectivité sans fil et une taille compacte. [2]

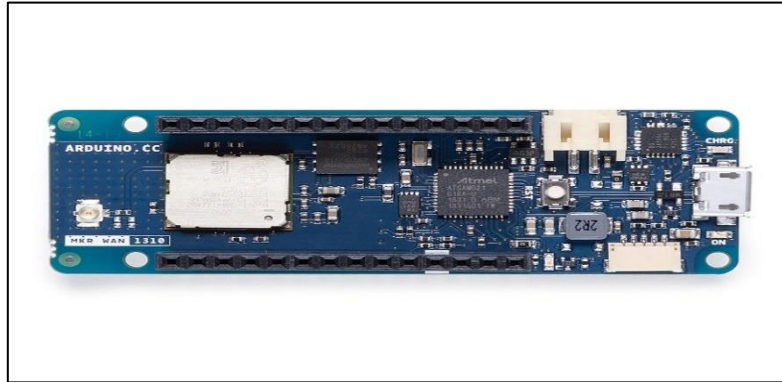


Figure I-4 la carte Arduino MKR 1310

I.5. La carte arduino Uno :

Le mot « Uno » signifie « un » en italien et a été choisi pour marquer la sortie d'Arduino Software (IDE) 1.0. La carte Uno et la version 1.0 d'Arduino Software (IDE) étaient les versions de référence d'Arduino.

La carte Arduino Uno est une carte électronique programmable qui permet de contrôler des composants électroniques tels que des capteurs, des moteurs et des LED. Elle est conçue pour être facile à utiliser, même pour les débutants en électronique et en programmation.

La carte Arduino Uno est basée sur un microcontrôleur ATmega328P, qui est le cerveau de la carte. Ce microcontrôleur est programmable à l'aide du logiciel Arduino, qui est une plateforme de programmation open source et gratuite.

La carte Arduino Uno est équipée de plusieurs broches d'entrée/sortie. Ces broches peuvent être utilisées pour connecter des capteurs, des actionneurs et d'autres composants électroniques. La carte dispose également de broches d'alimentation pour fournir du courant à ces composants.

La carte Arduino Uno est alimentée par un câble USB ou par une source d'alimentation externe. Elle est équipée d'un régulateur de tension intégré qui permet de réguler la tension d'entrée pour s'adapter aux exigences du microcontrôleur.

La carte Arduino Uno est également équipée d'un port série (UART) qui permet la communication avec d'autres dispositifs électroniques, tels que des ordinateurs ou d'autres cartes Arduino. Elle dispose également d'une interface de programmation en circuit (ICSP) qui permet la programmation du microcontrôleur directement sur la carte. Il existe deux modèles d'Arduino Uno: l'un avec un microcontrôleur de grande taille, et un autre avec un microcontrôleur dit SMD (SMD: Surface Mounted Device, soit composants montés en surface).

en opposition aux composants qui traversent la carte électronique et qui sont soudés du côté opposé). D'un point de vue utilisation,

il n'y a pas de différence entre les deux types de microcontrôleurs. Les couleurs de l'Arduino peuvent varier du bleu au bleu-vert, en fonction des modèles et années de production.[3]

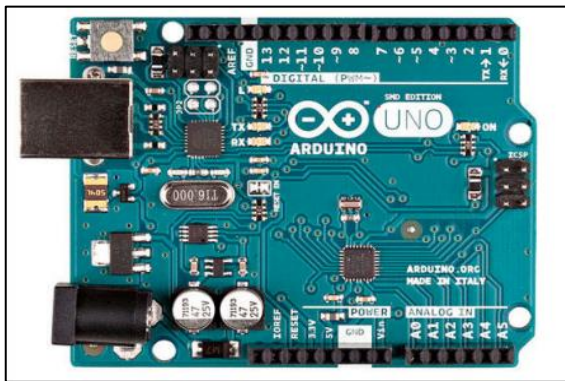


Figure I-5 Arduino Uno SMD

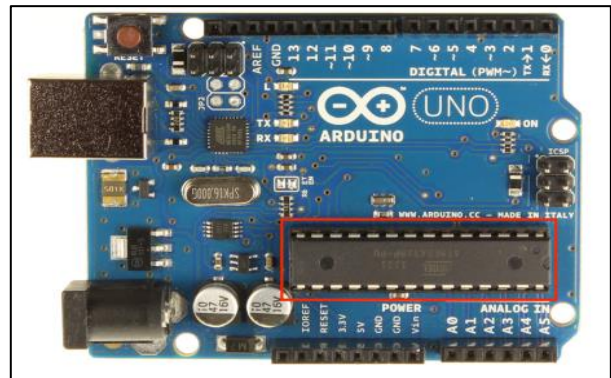


Figure I-6 Arduino Uno

I.5.1. Les composantes d'Arduino Uno :

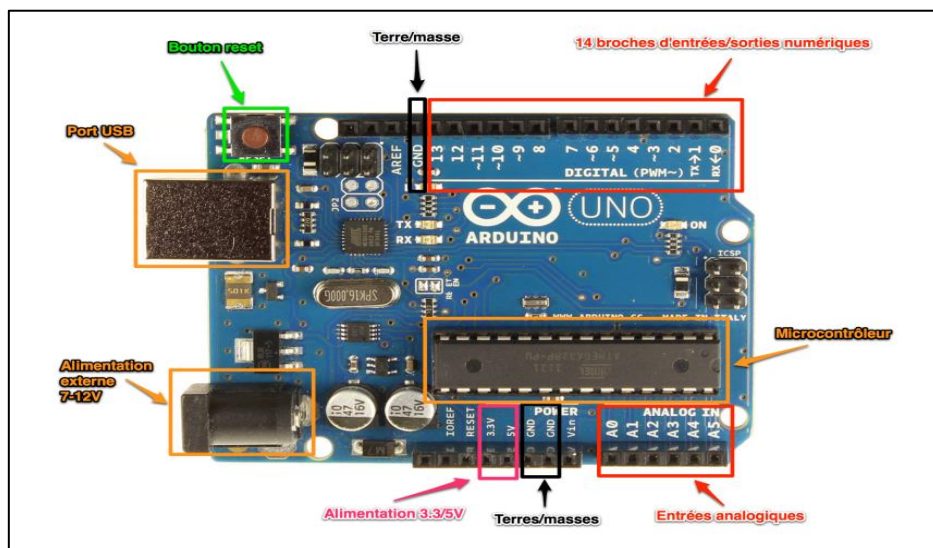


Figure I-7 Présentation de la carte ARDUINO-UNO

Arduino Uno est une carte électronique basée sur un microcontrôleur ATmega328P, qui comprend des composants comme des ports d'entrée/sortie, des ports d'alimentation, un convertisseur analogique-numérique, une mémoire SRAM et une interface USB pour la programmation et la communication avec d'autres appareils.

I.5.1.1. Le microcontrôleur Atmel :

le microcontrôleur Atmel est un circuit intégré qui rassemble les éléments essentiels d'un ordinateur : processeur, mémoires (mémoire morte pour le programme, mémoire vive pour les données), unités périphériques et interfaces d'entrées sorties.

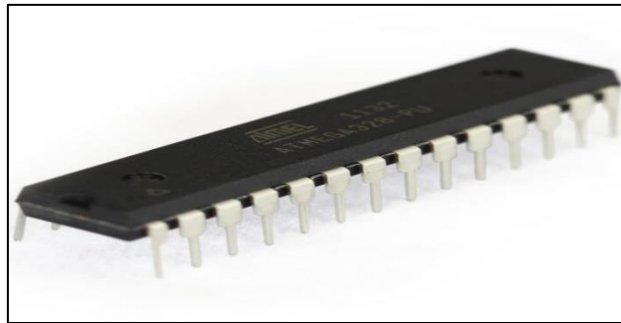


Figure I-8 microcontrôleurs ATMEGA328P-PU

Les éléments intégrés dans le microcontrôleur sont :

- **Le processeur** ; est souvent nommé CPU, c'est le composant principal qui exécute les opérations arithmétiques et logiques.
- **La mémoire** : le microcontrôleur contient cinq types de mémoires
 - _ Flashe : c'est une mémoire effaçable, c'est celle qui contient le programme à exécuter.
 - _ RAM : ou mémoire vive elle contient les variables de programme, elle s'efface lorsque on coupe l'alimentation au microcontrôleur.
 - _ EEPROM : c'est un disque dur
 - _ Les registres
 - _ La mémoire cache : elle fait la liaison entre les registres et la RAM
- **CONVERTISSEUR** :
 - _ CAN ; convertisseurs analogique-numériques,
 - _ CNA ; convertisseurs numériques-analogiques [4]

I.5.1.2. entrées analogiques :

Sur la carte Arduino Uno, il y a six entrées analogiques (A0 à A5) qui permettent de mesurer des tensions comprises entre 0 et 5 volts. Chacune de ces entrées est connectée à un convertisseur analogique-numérique intégré, qui convertit les signaux analogiques en signaux numériques pouvant être traités par le microcontrôleur de la carte Arduino .

I.5.1.3. Broches PWM :

Les broches PWM de la carte Arduino Uno sont marquées d'un (~) à côté de leur numéro de broche, soit les broches 3, 5, 6, 9, 10 et 11. Les broches PWM (Pulse-Width Modulation) sont des broches spéciales sur la carte Arduino Uno qui permettent de produire des signaux de sortie avec une largeur d'impulsion variable. il est possible de générer un signal PWM sur une broche PWM de la carte Arduino Uno par utilisation de la fonction "analogWrite()".

I.5.1.4. entrée/sortie numériques :

La carte Arduino Uno dispose de 14 broches d'entrée/sortie numériques qui peuvent être utilisées en tant qu'entrées numériques ou sorties numériques. Ces broches sont étiquetées de 0 à 13. les broches 0 et 1 sont utilisées pour la communication série (RX et TX) avec un ordinateur ou un autre dispositif série.

I.5.1.5. connecteur USB :

Permet d'alimenter la carte en Énergie électrique (5V). Permet de téléverser le programme dans la carte.

I.5.1.6. prise jack :

Permet de brancher une alimentation (pile, batterie, adaptateur Secteur, + au centre Vin 7 à 12 V

I.5.1.7. Reset :

La fonction reset permet d'interrompre le programme et de le remettre à zéro. Il est possible d'effectuer cette même opération en coupant et en remettant le courant sur la carte Arduino. A la mise sous tension, elle commencera toujours par reprendre le programme du début, ce qui équivaut donc à une réinitialisation.

I.5.1.8. Horloge :

L'horloge définit la fréquence ou la vitesse avec laquelle le microcontrôleur exécute les tâches, et est exprimée en hertz (Hz). Elle varie d'un microcontrôleur à l'autre. Sur les cartes Arduino, l'horloge est généralement cadencée à 16 MHz.

I.5.1.9. LED :

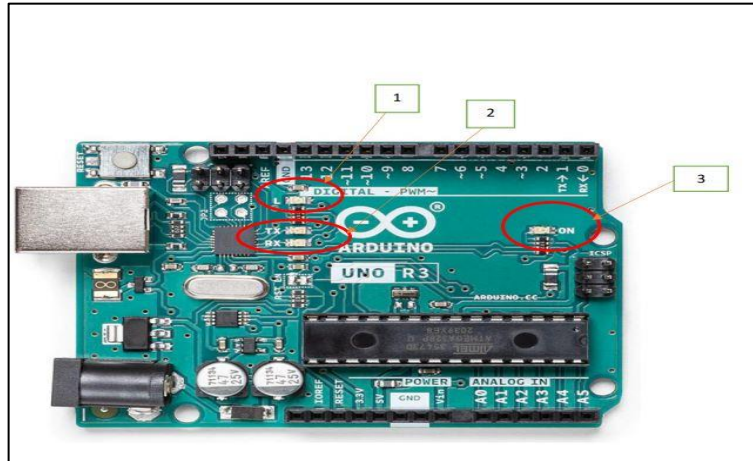


Figure I-9 Les LEDS

1 : Elle est connectée à une broche du microcontrôleur, elle clignote durant l'entrée du code, et s'arrête d'être allumée lorsque l'entrée du code est terminée.

2 : ces deux LED servent à visualiser l'activité sur la voie série

TX : LED de transmission

RX : LED de réception, clignote pendant le processus de réception

3 : le clignotement de cette LED témoigne de la bonne alimentation de la carte

I.5.2. L'alimentation de l'Arduino :

Le microcontrôleur est alimenté par une tension de 5V, au-delà de cette valeur il va s'endommager, pour éviter cela le régulateur de tension se charge de réduire la tension à 5Volts.

Il existe deux méthodes pour fournir cette tension soit par la prise USB utilisée pour la connecter à un ordinateur, soit par une des prises d'alimentation présentes sur la carte.

La carte Arduino Uno compose trois prises d'alimentation :

La prise JACK standard Cette tension doit être continue et peut par exemple être fournie par une pile 9V.

La broche VIN : Il faut raccorder la branche positive d'une source de tension (5V à 12V) à cette broche, et la broche négative sur une broche GND

La broche 5V : Il faut raccorder la branche positive d'une source de tension 5V à cette broche, et la broche négative sur une broche GND.

Connecteur USB : Cette entrée est utilisée pour relier la carte Arduino à un ordinateur, elle permet à la fois d'alimenter la carte et de faire passer le programme à exécuter au microcontrôleur.

I.5.3. Les caractéristiques techniques de l'Arduino Uno :

- _ Microcontrôleur ATmega328P
- _ Tension de fonctionnement 5V
- _ Tension d'entrée (recommandée) 7-12V
- _ Tension d'entrée (limite) 6-20V
- _ Broches d'E/S numériques 14 (dont 6 offrent une sortie PWM)
- _ Entrées analogiques 6
- _ Courant continu par broche E/S 20 mA
- _ Courant continu pour broche 3.3V 50 mA
- _ Mémoire Flash 32 KB (ATmega328P) dont 0.5 KB utilisés par le bootloader
- _ SRAM 2 KB (ATmega328P)
- _ EEPROM 1 KB (ATmega328P)
- _ Vitesse d'horloge 16 MHz
- _ LED_BUILTIN 13
- _ Longueur 68.6 mm
- _ Largeur 53.4 mm
- _ Poids 25 g [2]

I.6. Conclusion :

En conclusion, le chapitre d'introduction sur Arduino nous a donné un aperçu de l'histoire du projet Arduino, nous a présenté les différents types de cartes de développement disponibles et a souligné que la carte de développement Arduino Uno est un point de départ idéal pour les débutants.

CHAPITRE II

MATERIELS ET COMPOSANTS

II.1. Introduction :

Le choix des matériaux est une tâche fondamentale et très complexe à prendre en compte lors de la conception de tout système électronique, y compris le système d'alarme que nous sommes en train de réaliser. Pour notre projet, nous avons choisi la carte Arduino Uno comme plateforme de développement. Cette carte microcontrôleur open-source est basée sur le microcontrôleur ATmega328P et largement utilisée dans les projets électroniques en raison de sa simplicité d'utilisation et de programmation. En plus de la carte Arduino, nous avons sélectionné plusieurs autres composants pour la réalisation de notre système d'alarme.

Nous avons choisi le capteur infrarouge passif pour détecter la présence de mouvement dans la pièce, l'écran LCD pour afficher des informations importantes, le buzzer pour émettre des sons d'alarme, et le clavier numérique pour entrer le code d'accès et pour activer l'alarme. Dans le premier chapitre, nous avons déjà présenté la carte Arduino Uno. Dans ce chapitre-ci, nous allons introduire ces différents composants matériels utilisés pour la réalisation du système de sécurité et décrire leur rôle dans le système. Nous allons également expliquer comment ces composants sont connectés à la carte Arduino pour assurer un fonctionnement coordonné et efficace du système d'alarme.

II.2. Capteur à infrarouge :

Un capteur est une interface entre un processus physique et une information exploitable. Il fournit une information en fonction de la sollicitation à laquelle il est soumis grâce à une électronique à laquelle il est associé.

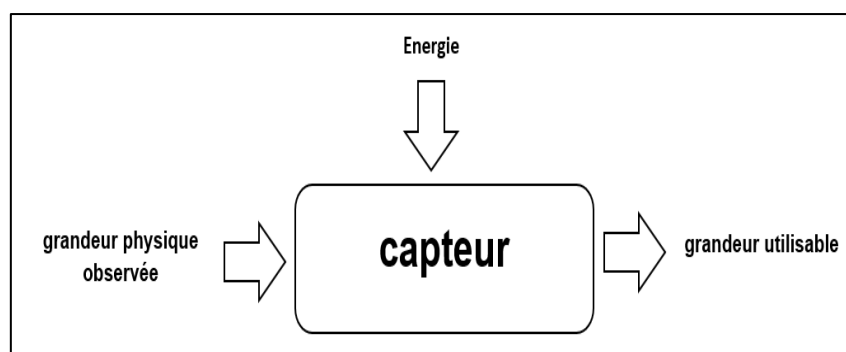


Figure II-1 principe général des capteurs

II.2.1. Capteur à infrarouge HC-SR501 :

Le capteur de mouvement infrarouge passif ou PIR (Passive Infra Red Sensor) est une pièce électrothermique, qui peut être utilisé pour détecter la présence de mouvement en

détectant les changements dans les niveaux de rayonnement infrarouge émis par les objets environnants et en le convertissant en un signal électrique, ce signal sera lu par le microcontrôleur.

Le terme passif signifie que le capteur n'utilise aucune énergie pour le processus de détection, il fonctionne uniquement en détectant l'énergie par d'autres objets.

Le détecteur infrarouge passif repose sur la réception du rayonnement infrarouge du corps humain pour déclencher une alarme ; lorsque le corps humain ou l'animal passe dans la plage de la détection de la sonde thermoélectrique, il détectera son mouvement en raison de l'énergie thermique émise du corps sous forme du rayonnement infrarouge. La température de surface du corps humain est de 36-27 ° C, et la majeure partie de son énergie rayonnante est concentrée dans la gamme de longueurs d'onde de 8-12 μm .

Le terme passif signifie que le capteur n'utilise aucune énergie pour le processus de détection, il fonctionne uniquement en détectant l'énergie par d'autres objets.

Le couvercle en plastique blanc est une lentille de Fresnel pour élargir l'angle de détection, et pour focaliser les signaux infrarouges sur un électrothermique sensor, ce couvercle peut être retiré pour voir le capteur pyroélectrique.

La sortie du capteur de détection de mouvement PIR peut être connectée directement à l'une des broches numériques Arduino (ou n'importe quel microcontrôleur). Si un mouvement est détecté par le capteur, cette valeur de broche sera définie sur « 1 ». Les deux potentiomètres sur la carte vous permettent d'ajuster la sensibilité et le temps de retard après la détection d'un mouvement.[5]

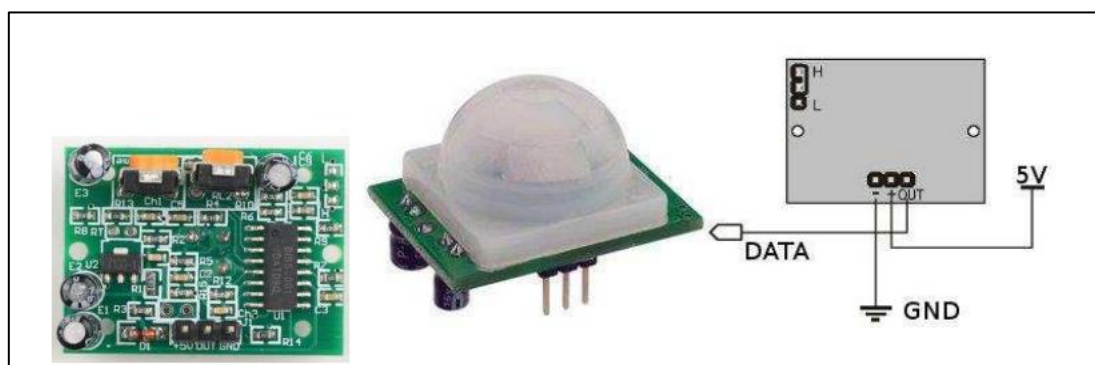


Figure II-2 Capteur de mouvement

II.2.2. Caractéristiques du capteur PIR :

- Large plage de tension d'entrée variant de 4.V à 12V (+5V recommandé)
- La tension de sortie est élevée/basse (3,3 V TTL)
- Peut faire la distinction entre le mouvement de l'objet et le mouvement humain
- Modes de fonctionnement - Répétable (H) et Non répétable (L)
- Distance de couverture d'environ 120° et 7 mètres
- Faible consommation d'énergie de 65mA
- Température de fonctionnement de -20° à +80° Celsius [6]

II.2.3. Connexion d'un Capteur à infrarouge à L'Arduino :

- _ Connectez la broche VCC du capteur PIR à la broche 5V de l'Arduino.
- _ Connectez la broche GND du capteur PIR à la masse (GND) de l'Arduino.
- _ Connectez la broche OUT du capteur PIR à une broche d'entrée numérique de l'Arduino (par exemple, la broche 5).

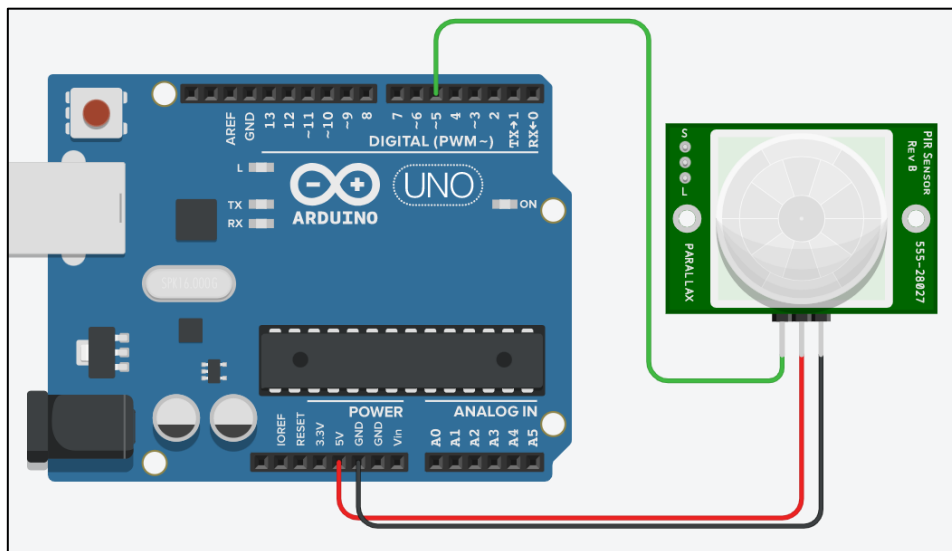


Figure II-3 Schéma de câblage de PIR avec Arduino

II.3. Écran LCD 16x2 :

LCD signifie "Liquid Crystal Display" et se traduit en français, par "Écran à Cristaux Liquides".

Les caractères peuvent s'allumer grâce à une grille carrée de cinq par sept pixels, chaque pixel peut avoir deux états ;1 noir ou 0 blanc.

La base du principe de fonctionnement d'un écran LCD repose sur l'orientation des cristaux liquides et la polarisation de la lumière par des filtres polarisants, on distingue deux cas :

- Lorsqu'aucune tension n'est appliquée entre les électrodes correspondant au pixel de l'écran, les cristaux liquides font tourner le plan de polarisation progressivement d'une électrode à l'autre. La lumière polarisée verticalement arrive polarisée horizontalement sur le polariseur horizontal. Elle peut donc passer
- Lorsqu'une tension suffisante est appliquée entre les électrodes, les cristaux liquides s'orientent perpendiculairement aux électrodes (selon les lignes de champ électrique. Les cristaux liquides ne font alors plus tourner le plan de polarisation de la lumière. La lumière reste polarisée verticalement, rencontre le polariseur horizontal et ne le traverse pas. Le pixel est noir (opaque).

Si on joue sur la tension entre les électrodes, on arrive à un état intermédiaire qui offre des nuances de gris. L'écran LCD est un dispositif passif, il nécessite un rétro-éclairage car il n'émet pas de lumière.

II.3.1. Broches d'écran LCD 16x2 :

Les écrans LCD ont une interface parallèle, ce qui signifie que le microcontrôleur doit manipuler plusieurs broches d'interface à la fois pour contrôler l'affichage. L'interface se compose des broches suivantes :

- **GND ET +5V** ; Broches d'alimentation
- **VO** ; broche de contraste, elle permet de régler la résistance qui est par rapport au LED
- **RS** ; Broche de sélection de registre, qui contrôle l'emplacement dans la mémoire de l'écran LCD où les données sont écrites. On peut sélectionner soit le registre de données, qui contient ce qui se passe à l'écran, soit un registre d'instructions, où le contrôleur de l'écran LCD recherche des instructions sur ce qu'il faut faire ensuite.
- **RW** ; Broche de lecture/écriture, qui sélectionne le mode de lecture ou le mode d'écriture
- **E** ; Broche Enable qui permet d'écrire dans les registres
- **D0, D1, D2, D3, D4, D5, D6, D7** ; Data bites, les broches qui servent à transférer les informations de la carte Arduino vers l'écran. [7]

Connexion d'un LCD à L'Arduino :

- Connectez la broche GND (masse) de l'écran LCD à la masse de l'Arduino.
- Connectez la broche VCC (alimentation) de l'écran LCD à la broche 5V de l'Arduino.
- Connectez la broche VO (contraste) de l'écran LCD à la masse de l'Arduino.
- Connectez la broche RS de l'écran LCD à la broche 10 de l'Arduino.
- Connectez la broche RW de l'écran LCD à la masse de l'Arduino.
- Connectez la broche E de l'écran LCD à la broche 9 de l'Arduino.

- Connectez la broche négative (-) du buzzer à la masse (GND) de l'Arduino.

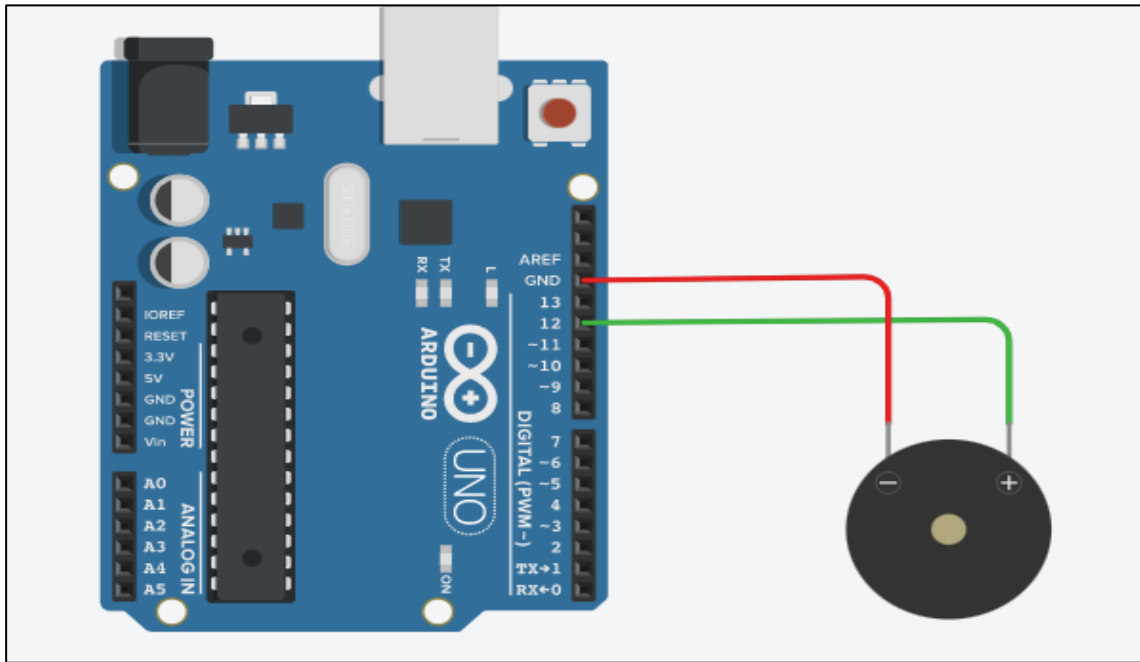


Figure II-5 Schéma de câblage pour Buzzer avec Arduino

II.5. Clavier numérique 4x4 :

Un clavier numérique de 4x4 est un type spécifique de clavier qui est composé de 16 touches disposées en 4 colonnes et 4 rangées. Chaque touche représente un chiffre de 0 à 9, ainsi que les symboles '*' et '#' ainsi que des lettres 'A', 'B', 'C', 'D'. Ce type de clavier est souvent utilisé pour les téléphones mobiles, les terminaux de paiement, les boîtiers de commande à distance et les serrures électroniques. Les touches sont disposées sous forme de matrice, permettant ainsi de gérer plusieurs touches avec un nombre minimal de connexions à un microcontrôleur. Lorsqu'un utilisateur appuie sur une touche, le clavier envoie un signal électrique au système électronique qui l'interprète et exécute la commande correspondante.[8]



Figure II-6 Clavier numérique 4X4

II.5.1. Principe de fonctionnement :

Un clavier numérique est un ensemble de boutons qui permettent à l'utilisateur de saisir des chiffres et des symboles. Dans les claviers numériques, les boutons sont disposés sous forme de matrice, ce qui permet de gérer plusieurs boutons avec moins de connexions à un microcontrôleur.

En effet, dans une matrice, les boutons d'une même colonne sont connectés à une entrée du microcontrôleur, et les boutons d'une même ligne sont connectés à une autre entrée. Lorsqu'un bouton est pressé, il ferme le circuit entre la ligne et la colonne correspondantes, permettant ainsi de déterminer quel bouton a été enfoncé. Ce système est particulièrement efficace car il permet de gérer un grand nombre de boutons avec un nombre minimal de connexions à un microcontrôleur. Par exemple, un clavier numérique de 16 boutons peut être connecté à seulement 8 entrées du microcontrôleur, ce qui permet de réduire les coûts et la complexité du circuit.

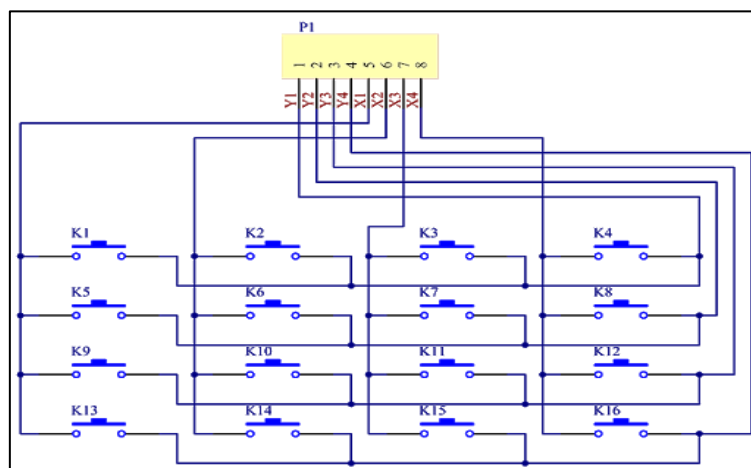


Figure II-7 Disposition des touches dans un clavier numérique 4*4

II.5.2. Connexion d'un clavier numérique à un microcontrôleur Arduino :

Instructions de câblage :

- Tout d'abord, il faut repérer les 8 broches du clavier numérique, soit 4 pour les colonnes et 4 pour les rangées.
- Connectez chaque broche de colonne à une broche numérique d'entrée de l'Arduino, et chaque broche de rangée à une broche numérique de sortie de l'Arduino.

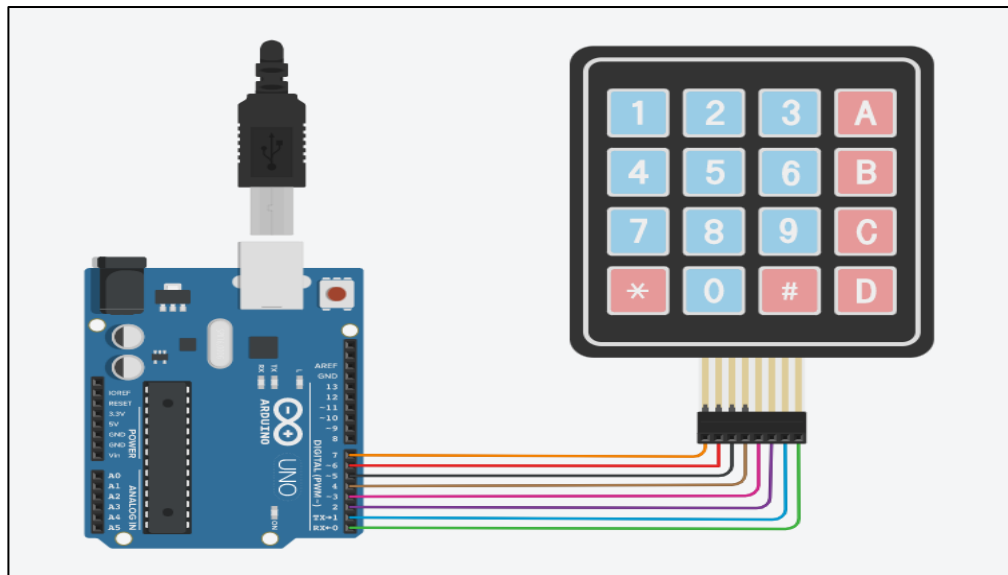


Figure II-8 Schéma de câblage pour clavier numérique avec Arduino

II.6. Plaque d'essai : (Bread Board)

Le plus grand avantage d'une plaque d'essai c'est qu'elle permet de réaliser des montages électroniques sans soudure, et elle est disponible en plusieurs taille et longueur différentes.[9]

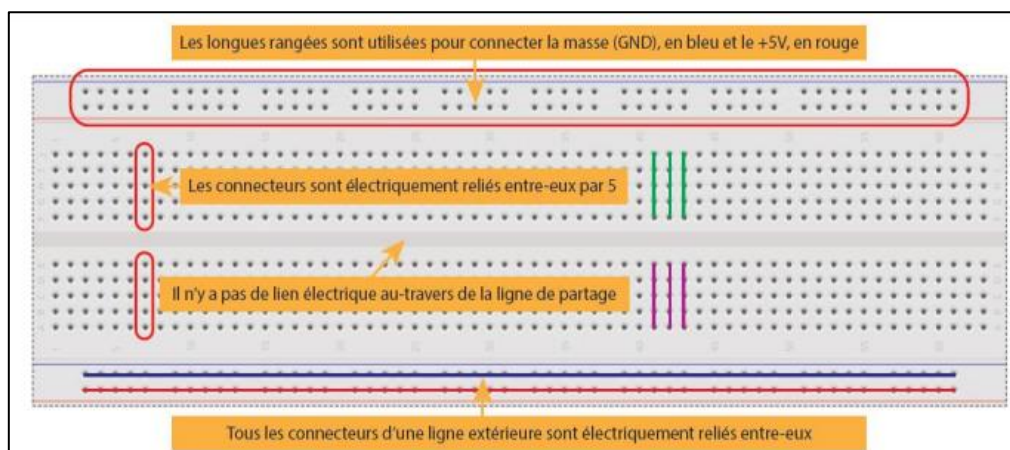


Figure II-9 Plaque d'essai

II.7. Potentiomètre rotatif :

Un potentiomètre (également connu sous le nom de potard) est un élément électronique passif autrement appelé résistance variable à trois bornes, dans laquelle la résistance est modifiée manuellement pour contrôler le flux de courant électrique. Un potentiomètre agit comme un diviseur de tension réglable.

Un potentiomètre est équipé de trois bornes. Deux d'entre elles sont connectées en permanence à un chemin de résistance, c'est-à-dire un fragment de la couche résistive. Le troisième conduit au curseur conducteur qui se déplace le long de ce chemin (avec un bouton ou un tournevis). Le curseur divise le chemin de résistance en deux résistances connectées en série. Lors de la modification du réglage du potentiomètre, le rapport de ces résistances change, mais leur somme en ohm reste constante.

Les potentiomètres fonctionnent donc en faisant varier la position d'un contact glissant sur une résistance uniforme. Dans ce type d'élément, la tension d'entrée est appliquée sur toute la longueur de la résistance et la tension de sortie est la chute de tension entre le contact fixe et le contact glissant.[9]



Figure II-10 Potentiomètre rotatif

II.8. Conclusion :

En résumé, le chapitre sur le matériel et les composants de notre projet de sécurité et d'alarme a été d'une importance capitale. En combinant des éléments tels qu'un capteur de mouvement PIR, un clavier numérique, un buzzer, un écran LCD et d'autres composants, nous avons pu construire un système de sécurité complet et efficace. Ces composants jouent un rôle essentiel dans la détection des intrusions, l'activation/désactivation sécurisée du système, l'émission d'alertes sonores et l'affichage d'informations.

CHAPITRE III

PRESENTATION DES LOGICIELS

UTILISES

III.1. L'IDE ARDUINO :

III.1.1. Introduction :

L'IDE Arduino est un environnement de développement open source pour les projets électroniques. Il est créé pour simplifier le processus de développement en offrant une interface graphique intuitive pour créer, éditer, compiler et téléverser du code sur une carte Arduino. utilise une version modifiée du compilateur GCC. L'IDE est livrée avec une bibliothèque des fonctions pré-écrites et prend en charge la programmation en C++. Il permet aux utilisateurs d'écrire, compiler et téléverser du code directement depuis l'interface, pour un processus de développement rapide et efficace.

III.1.1. Présentation de l'interface de l'IDE Arduino :

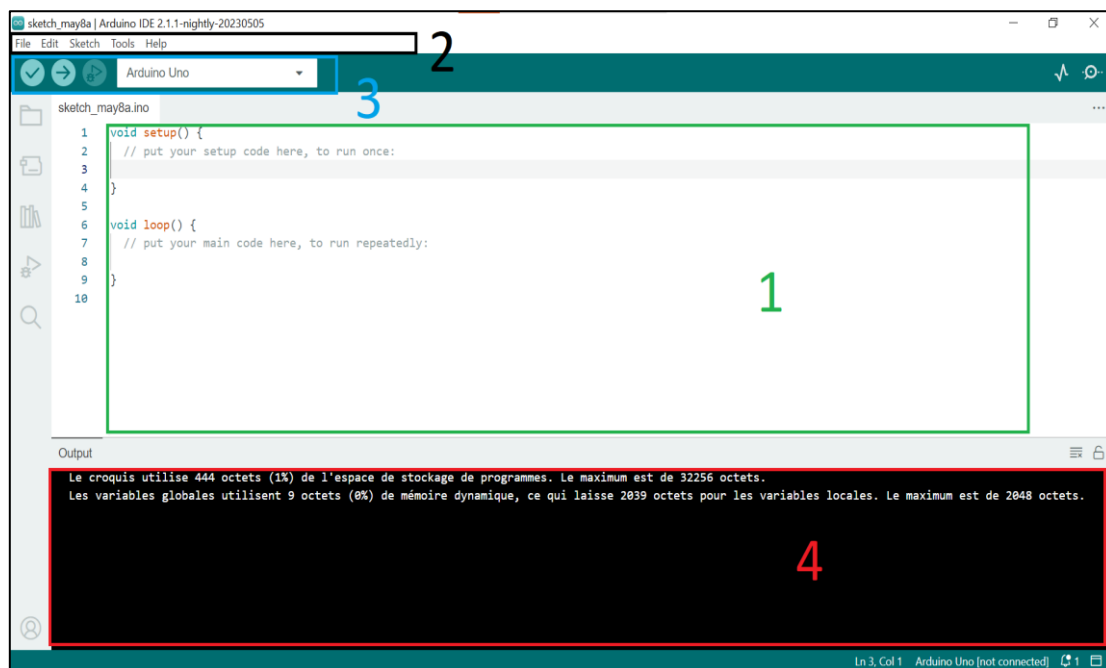


Figure III -1 Présentation de l'interface graphique de l'IDE Arduino

L'interface graphique de l'IDE Arduino est conçue pour être intuitive et facile à utiliser, afin de simplifier le processus de développement pour les débutants comme pour les professionnels. Elle est divisée en plusieurs zones :

1. La zone principale de l'interface de l'IDE Arduino est l'éditeur de code, où l'utilisateur peut écrire son code. Cette zone est équipée de fonctionnalités telles que la coloration

syntaxique, qui aident à faciliter la lecture et la compréhension du code. dans cette zone, l'utilisateur peut écrire son code pour les deux fonctions principales du programme Arduino : la fonction void setup() et la fonction void loop(). La fonction void setup() est appelée une seule fois au début de l'exécution du programme, et elle est utilisée pour initialiser les variables et les configurations matérielles nécessaires pour le fonctionnement du programme. La fonction void loop() est exécutée en boucle continue tant que l'Arduino est sous tension, et elle est utilisée pour définir les actions à exécuter par le programme.

2. La zone en haut de l'écran se trouve la barre de menu, qui permet d'accéder à toutes les fonctionnalités de l'IDE.
3. Cette zone propose des boutons permettant de compiler le programme, de sélectionner le type de carte Arduino à utiliser et de téléverser le code sur la carte. Cette zone joue un rôle important dans le processus de développement de projets électroniques avec l'IDE Arduino.
4. La zone noire située en bas de l'interface de l'IDE Arduino. Elle est principalement utilisée pour afficher les résultats de compilation. Il affiche également des informations sur les erreurs et les avertissements lors de la compilation et de l'exécution du code. Cette zone est donc très utile pour le débogage des projets électroniques développés avec l'IDE Arduino.

III.1.2. Choisir de la carte arduino :

Exemple de choisir l'Arduino Uno : Tools > Board "arduino Uno" > arduino AVR Board > Arduino Uno.

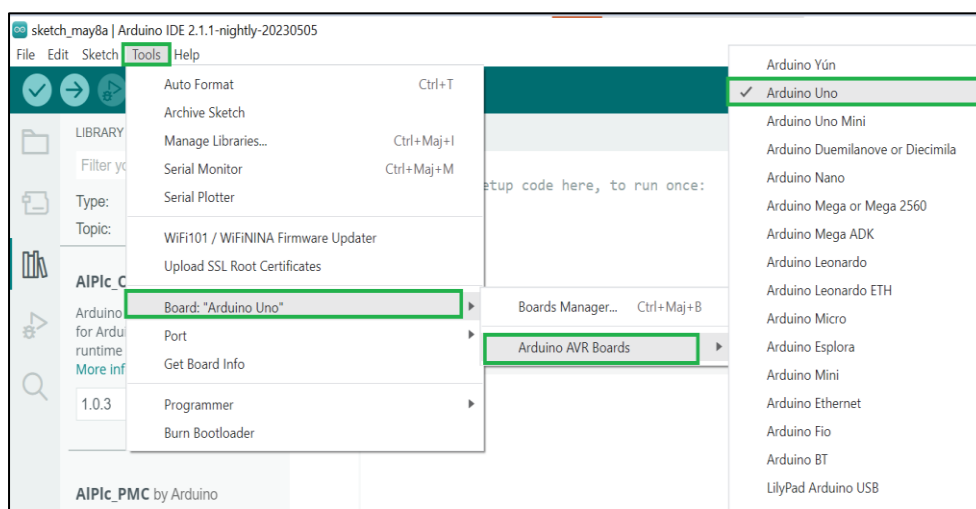


Figure III -2 Interface de sélection de la carte Arduino dans l'ID

III.1.3. Ouvrir l'exemple BLINK :

Les exemples de code fournis dans l'IDE Arduino permettent aux utilisateurs de comprendre comment fonctionnent les différentes fonctions et composants électroniques. Par exemple, voici l'exemple de code "Blink" qui permet de faire clignoter une LED connectée à une carte Arduino.

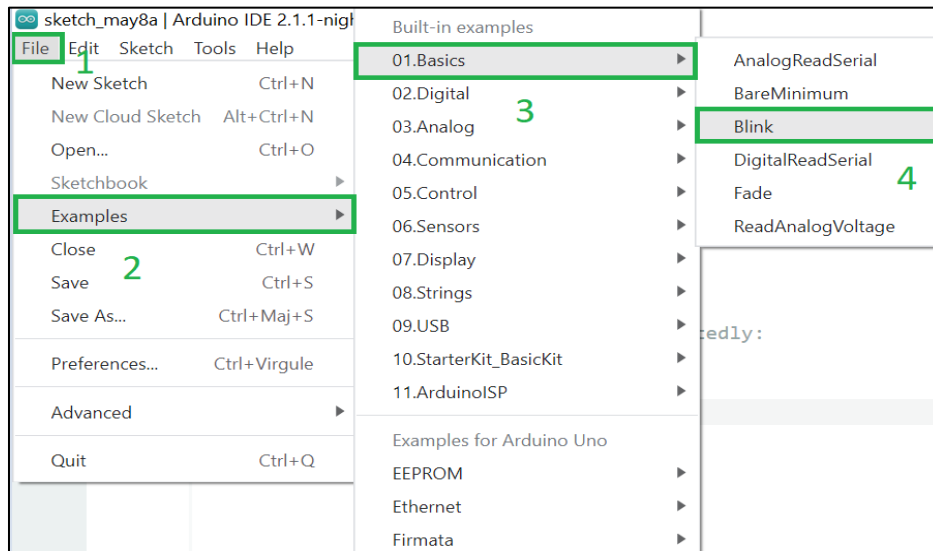


Figure III -3 Exemple de code : utilisation de la fonction Blink

III.1.4. Bibliothèques électroniques :

Pour intégrer une bibliothèque d'un élément électronique dans votre code, la bibliothèque doit tout d'abord être téléchargée et installée dans l'IDE Arduino.

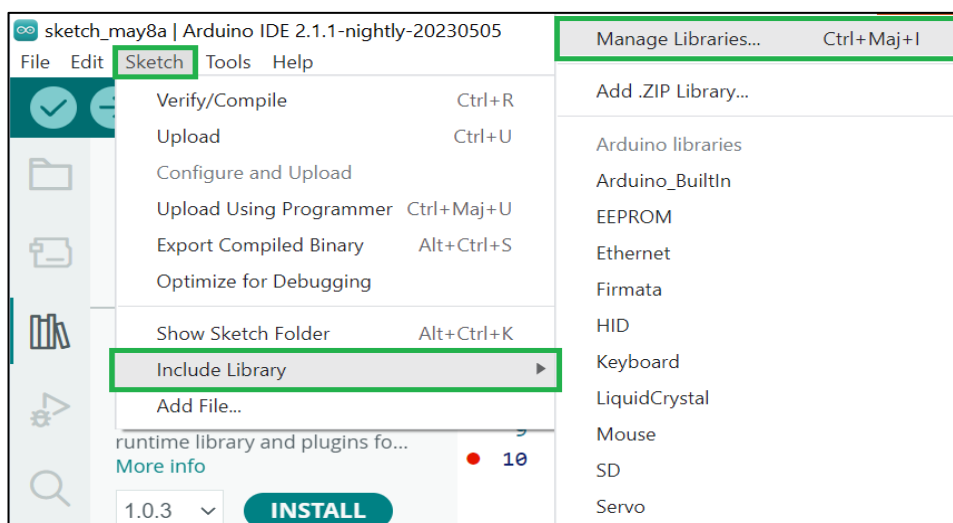


Figure III -4 Utilisation des bibliothèques dans l'IDE Arduino

III.1.5. La structure d'un programme :



Figure III -5 Exemple Blink

(1) : Le programme Arduino commence par l'inclusion des bibliothèques nécessaires, qui fournissent des fonctionnalités supplémentaires pour le projet. Ensuite, les variables globales sont déclarées, ce sont des variables accessibles à la fois dans la fonction void setup() et dans la fonction void loop().

(2) : La fonction void setup() est appelée une seule fois au démarrage du programme. Elle est utilisée pour configurer les broches d'entrée/sortie, initialiser les bibliothèques, effectuer d'autres initialisations nécessaires et exécuter des instructions qui ne doivent être réalisées qu'une seule fois.

(3) : La fonction void loop() est la boucle principale du programme. Une fois que la fonction void setup() a été exécutée, la fonction void loop() est appelée en boucle indéfiniment. Elle contient les instructions qui doivent être exécutées en continue, telles que la lecture de capteurs, le traitement des données, les actions à réaliser en fonction des conditions, etc.

III.2. Présentation du logiciel PROTEUS :

III.2.1. Introduction :

Proteus ISIS est un logiciel de conception électronique assistée par ordinateur (CAO) développé par Labcenter Electronics. Il permet aux ingénieurs et aux étudiants en électronique de concevoir, simuler et tester des circuits électroniques avant de les construire dans le monde réel.

Proteus ISIS est une suite logicielle complète qui inclut des outils pour la capture de schémas, la simulation de circuits, la conception de circuits imprimés (PCB), et la génération de fichiers de fabrication de PCB. Il a été créé pour la première fois en 1988 et est devenu l'un des logiciels les plus populaires dans le domaine de la conception électronique. [10]

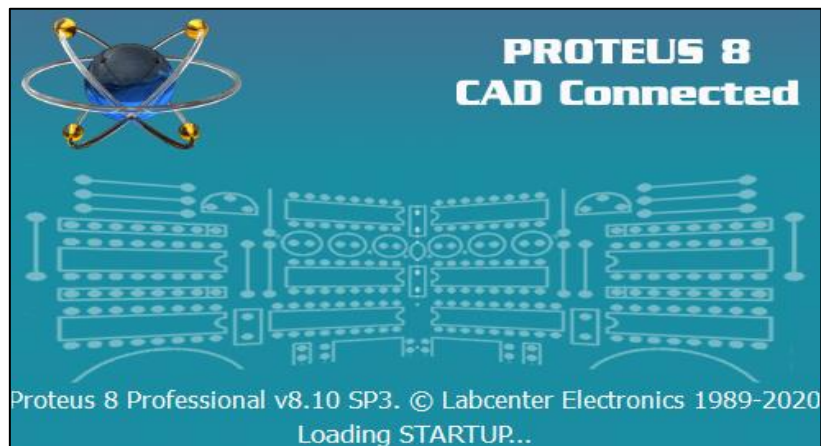


Figure III -6 Proteus 8

III.2.2. Présentation de l'interface Proteus 8 :

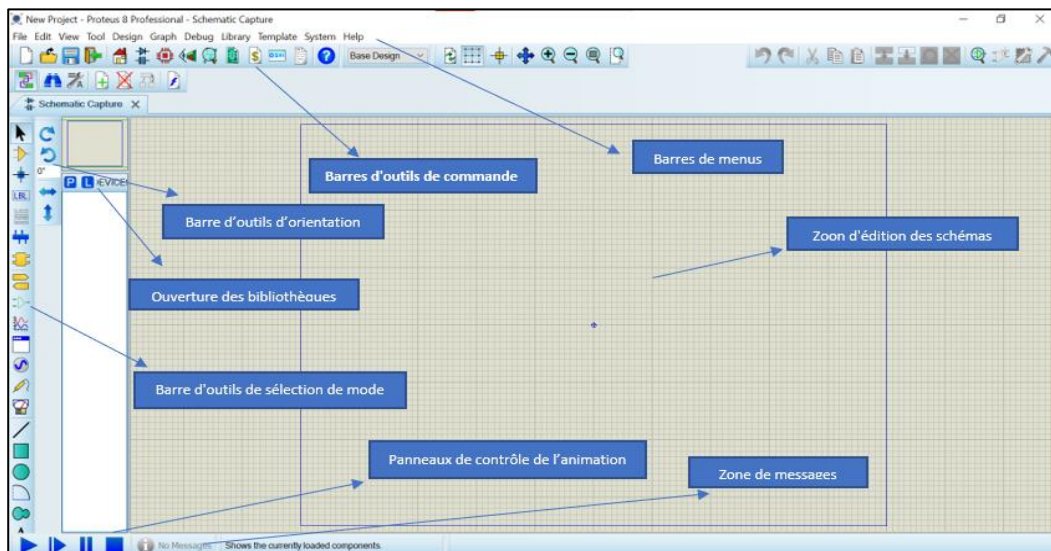


Figure III -7 Interface de Proteus 8

- **Barre de menu :** cette barre permet de gérer le travail (ouverture, sauvegarder, mode d'affichage, etc.)

- **Barres d’outils de commande** : ces barres fournissent un accès équivalent aux commandes des menus.
- **Barre d’outils de sélection de mode** : cette barre permet de sélectionner un outil parmi 3 modes d’Edition disponibles : mode principale, mode gadgets et mode graphique.
- **Barre d’outils d’orientation** : cette barre permet d’afficher et de contrôler la rotation et la réflexion d’un objet.
- **Zone d’Edition des schémas** : tous les schémas apparaitront dans cette zone et seront visualisés avec le coefficient d’agrandissement choisi.

III.2.3. Création d'un nouveau projet :

Comme le montrent les figures suivantes, pour créer un nouveau projet dans Proteus ISIS, il suffit de sélectionner l'option "Nouveau Projet" dans le menu "Fichier", puis de spécifier le nom du projet et le type de projet souhaité.

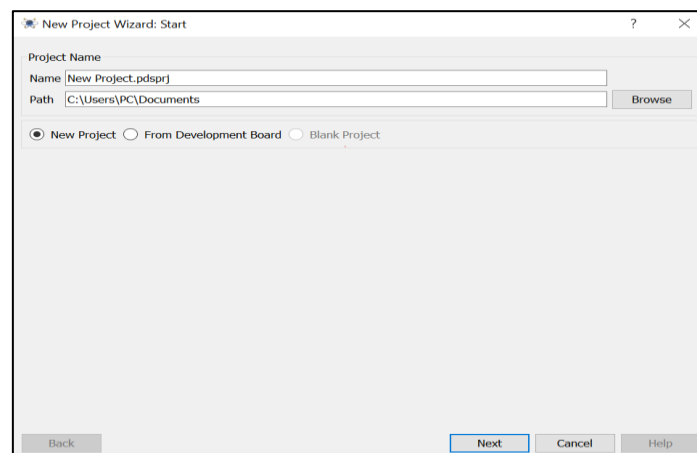


Figure III -8 Créer un nouveau projet sur ISIS

III.3. Conclusion :

En résumé, la maîtrise de l'IDE Arduino et de Proteus ISIS constitue un élément essentiel pour le succès de nos projets électroniques. Ces logiciels offrent les fonctionnalités nécessaires pour la programmation, la simulation et la validation de nos circuits. Ils nous permettent de développer nos projets avec précision et efficacité, en minimisant les erreurs .

CHAPITRE IV

SIMULATION ET REALISATION DU

SYSTEME DE SECURITE ET D'ALARME

IV.1. Introduction :

Dans ce chapitre, nous allons explorer la simulation d'un système de sécurité utilisant Arduino, en combinant différents composants tels que le Keypad (clavier), le Buzzer et l'afficheur LCD. Cette simulation nous permettra de comprendre comment ces composants interagissent et travaillent ensemble pour créer un système de sécurité efficace.

IV.2. Description de système :

Notre système de sécurité est principalement composé d'une carte Arduino Uno, qui agit comme le cœur du système, ainsi que de plusieurs autres composants clés. Nous utilisons un capteur de mouvement pour détecter tout mouvement dans la zone surveillée.

Une fois un mouvement détecté, le système déclenche l'alarme à l'aide d'un buzzer, pour alerter les personnes à proximité de la présence d'un mouvement.

Pour désactiver l'alarme, un code d'accès est requis. Un clavier de type Keypad est utilisé pour entrer ce code, tandis qu'un écran LCD affiche les informations nécessaires pour guider l'utilisateur. Une fois le code correctement saisi, le système désactive l'alarme et rétablit la sécurité.

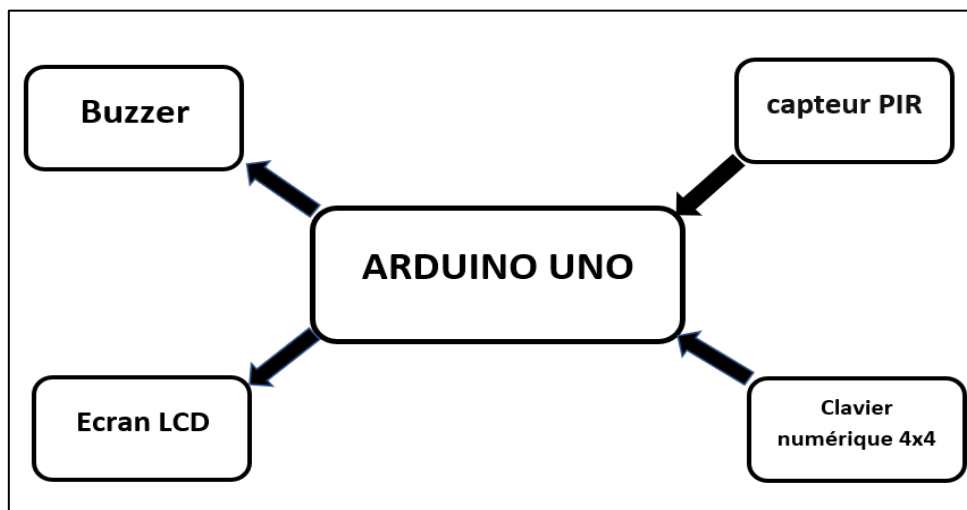


Figure IV-1 Schéma du système de sécurité et d'alarme

IV.3. Organigramme du système de sécurité Arduino :

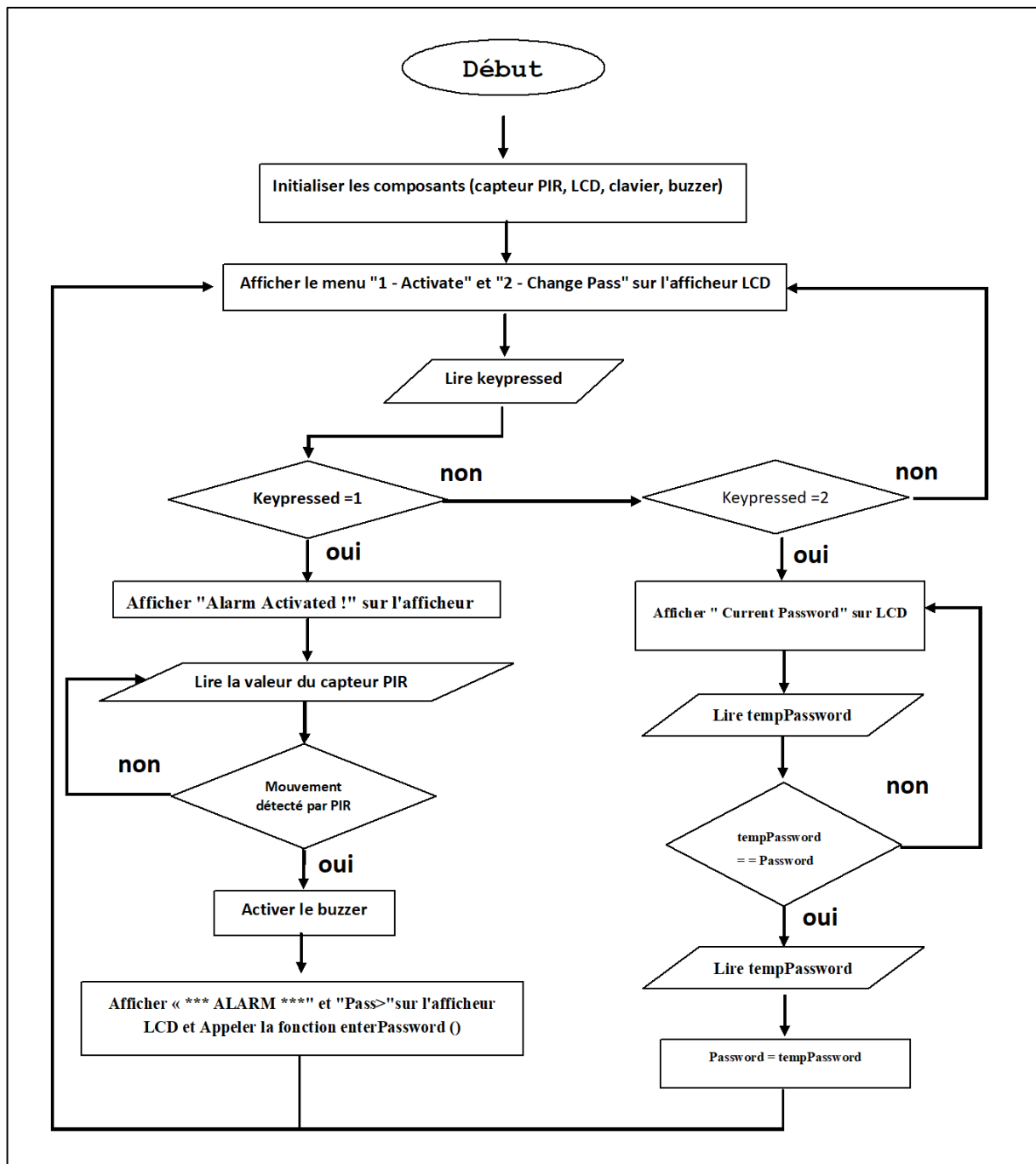


Figure IV-2 Organigramme de fonctionnement général de notre système

IV.4. Simulation du système d’alarme :

Dans le cadre de notre projet du système de sécurité basé sur Arduino, nous avons décidé d'utiliser la simulation sur « Proteus 8 » pour tester et évaluer notre solution avant de passer à

la phase pratique. Cette approche nous a permis de simuler le fonctionnement de notre système et de vérifier son comportement dans différents scénarios.

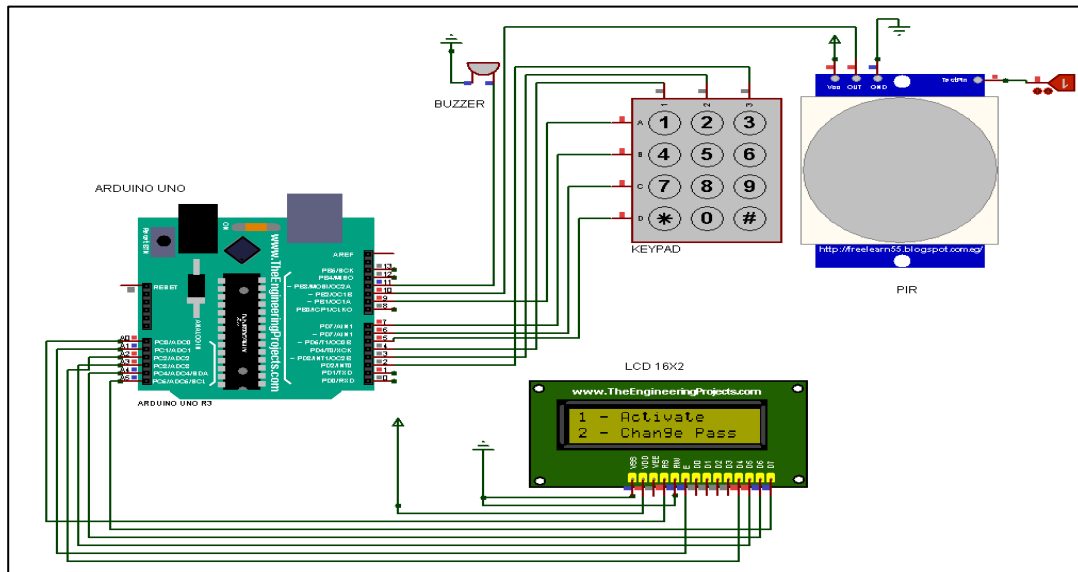


Figure IV-3 Schéma du circuit électrique du système sur Proteus

Dans la boucle principale de notre programme, les fonctionnalités clés de notre système de sécurité ont été gérées. Si l'activation de l'alarme était demandée, un compte à rebours était affiché sur l'écran LCD et l'alarme était activée lorsque le délai était écoulé.

Lorsque l'alarme était activée, l'état du capteur PIR était vérifié pour détecter les mouvements. Si un mouvement était détecté, l'alarme était déclenchée en activant le buzzer et en affichant un message d'alerte sur l'écran LCD.

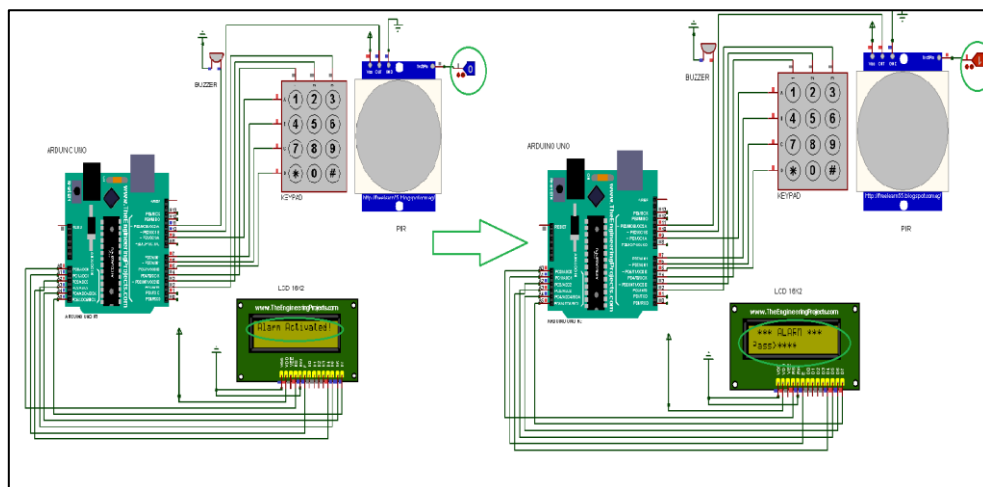


Figure IV-4 la communication entre le capteur PIR, l'écran LCD et Arduino dans la simulation

IV.5. La réalisation finale de projet :

Dans la première partie, nous souhaitons présenter une photographie mettant en évidence l'ensemble du matériel que nous avons utilisé. Ce dernier a été décrit en détail précédemment.



Figure IV-5 Vue d'ensemble du matériel utilisé

IV.5.1. Téléversement du code vers Arduino :

Les figures ci-dessous présente les étapes à suivre pour téléverser avec succès le code vers l'Arduino :

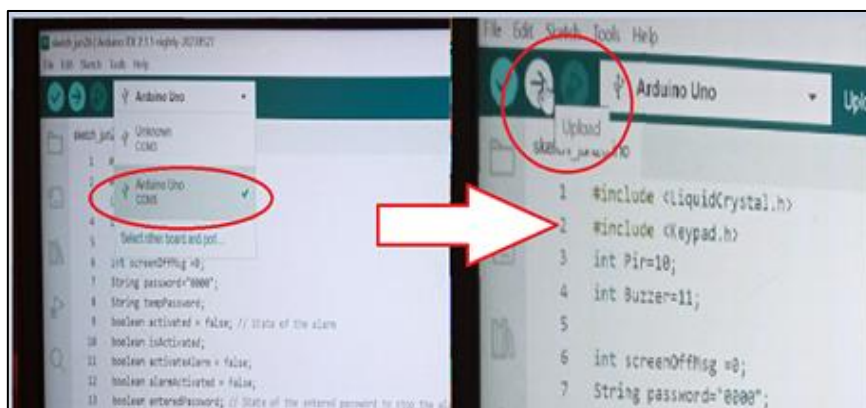


Figure IV-6 Procédure de téléversement du code vers l'Arduino

IV.5.2. Mise en marche du système :

Dans les figures suivantes, nous présentons les étapes détaillées pour la mise en marche du système de sécurité Arduino .

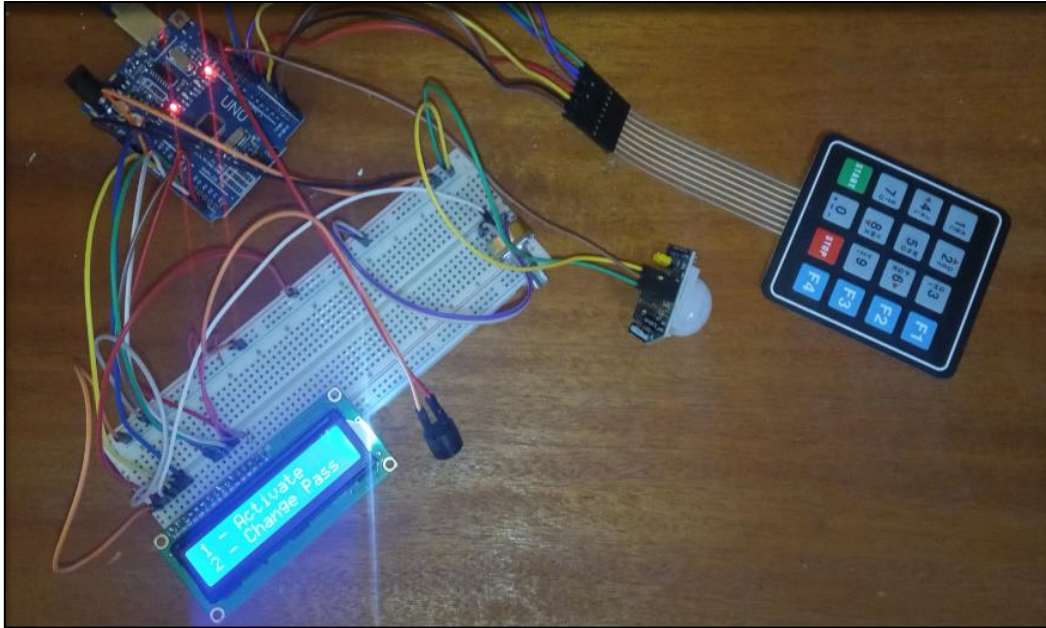


Figure IV-7 Réalisation pratique du système de sécurité et d'alarme

IV.5.2.1. Activation du système de sécurité et d'alarme :

Dans cette partie, nous détaillerons le processus d'activation du système de sécurité après un compte à rebours de 3 secondes. Une fois que l'utilisateur aura entré "1" sur le clavier numérique, le système sera activé après un compte à rebours de 3 secondes. L'écran LCD affichera ensuite l'état actuel du système. Parallèlement, le capteur PIR surveillera les mouvements, et si un mouvement est détecté, l'alarme se déclenchera à l'aide du buzzer. Les figures ci-dessous illustrent les étapes nécessaires :

- Etape 1 : Déclenchement du compte à rebours après avoir sélectionné "1"

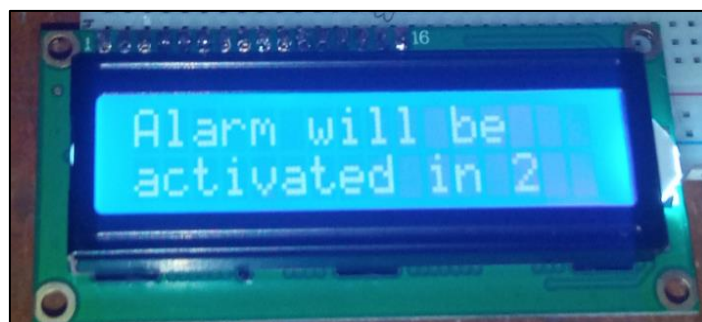


Figure IV-8 Compte à rebours avant l'activation du système

- Etape 2 : Activation du système et affichage sur l'écran LCD "Alarme activée !"

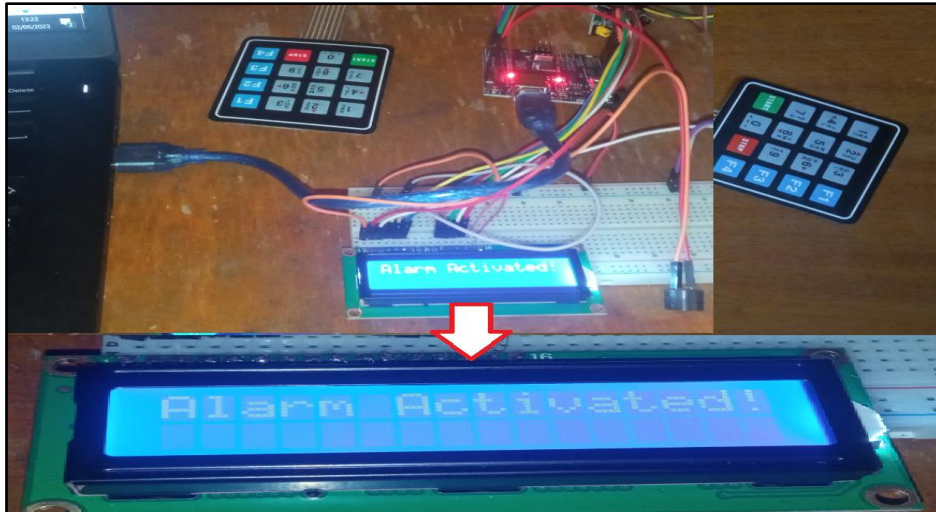


Figure IV-9 Affichage de l'état du système sur l'écran LCD après l'activation

- Etape 3 : lorsque le capteur PIR détecte un mouvement, l'alarme se déclenche. Simultanément, l'écran LCD affiche un message demandant à l'utilisateur d'entrer le mot de passe pour désactiver l'alarme.

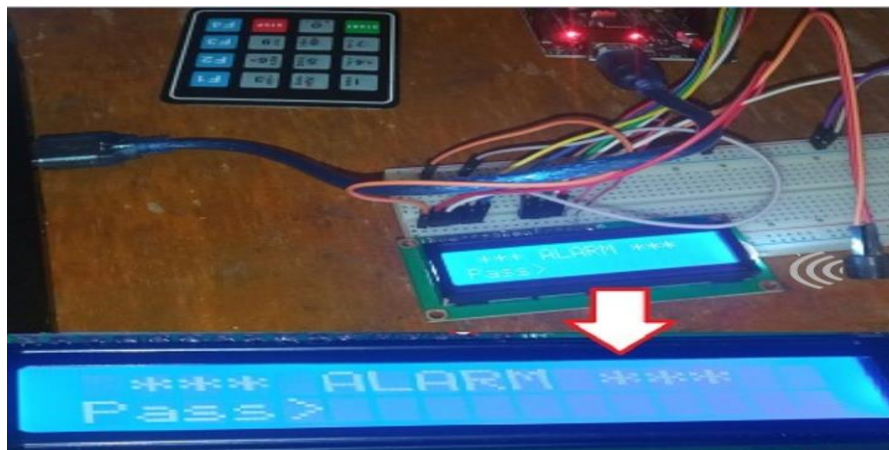


Figure IV-10 Déclenchement de l'alarme

IV.5.2.2. Changement du mot de passe pour le système de sécurité :

L'utilisateur doit choisir l'option correspondante dans le menu du système, qui sera affichée sur l'écran LCD. Une fois cette option sélectionnée, l'écran LCD demandera à l'utilisateur d'entrer le dernier mot de passe utilisé, qui est généralement initialisé à "0000". Après avoir entré le dernier mot de passe, l'utilisateur sera invité à entrer le nouveau mot de passe qu'il souhaite définir. Cette procédure permet à l'utilisateur de modifier le mot de passe du système selon ses besoins, renforçant ainsi la sécurité globale du système de sécurité.

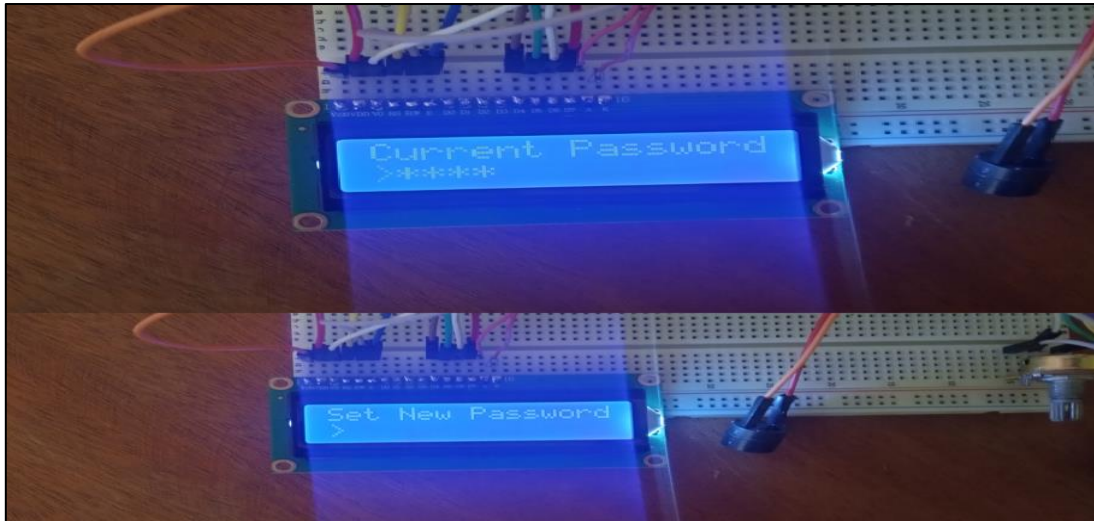


Figure IV-11 Changement du mot de passe sur l'écran LCD du système de sécurité

IV.6. Conclusion :

En résumé, grâce à la simulation et à la mise en œuvre du projet du système de sécurité et d'alarme, nous avons pu créer un système fonctionnel et fiable. Les composants PIR, buzzer, LCD, clavier et carte Arduino Uno fonctionnent ensemble pour fournir une protection optimale et une interface conviviale.

Conclusion générale

Ce travail de projet de fin d'étude vise à réaliser un système de sécurité et d'alarme efficace et flexible afin de prévenir les risques de cambriolage en utilisant la technologie Arduino comme cœur de ce système. il est important de souligner que ce système ne se limite pas seulement à la protection matérielle des habitants de la zone surveillée, mais il a également un impact significatif sur leur tranquillité d'esprit et leur sentiment de sécurité.

Dans ce travail, nous avons développé un système de sécurité et d'alarme en utilisant des composants appropriés. Nous avons également utilisé des logiciels pour la simulation avant de passer à la phase de réalisation. Nous avons utilisé un capteur PIR pour détecter les mouvements indésirables, un buzzer pour alerter les utilisateurs des zones surveillées, un clavier numérique pour faciliter la communication entre le système et l'utilisateur, et enfin un écran LCD pour afficher et guider l'utilisateur.

Pour la phase de simulation, nous avons utilisé un logiciel pour simuler notre système dans différents scénarios. À la fin de ce travail, nous avons réussi à mettre en marche notre système dans le monde réel, en tenant compte des contraintes financières et d'efficacité.

Ce projet de fin d'étude est composé de **quatre chapitres**. Dans notre **premier chapitre**, nous avons effectué une introduction générale sur les cartes Arduino, en fournissant un aperçu historique et en décrivant les différents types d'Arduino disponibles. Nous avons ensuite focalisé notre attention sur la carte Arduino Uno, qui a été choisie pour notre projet.

Dans le **deuxième chapitre**, nous avons présenté en détail les composants du système de sécurité que nous avons réalisé. Nous avons décrit le fonctionnement de chaque composant.

Dans le **troisième chapitre**, nous avons exploré l'environnement de développement Arduino IDE et le logiciel de simulation ISIS pour concevoir et simuler notre système de sécurité.

Enfin, dans le **quatrième chapitre**, nous avons réalisé avec succès une simulation de notre système de sécurité, démontrant son bon fonctionnement et son efficacité dans la détection des mouvements et la génération d'alertes sonores appropriées. Nous avons également effectué la mise en œuvre pratique du système, en connectant les composants physiques et en testant leur fonctionnement réel.

Bibliographie

- [1]. BENYOUB TAHAR, MADI AZZEDDINE. *"Testeur des circuits logiques à base d'Arduino."* (2020).
- [2]. Retrieved April 26, 2023, from <https://store.arduino.cc/>
- [3]. Retrieved May 4, <http://www.wsis-community.org/pg/groups/14358/open-educational-resources-oer>
- [4]. ASTALESENVEN, ESKIMON, OLYTE. *"Arduino pour bien commencer en électronique et en programmation"*.
- [5]. Retrieved May 30, from <https://www.biomaker.org/bloccatalogue/2021/12/17/infrared-pyroelectric-motion-sensor-hc-sr501>
- [6]. EL YAHIAOUI K, BOUKOUTAYA A, *"Réalisation d'une maison intelligente à base d'Arduino"* ., UNIVERSITE MOHAMED V
- [7]. Retrieved May 3, 2023, from <https://docs.arduino.cc/learn/electronics/lcd-displays>
- [8]. ZEMIR, Z., KHELIL CHERFI, T. and RAGGAI, D., 2019. *Etude et réalisation d'une maison intelligente.*
- [9]. El AMRI, K ., IRAOUI, A. 2020" *Commande d'injection d'insuline autour d'Arduino."* ., Université Chouaib Doukkali .
- [10]. Retrieved May 9, 2023, from <https://www.labcenter.com/>

Annexe

Script complet du projet de système de sécurité Arduino :

```
#include <LiquidCrystal.h>
#include <Keypad.h>
int Pir=10;
int Buzzer=11;
int screenOffMsg =0;
String password="0000";
String tempPassword;
boolean activated = false; // State of the alarm
boolean isActivated;
boolean activateAlarm = false;
boolean alarmActivated = false;
boolean enteredPassword; // State of the entered password to stop the alarm
boolean passChangeMode = false;
boolean passChanged = false;

const byte ROWS = 4; //four rows
const byte COLS = 4; //four columns
char keypressed;
//define the symbols on the buttons of the keypads
char keyMap[ROWS][COLS] = {
  {'1','2','3','F1'},
  {'4','5','6','F2'},
  {'7','8','9','F3'},
  {'*','0','#','F3'}
};
byte rowPins[ROWS] = { 9, 8, 7, 6}; //Row pinouts of the keypad
byte colPins[COLS] = { 5, 4, 3,2};
Keypad myKeypad = Keypad( makeKeymap(keyMap), rowPins, colPins, ROWS,
COLS);
LiquidCrystal lcd(A0,A1,A2,A3,A4,A5);

void setup(){
  Serial.begin(9600);
  pinMode(Pir,INPUT);
  pinMode(Buzzer,OUTPUT);
  lcd.begin(16,2);
}

void loop() {
  if (activateAlarm) {

    lcd.clear();
    lcd.setCursor(0,0);
    lcd.print("Alarm will be");
    lcd.setCursor(0,1);
```

```

    lcd.print("activated in");

    int countdown = 3; // 3 seconds count down before activating the alarm
    while (countdown != 0) {
        lcd.setCursor(13,1);
        lcd.print(countdown);
        countdown--;
        delay(500);
    }
    lcd.clear();
    lcd.setCursor(0,0);
    lcd.print("Alarm Activated!");
    activateAlarm = false;
    alarmActivated = true;
}
if (alarmActivated == true){
    if ( digitalRead(Pir)==HIGH) {
        tone(Buzzer,2000); // Send 1KHz sound signal###
        lcd.clear();

        enterPassword();
    }
}
if (!alarmActivated) {
    if (screenOffMsg == 0 ){
        lcd.clear();
        lcd.setCursor(0,0);
        lcd.print("1 - Activate");
        lcd.setCursor(0,1);
        lcd.print("2 - Change Pass");
        screenOffMsg = 1;
    }
    keypressed = myKeypad.getKey();
    if (keypressed == '1'){ //If A is pressed, activate the alarm
        tone(Buzzer, 100, 500);
        activateAlarm = true;
    }
    else if (keypressed == '2') {
        lcd.clear();
        int i=1;
        tone(Buzzer, 2000, 100);
        tempPassword = "";
        lcd.setCursor(0,0);
        lcd.print("Current Password");
        lcd.setCursor(0,1);
        lcd.print(">");
        passChangeMode = true;
        passChanged = true;
        while(passChanged) {

```

```

    keypressed = myKeypad.getKey();
    if (keypressed != NO_KEY){
        if (keypressed == '0' || keypressed == '1' || keypressed == '2' ||
keypressed == '3' ||
            keypressed == '4' || keypressed == '5' || keypressed == '6' ||
keypressed == '7' ||
            keypressed == '8' || keypressed == '9' ) {
            tempPassword += keypressed;
            lcd.setCursor(i,1);
            lcd.print("*");
            i++;
            tone(Buzzer, 2000, 100);
        }
    }
    if (i > 5 || keypressed == '#') {
        tempPassword = "";
        i=1;
        lcd.clear();
        lcd.setCursor(0,0);
        lcd.print("Current Password");
        lcd.setCursor(0,1);
        lcd.print(">");
    }
    if ( keypressed == '*') {
        i=1;
        tone(Buzzer, 2000, 100);
        if (password == tempPassword) {
            tempPassword="";
            lcd.clear();
            lcd.setCursor(0,0);
            lcd.print("Set New Password");
            lcd.setCursor(0,1);
            lcd.print(">");
            while(passChangeMode) {
                keypressed = myKeypad.getKey();
                if (keypressed != NO_KEY){
                    if (keypressed == '0' || keypressed == '1' || keypressed ==
'2' || keypressed == '3' ||
                        keypressed == '4' || keypressed == '5' || keypressed ==
'6' || keypressed == '7' ||
                            keypressed == '8' || keypressed == '9' ) {
                        tempPassword += keypressed;
                        lcd.setCursor(i,1);
                        lcd.print("*");
                        i++;
                        tone(Buzzer, 2000, 100);
                    }
                }
            }
            if (i > 5 || keypressed == '#') {

```



```

k=5;
lcd.clear();
lcd.setCursor(0,0);
lcd.print(" *** ALARM *** ");
lcd.setCursor(0,1);
lcd.print("Pass>");
}
if ( keypressed == '*' ) {
  if ( tempPassword == password ) {
    activated = false;
    alarmActivated = false;
    noTone(Buzzer);
    screenOffMsg = 0;
  }
  else if (tempPassword != password) {
    lcd.setCursor(0,1);
    lcd.print("Wrong! Try Again");
    delay(100);
    lcd.clear();
    lcd.setCursor(0,0);
    lcd.print(" *** ALARM *** ");
    lcd.setCursor(0,1);
    lcd.print("Pass>");
  }
}
}
}
}

```