

REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE
MINISTRE DE L'ENSEIGNEMENT SUPERIEUR ET DE LA RECHERCHE
SCIENTIFIQUE

Université de Mohamed El-Bachir El-Ibrahimi - Bordj Bou Arreridj

Faculté des Sciences et de la technologie

Département d'Electronique

Rapport

Projet de Fin de Cycle (PFC)

MCIL 3

FILIERE : Électronique

Spécialité : Industries Électroniques

Par

- **NACER Zakaria**
- **TRIA Abdallah**
- **IGUELOUZENE Narimane**

Intitulé

***Étude et Réalisation d'une Carte de Commande de Contrôle d'Accès à Base
d'Arduino***

Présenté le :

Devant le Jury composé de :

<i>Nom & Prénom</i>	<i>Grade</i>	<i>Qualité</i>	<i>Etablissement</i>
Mme A. LAOUAMRI	MAA	Président	Univ-BBA
Mr A. BOUSSAHOUL	MAA	Examineur	Univ-BBA
Mme N. BIOUD	MCA	Encadreur	Univ-BBA
Mr L. BOUTAHAR	DOCTORANT	Coencadreur	Univ-BBA

Remerciements

Nos remerciements Avant tout, louange à «ALLAH»

qui nous a donnés la force et la patience de finaliser ce modeste Travail.

Nous remercions considérablement notre encadrant Mme N. BIOUD et notre coencadreur Mr L. BOUJAHAR qui nous ont accueilli chaleureusement et ils ont accepté de diriger ce mémoire, et qui étaient toujours disponibles, Un grand merci pour leurs encouragements et leurs conseils judicieux de très grande utilité.

Nous remercions les membres de jury, chacun par Son nom, qui nous ont honorés en acceptant d'examiner notre travail.

Nous tenons à remercier sincèrement tous les enseignants Du département d'électronique de l'université de Mohamed El-Bachir El-Ibrahimi - Bordj Bou Arreridj.

Nous adressons nos sincères remerciements à tous ceux qui ont contribué, de près ou de loin, à la réalisation de ce travail.

Enfin, nous remercions tous nos amis

Dédicace

*Nous dédions ce grand travail à ceux qui nous ont donné
la vie, nos chers parents qui se sont sacrifiés jour et nuit
pour notre bonheur et notre réussite, et ont fait de sorte
que rien ne nous manque. Que dieu les garde et les protège.*

*A nos frères et sœurs, camarades, et amis, à qui nous
souhaitons plein de bonheur et de réussite dans leurs vies.*

A tous ceux qui nous sont chers.

A tous ceux que nous connaissons de près ou de loin.

A tous ceux que nous aimons.

Et à tous ceux qui nous aiment.

Nous vous dédions ce travail.

Zakaria, Abdallah et Narimane

Liste des figures

Figure 1.1: Serrure à goupilles.....	2
Figure 1.2: lecteur RFID avec ses accessoires.....	4
Figure 1.3: Serrure à empreinte digitale.....	5
Figure 1.4: Tourniquets.....	5
Figure 1.5: les composants de l'Arduino.....	7
Figure 1.6: la carte de commande ArduinoUno avec ses connecteurs.....	8
Figure 1.7: Image récapitulative des composants carte Arduino.....	10
Figure 2.1: Organigramme de fonctionnement.....	16
Figure 2.2: Schéma synoptique du système.....	17
Figure 2.3: Câblage du clavier.....	18
Figure 2.4: Câblage d'un afficheur LCD +I2C.....	19
Figure 2.5: Câblage du servomoteur	20
Figure 2.6: Logo du logiciel IDE Arduino.....	21
Figure 2.7: Éditeur de texte de l'IDE Arduino.....	22
Figure 2.8: Schéma général réel.....	28
Figure 2.9: Schéma général (tinkercad).....	29
Figure 2.10: Schéma représentatif du câblage entre les différents organes.....	29
Figure 2.11: Prototype de la réalisation.....	30

Sommaire

Introduction générale.....	1
Chapitre 1 : contrôle d'accès et carte de commande	
1.1 Introduction.....	2
1.2 Systèmes de contrôle d'accès.....	2
1.2.1 Définition	2
1.2.2 Objectifs du contrôle d'accès	2
1.2.3 Types de contrôle d'accès.....	3
1.2.3.1 Contrôle d'accès manuel.....	3
1.2.3.2 Contrôle d'accès mécanique.....	3
1.2.3.3 Contrôle d'accès électronique.....	4
1.2.3.4 Contrôle d'accès mécatronique.....	4
1.2.3.5 Contrôle d'accès physique.....	5
1.2.4 Fonctionnement du système d'accès.....	6
1.3 Carte de commande (Arduino).....	7
1.3.1 Définition.....	7
1.3.2 Types de carte de commande Arduino	8
1.3.2.1 ArduinoUno (R3).....	8
1.3.2.2 Arduino Nano.....	9
1.3.2.3 Arduino Micro.....	9
1.3.3 Composants de la carte de commande Arduino	9
1.3.3.1 Microcontrôleur.....	10
1.3.3.2 port USB.....	11
1.3.3.3 Contrôle USB.....	11
1.3.3.4 Connecteur d'alimentation	11
1.3.3.5 Variateur.....	11
1.3.3.6 Cristal.....	11

1.3.3.7 Réception (RX) et transmission (TX) série.....	12
1.3.3.8 Broches numériques.....	12
1.3.3.9 Broches analogiques.....	12
1.3.3.10 Broches d'alimentation du capteur.....	12
1.3.3.11 Broches PWM.....	13
1.3.3.12 Broche AREF.....	13
1.3.3.13 Broches GND.....	13
1.3.3.14 Bouton de réinitialisation.....	13
1.4 Conclusion.....	14

Chapitre 2: simulation et réalisation pratique

2.1 Introduction.....	15
2.2 Principe de notre réalisation.....	15
2.3 Réalisation du montage.....	15
2.3.1 Organigramme de fonctionnement.....	16
2.3.2 Schéma synoptique du système.....	17
2.4 Câblage du montage.....	18
2.4.1 Câblage du clavier.....	18
2.4.2 Câblage d'un afficheur LCD +I2C.....	19
2.4.3 Câblage du servomoteur.....	20
2.5 Logiciel (IDE Arduino).....	21
2.5.1 Code source et schéma du montage.....	23
2.5.1.1 Code source et langage.....	23
2.5.1.2 Le schéma général du montage.....	28
2.6 Prototype de notre projet.....	30
2.7 Conclusion.....	30
Conclusion générale.....	31

Introduction générale

Le contrôle d'accès est un dispositif qui permet de contrôler à distance les accès d'un bâtiment. Il s'effectue généralement grâce à un passe, une carte magnétique, un badge, un code à plusieurs chiffres, un signal électrique.

Notre projet consiste à la réalisation d'un système de contrôle d'accès commandé par une carte arduino Uno. Ce système est piloté par un clavier, ce dernier est pratique car il permet de d'introduire facilement la combinaison secrète, le système digicode limite l'accès aux seules personnes en possession du code d'accès.

Ce mémoire est réparti en deux chapitres, à travers lesquels nous décrivons le travail effectué pour la conception et la réalisation de notre dispositif.

Le premier chapitre présente les différentes techniques de contrôle d'accès ainsi qu'une étude théorique descriptive sur la carte de commande de notre système qui représente l'élément de base de notre réalisation

Le deuxième chapitre sera dédié à la simulation et à la réalisation pratique de notre dispositif. Enfin, nous terminerons avec une conclusion générale et perspective.

Chapitre 1

Contrôle d'Accès

et

Carte de Commande

1.1 Introduction

Dans le présent chapitre nous allons faire une présentation générale des différents types de systèmes de contrôle d'accès, ensuite on entamera une étude descriptive sur la carte arduino Uno en se basant sur ses différentes composantes, ses broches et son alimentation.

1.2 Systèmes de contrôle d'accès

1.2.1 Définition

Un système de contrôle d'accès physique signifie généralement un système de sécurité électronique qui servira généralement des connaissances qui prennent la forme d'un badge d'accès pour permettre aux gens d'entrer dans certaines zones puis qu'ils sont en mesure de s'inscrire d'où et quand proviennent les revenus.

Pour quoi utiliser un système de contrôle d'accès au lieu de commutateurs:

Les clés physiques sont le moyen le plus simple de contrôler l'accès à la porte et le moyen commun et simple pour de nombre uses organisations, cependant, ces petites entreprises présentent de nombreuses lacunes et limitations lors de l'utilisation de clés mécaniques, surtout si le travail se développe.



Figure 1.1: Serrure à goupilles

1.2.2 Objectifs du contrôle d'accès

Le contrôle d'accès permet de savoir qui, ou quand, une personne est autorisée ou refusée d'accès

Le but de cette autorisation d'accès est de protéger les personnes, les biens ou les informations. Cela peut concerner.

Qui se soucie du contrôle d'accès

- Les employés de l'entreprise ou uniquement les visiteurs et les fournisseurs.
- Certains employés pour des lieux sensibles (bureaux, études, salles informatiques,...)
- Toutes les heures ou à certains heures du jour ou de la nuit
- Personnes, véhicules et marchandises.[1]

1.2.3 Types de contrôle d'accès

Il ya 5 différents types du contrôle d'accès:

1.2.3.1 Contrôle d'accès manuel

Pour le contrôle d'accès manuel, les personnes sont responsables d'autoriser ou de refuser l'entrée à quiconque, en fonction de critères prédéfinis. Par exemple, quelqu'un présente son billet avant de monter dans l'avion. Cette méthode est très courante dans les endroits bondés tels que les cinémas ou les espaces. En fait, cela est dû à la difficulté d'obtenir des informations préalables pour chaque individu car il n'est pas nécessaire d'obtenir l'identité de chaque individu.

1.2.3.2 Contrôle d'accès mécanique

Par exemple, dans chaque bâtiment, il y a un point d'accès ouvert ou accessible par une clé spéciale qui n'ouvre qu'une serrure spéciale. Ce type de contrôle d'accès est le plus facile à utiliser.

1.2.3.3 Contrôle d'accès électronique

Pour sécuriser les points d'accès. Le lecteur doit être muni d'une carte, d'une puce ou d'autres badges avec les informations d'identification correctes afin qu'une personne puisse les transférer. Il enregistre également les données d'entrée des personnes de temps et de lieu



Figure 1.2: lecteur RFID avec ses accessoires

1.2.3.4 Contrôle d'accès mécatronique

C'est une combinaison de systèmes électroniques et mécaniques pour améliorer la sécurité. Le système électronique est d'abord vérifié en vérifiant la carte/le code / les autres supports utilisés. Ce n'est qu'après avoir réussi ce test qu'une clé de la serrure mécanique peut être utilisée pour ouvrir la porte. Ce type de collecte est généralement utilisé dans des endroits qui nécessitent plus de protection en raison de l'importance ou du secret de la propriété située derrière le point d'accès prévu, et il peut également être utilisé dans des installations généralement afin d'assurer une protection intensive des individus pour leur propriété. La sécurité est primordial.



Figure 1.3: Serrure à empreinte digitale

1.2.3.5 Contrôle d'accès physique

Quelle que soit la qualité du système de verrouillage ou du système de Contrôle d'accès , lorsque les gens peuvent traverser les points d'accès, cela n'a pas beaucoup d'importance. Par conséquent, il est nécessaire et nécessaire que ce système soit complet et réponde à tous les besoins de sécurité. À titre d'exemples frappants: barrières à capteurs réelles, tourniquets de hauteur moyenne, tourniquets pleine hauteur, tourniquets triples, serrures de sécurité, et si les systèmes de contrôle d'accès sont résidentiels ou commerciaux, ils soutiennent ainsi leur protection contre toute pénétration par toute personne non autorisée entrant ou accédant à un point d'accès. [2]



Figure 1.4: Tourniquets

1.2.4 Fonctionnement du système d'accès

Il y a 5 étapes pour faire fonctionner un système d'accès :

1. authenticité:

Tout d'abord, l'authenticité des données de l'utilisateur est vérifiée après sa soumission s'il connaît le système

2. Autorisation:

La console détermine si l'utilisateur est autorisé à accéder au point d'entrée de la porte ou du port...

3. l'accès :

Une fois autorisé, le mot de passe de la console envoie un message pour ouvrir le point d'entrée. Si la porte est ouverte, le système suit l'utilisateur qui ouvre la porte.

4. Gestion:

Parmi les tâches du système de contrôle d'accès figurent l'ajout ou la suppression d'utilisateurs et d'activités et diverses informations. le logiciel de gestion est la synchronisation automatique avec la console est l'utilisation de la plupart des systèmes de contrôle d'accès et permet aux administrateurs d'accéder à leurs systèmes de contrôle d'accès à tout moment et en tout lieu par un logiciel informatique

5. Vérification:

Les gestionnaires préparent des rapports de journal grâce aux options de vérification fournies par la plupart des systèmes de contrôle d'accès et les rapports peuvent aider à se conformer aux normes conformité et à vérifier la validité du travail du système

1.3 Carte de commande (Arduino)

1.3.1 Définition

C'est un ensemble matériel et logiciel qui permet d'apprendre l'électronique tout en se familiarisant avec la programmation informatique. Arduino est en source libre ; vous pouvez donc télécharger le schéma d'origine et l'utiliser pour élaborer votre propre carte et la vendre sans payer des droits d'auteur ".



Figure 1.5: Composants de l'Arduino

Une carte Arduino est une petite (5,33 x 6,85 cm) carte électronique équipée d'un microcontrôleur . Le microcontrôleur permet, à partir d'événements détectés par des capteurs, de programmer et commander des actionneurs ; la carte Arduino est donc une interface programmable .La carte Arduino la plus utilisée est la carte ArduinoUno.

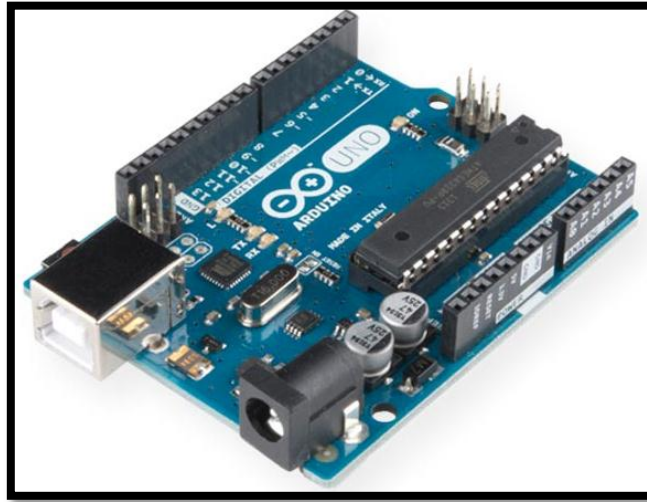


Figure 1.6: la carte de commande ArduinoUno avec ses connecteurs

1.3.2 Types de carte de commande Arduino

La famille Arduino est évoluée à partir de 2007, après ce date on constate plusieurs types tel que :

ArduinoUno (R3)-Arduino Nano-Arduino Micro - Arduino Due-Carte LilyPadArduino-Bluetooth Arduino-Arduino dix mille - Carte ArduinoRedBoard-Carte ArduinoMega (R3)-Arduino Explorer-Arduino zéro ...

1.3.2.1 ArduinoUno (R3)

L'Uno est une énorme option pour votre Arduino initial. Cette carte Arduino dépend d'un microcontrôleur basé sur ATmega328P. Par rapport aux autres types de cartes Arduino elle est très simple à utiliser comme la carte de type ArduinoMega.

ArduinoUno est la carte la plus fréquemment utilisée et c'est la forme standard en dehors de toutes les cartes Arduino existantes. Cette planche est très utile pour les débutants. Veuillez-vous référer à ce lien pour en savoir plus sur Carte ArduinoUno

1.3.2.2 Arduino Nano

Il s'agit d'une petite carte basée sur les microcontrôleurs comme ATmega328P sinon ATmega628 mais la connexion de cette carte est la même que celle de la carte Arduino UNO. Ce type de carte microcontrôleur est de très petite taille, durable, flexible et fiable

En comparaison avec la carte ArduinoUno, elle est de petite taille. Les appareils tels que mini USB et Arduino IDE sont nécessaires pour construire les projets.

1.3.2.3 Arduino Micro

La carte Arduino Micro dépend principalement du microcontrôleur basé sur ATmega32U4 qui comprend 20 jeux de broches où les 7 broches sont des broches PWM, 12 broches d'entrée analogiques. Cette carte comprend différents composants comme un en-tête ICSP, un bouton RST, une petite connexion USB, un oscillateur à cristal de 16 MHz. La connexion USB est intégrée et cette carte est la version réduite de la carte Leonardo.[3]

1.3.3 Composants de la carte de commande Arduino

La carte arduino se compose de plusieurs éléments ,On compte: Microcontrôleur, Port USB, Contrôle USB, connecteur d'alimentation, variateur,(cristal,Réception (RX) et transmission (TX) série,Broches PWM..) Entrées analogiques, Entrées /Sorties numériques .

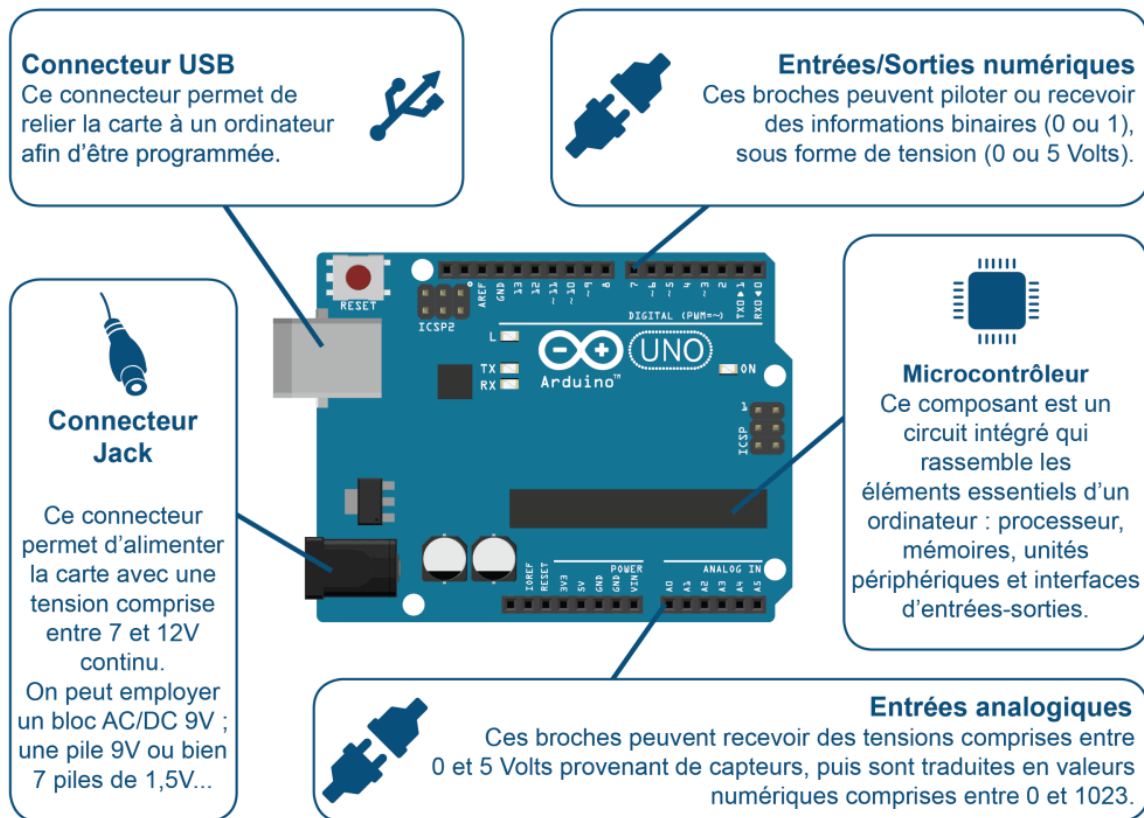


Figure 1.7: Image récapitulative des composants carte Arduino

1.3.3.1 Microcontrôleur

C'est un circuit intégré qui est défini comme la partie qui a le pouvoir de traiter toutes les informations et, par conséquent, est la zone où le code appelé «Sketch» est enregistré dans le logiciel Arduino. Donc, il est estimé comme le cerveau d'une carte Arduino et se caractérise par être un élément noir avec des pieds métalliques.

En ce sens, le microcontrôleur sur une carte Arduino fonctionne comme un circuit intégré programmable qui, par défaut, a la capacité d'effectuer des opérations mathématiques complexes à grande vitesse.

1.3.3.2 port USB

C'est un composant également identifié comme «Connecteur USB» et c'est celui qui vous permet de connecter la carte Arduino à un certain ordinateur. De cette manière, il offre la possibilité de charger le code requis et d'alimenter correctement la carte.

1.3.3.3 Contrôle USB

Fondamentalement, le contrôle USB consiste en un modérateur qui opère entre le microcontrôleur et le logiciel . Raison pour laquelle, cela se concentre sur la conversion des informations transmises par le microcontrôleur en données qui vont à l'équipement utilisé.

1.3.3.4 Connecteur d'alimentation :

Comme son nom l'indique, c'est un élément qui offre la possibilité d'alimenter la carte avec une tension en courant continu pouvant aller de 7 à 12 volts. Compte tenu de cela, chaque fois que la carte Arduino est connectée à une prise électrique, le voyant d'alimentation LED devra s'allumer . Sinon, quelque chose ne va probablement pas.

1.3.3.5 Variateur

Dans une carte Arduino, le régulateur de tension est chargé de contrôler la quantité de tension restante sur la carte . En d'autres termes, il agit pratiquement comme un chien de garde qui empêchera l'entrée d'une tension supplémentaire avec laquelle il pourrait endommager le circuit. Généralement, ce régulateur de tension prend en charge une sortie stable de 5 volts sans dépendre de la tension d'entrée

1.3.3.6 Cristal

Plus précisément, cet élément a la capacité de régler l'impulsion ou le temps de travail du microcontrôleur afin de fonctionner parfaitement sur une carte Arduino. Ainsi, un cristal est une sorte de circuit oscillateur électronique qui crée un signal électrique avec une fréquence précise afin de contrôler le temps de travail exact.

1.3.3.7 Réception (RX) et transmission (TX) série

Ce sont des sortes de marques qui, en électronique, sont utilisées pour indiquer les broches responsables de la communication série . En conséquence, il se compose d'un composant de carte Arduino Uno qui se concentre sur l'externalisation lorsque la carte reçoit ou transmet des données, visuellement. Compte tenu de cela, il consiste en une transmission qui est effectuée via les broches zéro et un.

1.3.3.8 Broches numériques

Ce sont ceux qui s'occupent de détecter s'il y a un ZERO ou UN logique et sont utilisés à la fois pour l'entrée numérique et la sortie numérique . Soit, par exemple, d'appuyer sur un bouton et d'allumer une LED , respectivement. Cela signifie que les broches numériques sont utilisées pour appuyer sur des boutons ou des appareils qui envoient et / ou reçoivent des informations numériques Ces broches vont de 0 à 13.

1.3.3.9 Broches analogiques

Ils correspondent à la surface des broches dans le cadre du «analogique» et grâce à cela, ils sont capables de détecter des signaux analogiques (température ou lumière, par exemple). Pour ce faire, ils disposent d'un segment de tension de fonctionnement qui oscille entre 0 et 5 volts . En conséquence, les broches analogiques sont des broches qui peuvent lire le signal d'un capteur analogique et le transformer en une valeur numérique qui peut être affichée . Ce qui signifie que, grâce aux broches analogiques, il est possible de mesurer des choses dans le monde réel

1.3.3.10 Broches d'alimentation du capteur

Ils fournissent une alimentation entre 6 et 12 volts qui, directement, va au régulateur de tension. Grâce à ces broches, il existe également la possibilité d'alimenter l'Arduino avec des piles ou des piles . Cependant, de cette manière, il n'y a pas de diode de protection de polarité ou de protection contre les surintensités.

1.3.3.11 Broches PWM

Ce sont des broches spécifiques qui agissent comme des broches numériques normales et, en même temps, peuvent être utilisées pour la «modulation de largeur d'impulsion» ou ce que l'on appelle «PWM» en anglais. Physiquement, ces broches se distinguent en contenant un tilde (~) à côté de certaines des broches numériques, telles que: 3, 5, 6, 9, 10 et 11.

1.3.3.12 Broche AREF

Il s'agit d'une seule broche qui, la plupart du temps, peut être laissée seule en place. Fondamentalement, il s'agit d'un élément qui a un support de référence analogique et est parfois utilisé pour pouvoir établir une tension de référence externe entre 0 et 5 volts. Qui fonctionne comme la limite supérieure des broches d'entrée analogiques.

1.3.3.13 Broches GND

Sur une carte Arduino, il y a toujours plusieurs broches GND et l'une d'entre elles peut être utilisée pour connecter le circuit à la terre. D'où son nom qui est l'abréviation de «land» en anglais. Pour sa part, lorsqu'il s'agit de le localiser, il se trouve juste à côté du pin AREF.

1.3.3.14 Bouton de réinitialisation

Il se réfère au bouton de réinitialisation qui est connecté à la broche numéro un du microcontrôleur, directement. Ce qui est mieux connu sous le nom de «Clear» ou «Master Clear». Maintenant, une fois que ce bouton est enfoncé, la broche de réinitialisation est temporairement connectée à la terre, afin de réinitialiser tout code chargé sur la carte Arduino. [4]

Il convient de noter que cela nécessite 5 volts pour exécuter le programme, tandis que le bouton interrompt la tension à 0 volts pour arrêter le même programme qu'Arduino est en cours d'exécution, afin de le redémarrer.

1.4 Conclusion

Les connaissances théoriques acquises dans ce chapitre nous permettront d'entamer le deuxième chapitre consacré à simulation et à la réalisation pratique.

La maîtrise du langage de programmation de la carte arduino est exigée pour l'élaboration d'un code source optimal qui répondra à nos besoin.

Chapitre 2

Simulation

et

Réalisation Pratique

2.1 Introduction

La simulation et la réalisation du système feront l'objet de ce dernier chapitre. on va simuler pour prévoir les réponse de notre système ensuite on passera à la programmation de la carte de commande et à son câblage avec les autres éléments du projet.

2.2 Principe de notre réalisation

Dans ce projet, nous nous sommes concentrés sur l'amélioration de la sécurité, de la fiabilité et de la résilience. L'idée générale de notre système est de permettre aux utilisateurs de déverrouiller avec un code PIN pour se débarrasser de la notion dominante de port de clés et de la peur de les oublier ou de les perdre

2.3 Réalisation du montage

2.3.1 Organigramme de fonctionnement

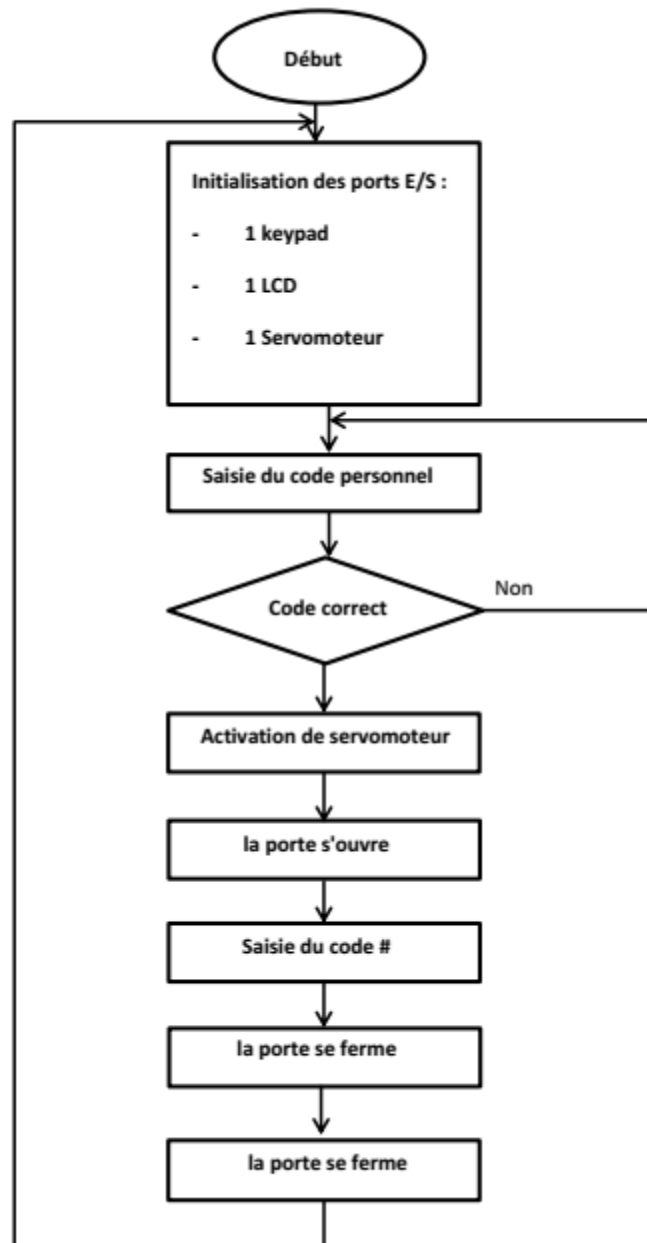


Figure 2.1: Organigramme de fonctionnement

2.3. 2 Schéma synoptique du système

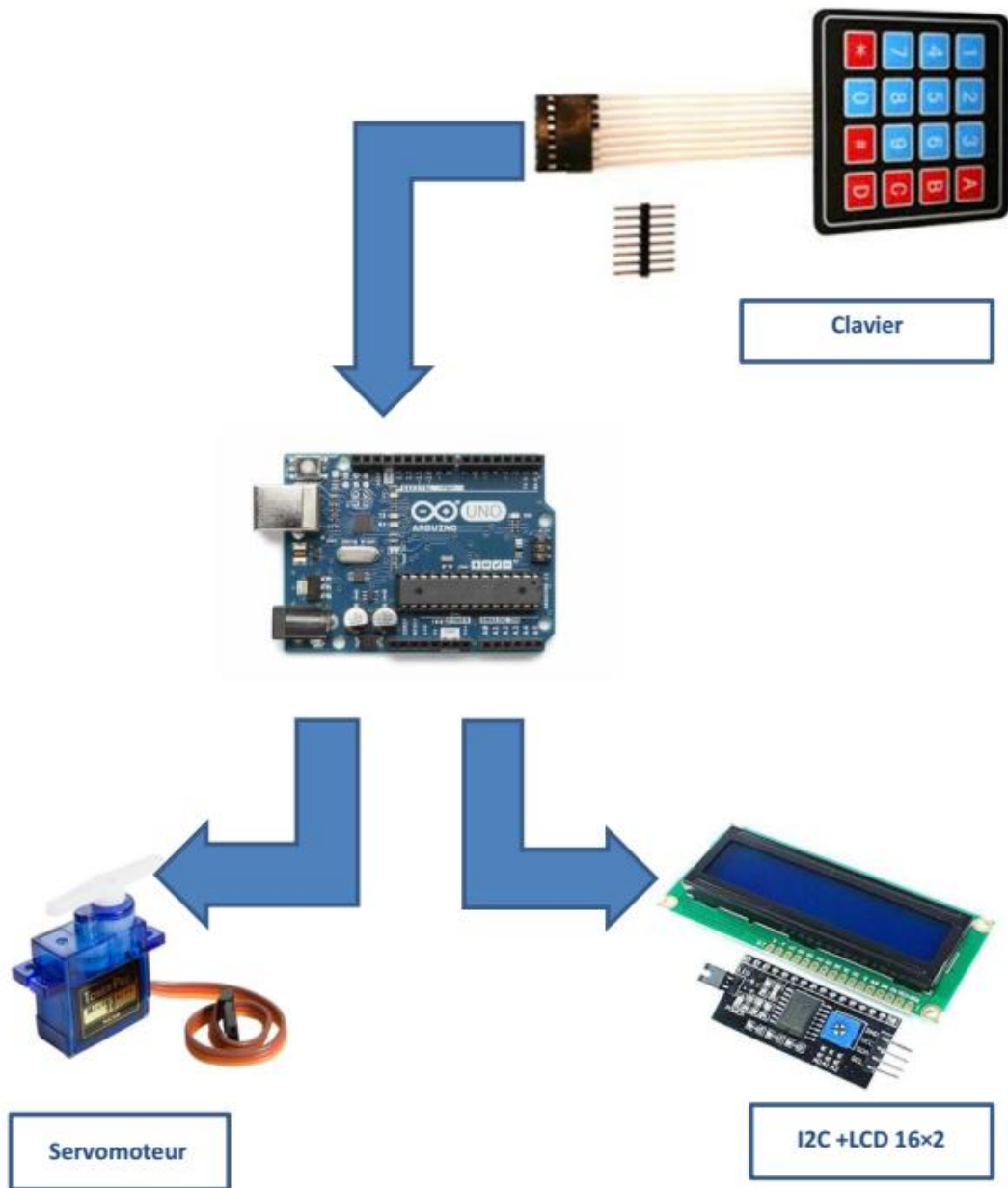


Figure 2.2: Schéma synoptique du système

2.4 Câblage du montage

2.4.1 Câblage du clavier

Le clavier comporte 16 touches, dont 10 numériques (0-9) et 6 touches marquées # * A B C D.

Il y a 8 fils en sortie du clavier, 4 lignes et 4 colonnes.

Vu de face, de gauche à droite :

pin 1-4 les quatre lignes, pin 5-8 les quatre colonnes.

On le branche sur les 8 entrées digitales successives de la carte ArduinoUno de D8 à D1 :

Pin 1--> D8 - pin 2--> D7 - pin 3--> D6 - pin 4--> D5 - pin 5--> D4 - pin 6--> D3 - pin 7--> D2 -
pin 8--> D1

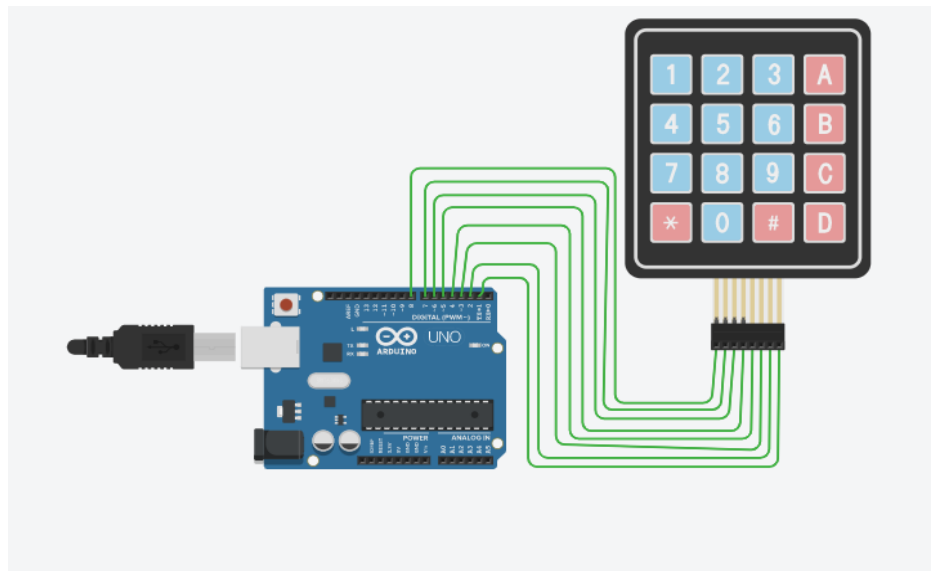


Figure 2.3: Câblage du clavier

2. 4. 2 Câblage d'un afficheur LCD +I2C

L'écran à cristaux liquides 16×02 avec module I2C est connecté à la carte Arduino avec seulement 4 fils : 2 fils de données et 2 fils d'alimentation.

La connexion du l'écran LCD 16×02 à l'Arduino se fait de manière standard pour le bus I2C :

- la broche SDA est connectée au port SDA.
 - la broche SCL au port SCL.
 - la broche VCC au port +5V.
 - la broche GND au port GND.
- Consultez l'image ci-dessous pour obtenir un schéma de câblage détaillé de l'écran LCD 16×02.

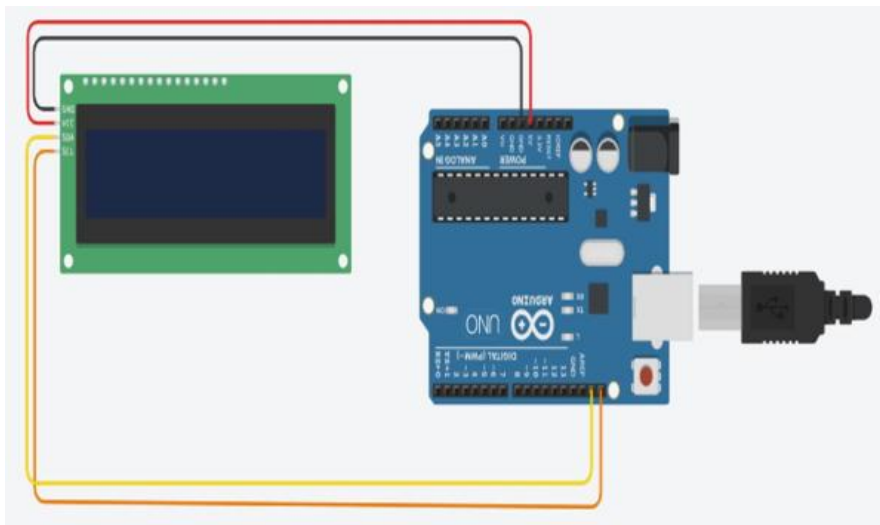


Figure 2.4: Câblage d'un afficheur LCD +I2C

2. 4. 3 Câblage du servomoteur

Les servomoteurs sont des moteurs un peu particuliers, qui peuvent tourner avec une liberté d'environ 180° et garder de manière relativement précise l'angle de rotation que l'on souhaite obtenir.

Nous avons utilisé une broche qui est une entrée numérique afin de commander notre moteur alimenté à partir de notre carte Arduino (5 V).

Dans notre programme nous utilisons la librairie SERVO qui regroupe un grand nombre de fonctions permettant de contrôler le servomoteur.

La carte Arduino a été liée au servomoteur selon le schéma ci- dessous :

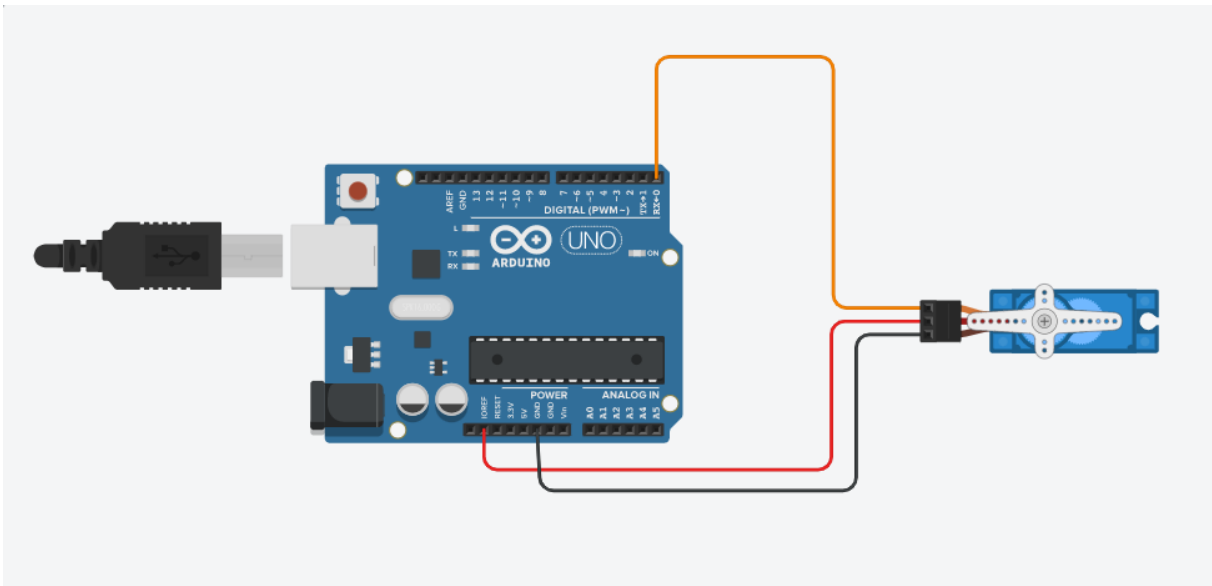


Figure 2.5: Câblage du servomoteur

2.5 Logiciel (IDE Arduino)

L'environnement de programmation Arduino (IDE en anglais) est une application écrite en Java. L'IDE permet d'écrire, de modifier un programme et de le convertir en une série d'instructions compréhensibles pour la carte. Le logiciel va nous permettre de programmer la carte Arduino, il nous offre une multitude de fonctionnalités.

La structure des programmes Arduino est un peu particulier, en apparence, des structures habituelles du langage C. La syntaxe est la même qu'en langage C.



Figure 2.6: Logo du logiciel IDE Arduino

Au début du programme, la déclaration des bibliothèques utilisée par le programme à compiler avec le programme.

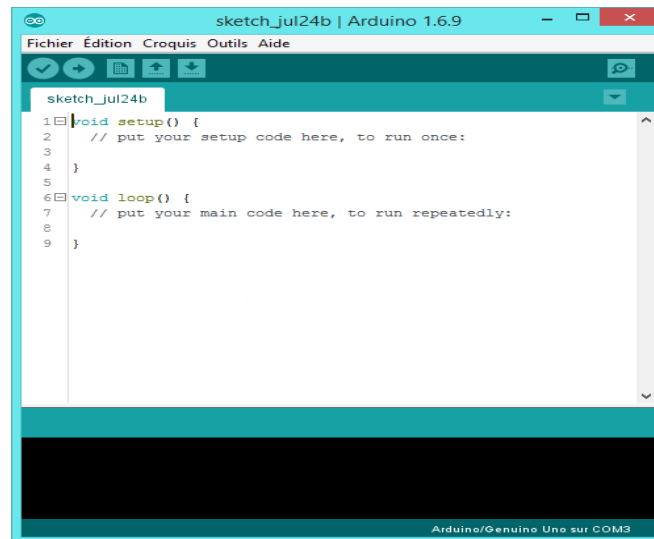
- Nouveau : pour créer un nouveau programme (sketch).
- Ouvrir : ouvrir un programme existant. Le menu n'est pas déroulant à cause d'un bug...pour obtenir un menu déroulant passer par file/open
- Enregistrer : sauvegarde le programme
- Moniteur série : pour ouvrir la fenêtre qui permet de visualiser les données transmises par le port série de l'Arduino.

- Le programme est lu par le microcontrôleur de haut vers le bas.
- Une variable doit être déclarée avant d'être utilisée par une fonction.

La structure minimale est constituée :

- En tête : déclaration des variables, des constantes, indication de l'utilisation de bibliothèques
- un setup (ou initialisation) cette partie n'est lue qu'une seule fois, elle comprend les fonctions qui doivent être réalisées au démarrage (utilisation des broches en entrées ou en sortie)

Une loop (boucle) : cette partie est lue en boucle, c'est ici que les fonctions sont réalisées.



```
sketch_jul24b | Arduino 1.6.9
Fichier Édition Croquis Outils Aide
sketch_jul24b
1 void setup() {
2   // put your setup code here, to run once:
3
4 }
5
6 void loop() {
7   // put your main code here, to run repeatedly:
8
9 }
```

Arduino/Genuino Uno sur COM3

Figure 2.7: Éditeur de texte de l'IDE Arduino

Pour démarrer avec Arduino sous Windows, on procède comme suit:

1. Obtenir une carte Arduino et un câble USB
2. Télécharger l'environnement Arduino
3. Raccorder la carte à l'ordinateur : La diode verte doit s'allumer.

4. Installation des pilotes du périphérique Série-USB
5. Lancer l'application Arduino
6. Ouvrir l'exemple « blink »
7. choisir la carte UNO
8. sélectionner le port série
9. charger le programme dans la carte Arduino

2. 5. 1 Code source et schéma du montage

2. 5. 1.1 Code source et langage

le langage utilisé avec Arduino est très proche du langage C ou C++. C'est notamment le cas pour la syntaxe et les types de variables. Cependant, ce langage présente quelques différences, principalement dans la structure d'un programme.

Un programme Arduino se décompose en deux parties principales : une première partie appelée setup, et une seconde appelée loop. Ces deux parties sont représentatives de la manière de fonctionner d'un microcontrôleur et sont appelées dans cet ordre. En effet, un microcontrôleur est destiné à exécuter les instructions contenues dans sa mémoire de manière cyclique et infinie.

Avant de s'exécuter, le microcontrôleur a besoin d'une phase de « configuration » de son environnement, notamment pour paramétrer des broches d'entrées/sorties ou tout simplement pour fixer des variables globales. Cette phase correspond à la fonction setup.

La fonction setup se présente de la manière suivante :

```
voidsetup(){  
  
[Instructions] ;
```

```
}
```

Les instructions contenues dans cette fonction ne sont exécutées qu'une seule fois, et ce, lors de la mise sous tension de la carte.

Une réinitialisation de la carte est considérée également comme une mise sous tension. La fonction setup est donc également exécutée après un reset.

Une fois la configuration effectuée, le microcontrôleur débute l'exécution des instructions contenues dans sa mémoire. Cette phase correspond à la fonction loop.

La fonction loop se présente de la manière suivante :

```
voidloop(){  
  
[Instructions] ;  
  
}
```

Cette fonction s'apparente à une boucle ne possédant aucune condition d'arrêt. Le seul moyen de l'arrêter est de couper l'alimentation de la carte ou de réinitialiser la carte. Les instructions contenues dans cette fonction sont donc répétées aussi longtemps que la carte sera alimentée.

Il est impératif que ces deux fonctions soient présentes dans le programme principal. L'ordre d'apparition de ces fonctions dans le programme n'a pas d'importance. Cependant, il est conseillé de garder la déclaration de ces fonctions dans cet ordre et d'éviter de les séparer. Cela permet de rendre plus lisible le code, et surtout cela permet à un autre lecteur de repérer plus rapidement le début de la fonction principale.

Il est également possible de déclarer des fonctions dans le même fichier que celui dans lequel est présent le programme Arduino. Ces déclarations de fonctions pourront s'insérer à tout endroit du programme, du moment que la syntaxe est respectée. Enfin, il est possible, tout comme

dans les programmes en C ou C++, d'inclure des bibliothèques de fonctions dans son programme. Cette inclusion devra s'effectuer, si nécessaire, avant la déclaration de toute fonction.

Il est important de structurer efficacement son programme, afin de permettre une meilleure relecture et surtout de repérer rapidement d'éventuelles erreurs. Un programme Arduino « efficace » pourrait se composer de la manière suivante :

-Inclusion ou non des bibliothèques utilisées-

-Déclaration des fonctions utilisées-

```
voidsetup(){
```

```
[Instructions exécutées une seule fois lors de la mise sous tension de la carte] ;
```

```
}
```

```
voidloop(){
```

```
[Instructions exécutées en boucle par le microcontrôleur] ;
```

```
}(2)
```

(livrearduino apprendre à développer pour créer des objets intelligents)

➤ **Le code réaliser fait les instructions suivantes**

- Lire les chiffres entrés par le clavier via les broches 1 ,2,3,4,5,6,7,8.
- Comparé le code entré avec le code enregistré.
- Si le code est correct, tourner le moteur (attacher au pin 0) à droite avec une valeur de 180°.

- Lorsque vous appuyez sur # (start), tourner le moteur (attacher au pin 0) à gauche avec une valeur de 180°.

➤ Le déroulement du programme

Le programme se déroule de la façon suivante :

- Après avoir pris en compte les instructions de la partie déclarative
- Puis après avoir exécuté la partie configuration (fonction setup ())
- Le programme bouclera sans fin (fonction Loop ()), exécutant de façon répétée de code compris dans la boucle sans fin.

➤ La partie déclaration

Nous commençons par importer les bibliothèques dont nous avons besoin dans notre code

```
1 #include <Keypad.h> // the library for the 4x4 keypad
2 #include <LiquidCrystal_I2C.h> // the library for the i2c 1602 lcd
3 #include <Servo.h> // the library to control the servo motor
4 LiquidCrystal_I2C lcd(0x27,20,4); // gets the lcd
5 Servo servo;
6
7 #define Password_Length 8 // the length of the password, if the passw
8 int Position = 0; // position of the servo
9 char Particular[Password_Length]; // the password length
```

On a besoin des bibliothèques de keypad, moteur servo, de lcd I2C

Ensuite, nous déclarons les variables de différent type caractère intègre et boolien

```

6
7 #define Password_Length 8 // the length of the password, if the
8 int Position = 0; // position of the servo
9 char Particular[Password_Length]; // the password length
10 char Specific[Password_Length] = "123456B"; // the password whi
11 byte Particular_Count = 0, Specific_Count = 0; // counts the an
12 char Key;
13 const byte ROWS = 4; // the amount of rows on the keypad
14 const byte COLS = 4; // the amount of columns on the keypad
15 char keys[ROWS][COLS] = { // sets the rows and columns

```

Ensuite, nous initialisons le clavier avec déclaration de brochage des lignes et des colons

```

16 // sets the keypad digits
17 {'1','2','3','A'},
18 {'4','5','6','B'},
19 {'7','8','9','C'},
20 {'*','0','#','D'}
21 };
22 bool SmartDoor = true; // the servo
23 // the pins to plug the keypad into
24 byte rowPins[ROWS] = {8, 7, 6, 5};
25 byte colPins[COLS] = {4, 3, 2, 1};
26 Keypad myKeypad(makeKeymap(keys), rowPins, colPins, ROWS, COLS); // gets the data from the keypad

```

➤ La partie de configuration(fonction setup)

Dans cette partie on fait l'initialisation des composants

On a initialisé le servo moteur à pin D0 d'Arduino

```

51 {
52 servo.attach(0); // attaches the servo to pin 0
53 ServoClose(); // closes the servo when you say this function

```

Ensuite, on a configuré LCD pour afficher le message de bienvenu (Salem MCIL)

```
51 {
52  servo.attach(0); // attaches the servo to pin 0
53  ServoClose(); // closes the servo when you say this function
54  lcd.init(); // initializes the lcd
55  lcd.backlight(); // turns on the backlight
56  lcd.setCursor(0,0); // sets the cursor on the lcd
57  lcd.print("salam MCIL3"); // prints the text/charater
58  lcd.setCursor(0,1); // sets the cursor on the lcd
59  lcd.print("Porte Protegee"); // prints text
60  delay(4000); // waits 4 seconds
61  lcd.clear(); // clears the lcd diplay
62
63 }
```

➤ **La partie de fonction Loop**

Cette partie de code constituer des instructions exécutées en boucle infini fonctionner les fonctions suivantes :

➤ **La fonction lock**

C'est la fonction responsable d'ouvrir la serrure ou bien tourner le moteur

➤ **La fonction unlock** C'est la fonction responsable de fermer

2. 5.1.2 Le schéma général du montage

Le schéma général (réel)

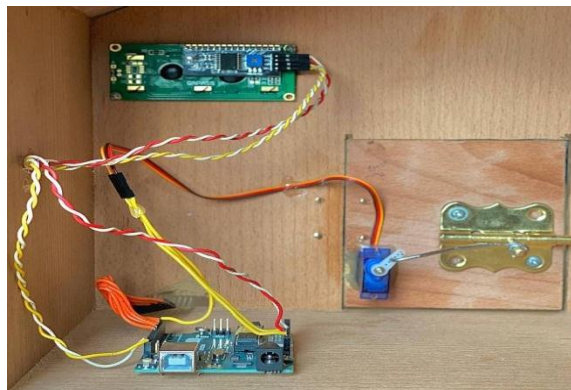


Figure 2.8: Schéma général réel

➤ Le schéma général (tinkercad)

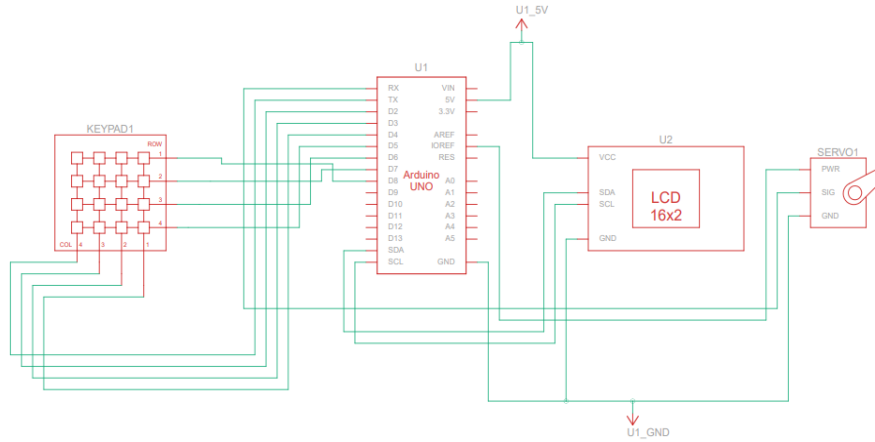


Figure 2.9: Schéma général (tinkercad)

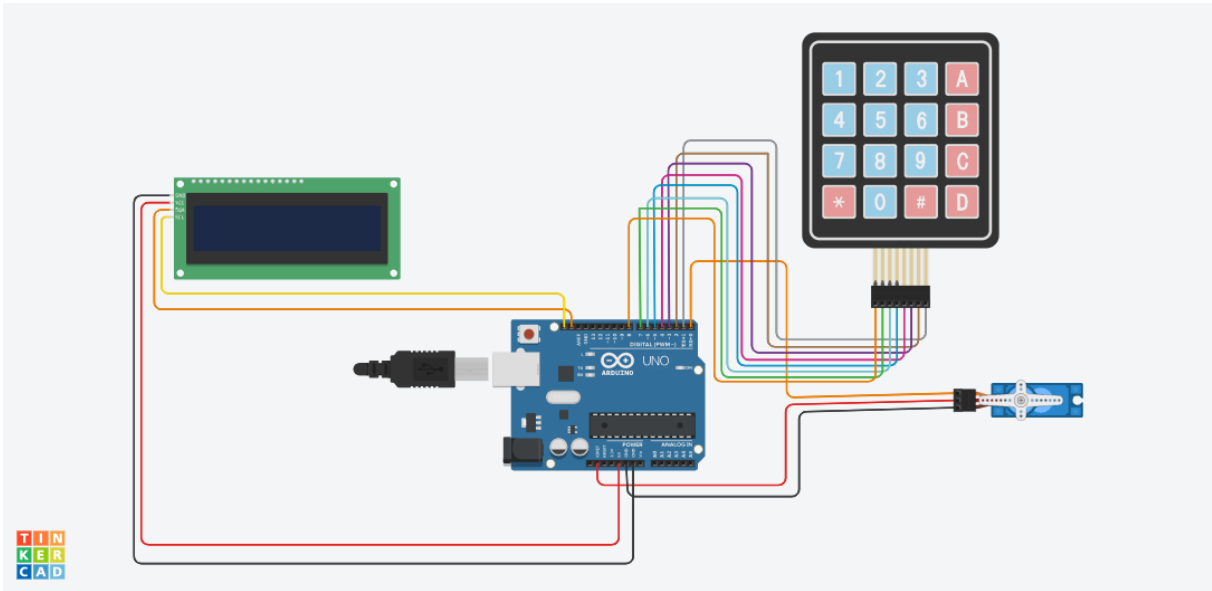


Figure 2.10: Schéma représentatif du câblage entre les différents organes

2.6 Prototype de notre projet



Figure 2.11: Prototype de la réalisation

2.7 Conclusion

Dans ce chapitre nous avons présenté le système de serrure électronique codée par code PIN. Pour réaliser ce projet nous avons consacré des outils soft et hard.

Concernant les logiciels, nous avons utilisé tinkercad et Arduino IDE. Pour le hard nous avons utilisé un servomoteur, un afficher LCD, un clavier, et une carte Arduino pour commander le système.

Conclusion générale

En conclusion, l'objectif de mettre en place d'un système d'accès sécurisé basé sur le code PIN afin de gérer l'ouverture et fermeture d'une porte a été atteint avec succès.

En termes de performance et d'efficacité, ce projet permet de se protéger et de se sécuriser contre tous les risques, qu'il s'agisse d'intrusion ou de perte de la clé.

Ce système est également facile à utiliser car il permet à l'utilisateur de suivre la saisie du code PIN et l'ouverture et la fermeture de la porte via l'écran LCD.

Ce projet nous a permis de se familiariser avec l'identification des différentes broches d'entrées / sorties de la carte Arduino, ainsi que l'apprentissage de la syntaxe du langage de programmation.

Comme perspective et dans le but d'améliorer les performances du système on peut intégrer un module d'accès wifi ou Bluetooth. Aussi, la possibilité de développer une application pour l'enregistrement de l'historique des différents accès.

Résumé

Notre projet s'intéresse à la sécurité domestique en particulier aux systèmes de verrouillage de porte, nous avons fait l'étude et la réalisation d'une serrure de porte codée par code PIN .

Le système est basé sur la plate-forme Arduino. Il garantira un accès sans clé car le code pin est entré à l'aide d'un clavier, Si le code pin est correct la porte s'ouvrira, en plus un afficheur LCD garantit l'interaction homme-machine.

ملخص

يهتم مشروعنا بأمن المنازل وعلى وجه التحديد أنظمة قفل الأبواب الإلكترونية، وقد قمنا بدراسة وإنتاج قفل باب مشفر برمز PIN.

يعتمد النظام على منصة اردوينو Arduino. يضمن الوصول بدون مفتاح لأنه يتم إدخال الرمز السري (PIN) باستخدام لوحة المفاتيح ، إذا كان الرمز السري صحيحًا ، فسيتم فتح الباب تلقائيًا ، بالإضافة إلى شاشة LCD التي تضمن التعامل المباشر مع النظام وفق الخيارات المتاحة.

Bibliographies

[1]. Classification des systèmes d'accès : mémoire de fin d'études :SYSTÈME DE
CONTROLE D'ACCES PHYSIQUE

[2]. <https://blog.dormakaba.com>

[3]. <http://fr.jf-parede.pt>

[4]. <https://www.informatique-mania.com>