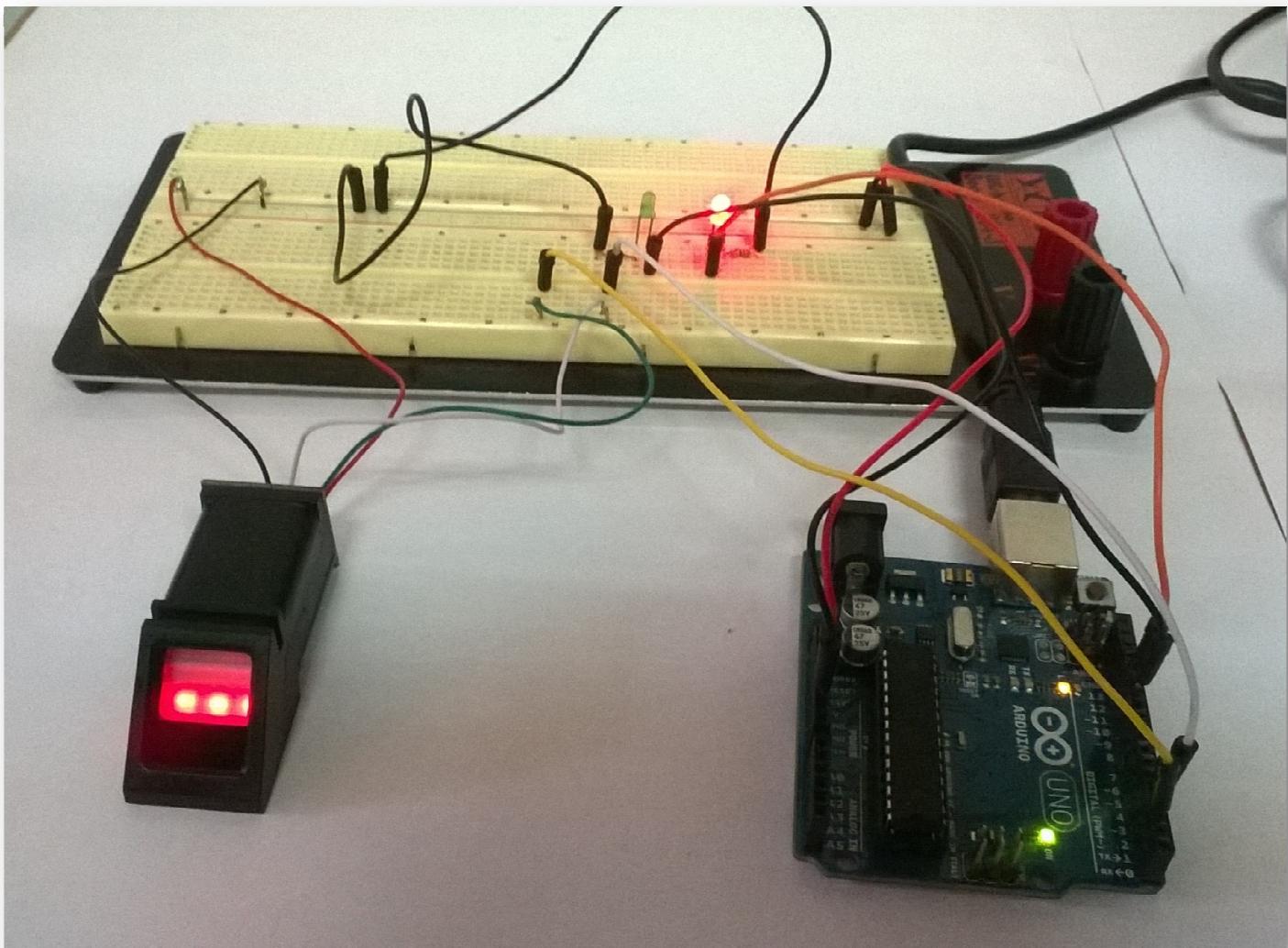




Cadi Ayyad University

Université Cadi Ayyad  
Faculté des sciences Semlalia  
Marrakech



## PROJET DE FIN D'ETUDE

# Control d'accès par empreintes digitale

Encadré par : M.ROUFI

Les jurés : M.ANKRIM

M.M.HASSANI

Mdm.S.IBNYAICH

Réalisé par : NAJMA Soufiane

AIT BBA Mohamed

AABIDA Abderrahime

Filière : SMP S6

Option : Electronique

Année Universitaire : 2014-2015

# Remerciement

Dans premier temps, Nous souhaitant d'exprimer notre vifs remerciement et notre gratitude à notre grande et respectueux professeur M.RAOUI d'avoir accepté de nous encadrer pour notre projet de fin d'étude, ainsi que pour ses soutiens, ses remarques pertinentes et son encouragement.

Lors de préparation de ce projet, nous avons également bénéficié de la collaboration bénévole de professeurs et d'amis qui ont bien voulu discuter avec nous les différents aspects de notre projet, ce qui nous a permis de mieux préciser certains aspects de ce travail.

Il nous est également agréable de remercier Monsieur le président du jury pour sa grande disponibilité. Nous tenons à remercier aussi toute l'équipe pédagogique de la faculté des sciences semlalia pour les efforts fournis et les conditions de travail afin de mieux réussir dans cette tâche.

Enfin, nous adressons nos plus sincères remerciements à notre familles et tous notre proches amis, qui nous ont toujours soutenu et encourage au cours de la réalisation de ce projet.

Merci à toutes et à tous.

# **SOMMAIRE:**

<b>INTRODUCTION.....</b>	<b>1</b>
<b>LE CONTROLE D'ACCES .....</b>	<b>2</b>
<b>EMPREINTES DIGITALES.....</b>	<b>3</b>
I. INTRODUCTION .....	3
II. LES CAPTEURS DE L'EMPREINTE .....	4
1. introduction.....	4
2. Les types des capteurs .....	4
<b>LA CARTE MICROCONTROLEUR (ARDUINO UNO R3) .....</b>	<b>7</b>
I. INTRODUCTION .....	7
II. DESCRIPTION DE LA CARTE "ARDUINO UNO R3 "(PARTIE MATERIEL) .....	8
1. Le Microcontrôleur « ATMEGA 328 ».....	8
2. Les Broches de la carte.....	9
III. PROGRAMMATION DE L' "ARDUINO" (PARTIE LANGAGE).....	10
1. Structure.....	12
2. Variables et constantes : .....	13
3. Fonctions : .....	13
4. Bibliothèques :.....	14
<b>LA PARTIE DE LA REALISATION .....</b>	<b>15</b>
I. INTRODUCTION .....	15
1. Spécification : .....	15
2. Les pièces utilisées : .....	15
II. PROGRAMMATION DE L'EMPREINTE.....	16
1. Première étape.....	16
2. La deuxième étape: .....	18
<b>CONCLUSION .....</b>	<b>21</b>
<b>LES ABRIVIATIONS.....</b>	<b>22</b>
<b>REFERENCES.....</b>	<b>23</b>

# **La Table des figures**

<b>FIGURE 1 : LA FORME DES CREUX ET DES BOSSES DE L'EMPREINTE.....</b>	<b>3</b>
<b>FIGURE 2 : LA TRACE DE L'EMPREINTE .....</b>	<b>3</b>
<b>FIGURE 3 : LE CAPTEUR THERMIQUE " ATMEL FINGERCHIP" .....</b>	<b>4</b>
<b>FIGURE 4 : LA CAPTURE SILICIUM D'EMPREINTE .....</b>	<b>5</b>
<b>FIGURE 5 : LE CAPTEUR ULTRASONIQUE.....</b>	<b>5</b>
<b>FIGURE 6 : UN CAPTEUR CAPACITIF .....</b>	<b>5</b>
<b>FIGURE 7 : UN CAPTEUR DU CHAMP ELECTRIQUE DE TYPE AUTHENTEC AES4000 .....</b>	<b>6</b>
<b>FIGURE 8 : LE CAPTEUR DE PRESSION .....</b>	<b>6</b>
<b>FIGURE 9 : LE PRINCIPE DE FONCTIONNEMENT DU CAPTEUR OPTIQUE.....</b>	<b>6</b>
<b>FIGURE 10 : STRUCTURE D'UN SYSTEME A MICROPROCESSEUR .....</b>	<b>7</b>
<b>FIGURE 11 : LES BROCHES DE LA CARTE "ARDUINO UNO R3".....</b>	<b>9</b>
<b>FIGURE 12.....</b>	<b>11</b>
<b>FIGURE 13 : LA FENETRE DE LOGICIEL DE LA PROGRAMMATION D' "ARDUINO " .....</b>	<b>11</b>
<b>FIGURE 14 : LE CAPTEUR D'EMPREINTE DIGITALE SEN04172P .....</b>	<b>15</b>
<b>FIGURE 15: LES PIECES UTILISEES.....</b>	<b>15</b>
<b>FIGURE 16.....</b>	<b>16</b>
<b>FIGURE 17.....</b>	<b>16</b>
<b>FIGURE 18.....</b>	<b>17</b>
<b>FIGURE 19.....</b>	<b>17</b>
<b>FIGURE 20.....</b>	<b>17</b>
<b>FIGURE 21: LE CAPTEUR D'EMPREINTE DIGITALE RELIER AVEC L'ARDUINO.....</b>	<b>18</b>
<b>FIGURE 22.....</b>	<b>18</b>
<b>FIGURE 23.....</b>	<b>18</b>
<b>FIGURE 24.....</b>	<b>18</b>

## INTRODUCTION

La sécurité représente une préoccupation au sein des entreprises et de commerce par l'accès à l'information. Est cela pour éviter l'accès par des personnes indélicates. Dans ce cadre une nouvelle technique de control d'accès a fait son apparition. Il s'agit de control d'accès par les systèmes biométriques. Ces systèmes sont utilisés aussi bien pour des control d'accès physique (l'œil, visage...) que pour des contrôles d'accès logique (mot de passe, carte à puce...).

Dans le cadre de notre projet on s'intéresse par le control d'accès par les empreintes digitales ces derniers existent déjà à notre naissance [6].En effet les empreintes digitales sont individuelles, cependant, elles ne sont en fait pas si différentes les unes des autres.

Pour réaliser ce contrôle d'accès à partir des empreintes digitales captées par des capteurs d'empreintes, on utilise un système "Arduino" qui nous donne la possibilité d'assemblé les performances de la programmation et l'électronique, plus précisément, nous allons programmer des systèmes électroniques.

## Le Contrôle d'accès

Les contrôle d'accès est une technique qui consiste à soumettre l'entrée d'un établissement à une autorisation préalable d'accès.

L'autorisation d'accès s'affecte de trois façons différentes de prouver son identité face à un système informatique :

- 1) L'autorisation d'accès par un mot passe ou un code.

Le Mot de passe est saisi par un clavier électronique. Ce clavier situé à l'intérieur d'un local fermé élimine toute possibilité d'ouverture non désiré.

- 2) L'autorisation d'accès par un objet, telle une carte à puce, un badge.

En générale les informations sont stockées dans une mémoire insérée sur un support plastique. Exemple : carte vitale ou carte bancaire.

- 3) L'autorisation d'accès grâce à une caractéristique physique propre (biométrie).

Exemple : (l'œil, visage, empreinte...).

Les types	Les avantages	Les inconvénients
<b>Premier type (code)</b>	Programmation avec un code que l'on peut choisir. Facilité de Changer le code. La sécurité peu s'élever.	La Possibilité de: -l'oublier -pirater
<b>Deuxième type (carte à puce)</b>	Grande confort d'utilisation. Un seul badge permet d'ouvrir plusieurs portes.	La Possibilité de: - perte de la carte -distinction -voler la carte....
<b>Troisième type (l'empreinte)</b>	On ne peut pas l'oublier. Difficile à copier. On ne peut pas le confié. Unique et personnel. Pas de risque de l'oublier et de le détruire.	Positionnement assez lent pour contrôle

## Empreintes digitales

### I. INTRODUCTION

Chez les êtres humains, les bouts des doigts ne sont pas lisses. Sur la dernière phalange de chaque doigt, ainsi que sur toute la paume de la main, la peau forme des creux et des bosses (figure 1). Les crêtes permettent une meilleure prise des objets. Les bosses quand à elles possèdent des pores qui permettent de laisser échapper la sueur créée par les glandes sudoripares.

Lorsque nous touchons un objet (ou une surface), ces liquides gras (sueurs, sécrétions grasses, acides aminés et déchets) se déposent en suivant exactement les crêtes du doigt: une empreinte digitale apparaît comme le montre dans la figure 2. Une empreinte peut se conserver des années sur la plupart des supports.

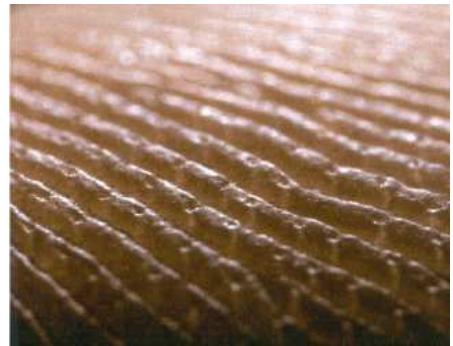


Figure 1 : la forme des creux et des bosses de l'empreinte

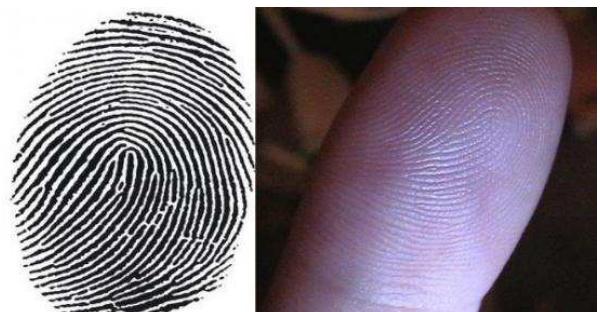


Figure 2 : la trace de l'empreinte

Les Empreintes digitales ne subiront aucune modification (même pendant la croissance) et disparaîtront uniquement par putréfaction. Même si l'on se brûle, ou si l'on se coupe, les détails papillaires se reconstitueront sans cesse à l'identique. De plus, chaque empreinte possède un grand nombre de caractéristiques qui dépendent du patrimoine génétique, ainsi que d'autres facteurs chez le fœtus comme, l'alimentation, la pression sanguine, la vitesse de croissance des doigts ... La probabilité de trouver deux empreintes semblables est d'une chance sur 64 milliards, autant dire que c'est impossible. Les empreintes digitales sont donc uniques, immuables et inaltérables tout au long de notre vie. C'est pour cela que la police scientifique a recours à la dactyloscopie, qui reste à l'heure actuelle la méthode la plus utilisée mais aussi la plus fiable pour identifier une personne. [3]

## II. LES CAPTEURS DE L'EMPREINTE

### 1. introduction

Les techniques utilisées pour capter les images des empreintes digitales sont diverses : capteurs optiques (caméras CCD), capteurs d'ultrasonique, capteurs de champ électrique, de capacité, de température...

Les difficultés générales de toutes ces techniques sont principalement :

- ❖ L'épaisseur très fine de certaines empreintes digitales dues parfois à l'origine ethnique de la personne ou encore de l'âge.
- ❖ La quantité énorme d'informations sur une très petite surface.
- ❖ L'état du doigt (humidité, sécheresse, coupures...).
- ❖ La pression plus ou moins forte exercée sur le détecteur. D'où l'importance de bien choisir l'appareil recueillant l'empreinte.

Le but de tous ces capteurs converge : La capture de l'image d'une empreinte digitale consiste à trouver les lignes tracées par les crêtes (en contact avec le capteur) et les vallées (creux)

### 2. Les types des capteurs

Nous retrouvons sept catégories de capteurs dont les plus importants sont :

#### a) Le capteur thermique :

La technique de capture thermique est utilisée par le FingerChip d'Atmel (figure 3). Le capteur mesure une différence de température obtenue selon que la peau touche (dans le cas d'une crête de l'empreinte) ou ne touche pas (pour une vallée) le capteur.

Le FingerChip est constitué d'une puce en silicium recouverte d'une couche de matériau piézo-électrique, c'est-à-dire sensible aux différences de température. La puce est elle-même formée d'une matrice de pixels adjacents. La différence de température, initialement apparue au contact du matériau piézo-électrique, est transformée de par les propriétés de ce matériau en charges électriques. [4]



Figure 3 : le capteur thermique "Atmel FingerChip"

### b) Le capteur en silicium :

Il utilise un de quatre effets observables sur les semi-conducteurs qui sont :

- l'effet piézo-électrique
- l'effet capacitif
- l'effet thermoélectrique
- l'effet photo-électrique.

Il est en général de très petite taille, d'une durée de vie assez longue, et son coût est très intéressant.

Mais, comme tout composant, il est fragile aux décharges électrostatiques et il peut être détruit si des règles de fabrication et d'installation ne sont pas observées. [4]

### c) Le capteur ultrasonique :

Il utilise une onde ultra sonore qu'il envoie vers le doigt, puis calcule le temps mis par l'onde pour faire un aller-retour et, point par point, fournit l'image de l'empreinte

Il est très précis, et hérite des propriétés des ultrasons de traverser certains matériaux (gants en latex, saletés, etc.). Mais il est volumineux et très coûteux. Il est intéressant pour une population d'utilisateurs très hétérogène. [4]



Figure 4 : la capture silicium d'empreinte



Figure 5 : le capteur ultrasonique

### d) Le capteur capacitif :

La mesure de la capacité électrique (La capacité représente la quantité de charge électrique stockée pour un potentiel électrique donné. Elle est définie:  $\sum \frac{Q_i}{U_i}$ ) entre la peau et le pixel est l'effet physique le plus souvent mis en œuvre. Comme le champ électrique mesuré entre la peau et le pixel est très faible, la protection surfacique doit être très mince (quelques microns) afin d'obtenir une sensibilité correcte. [4]

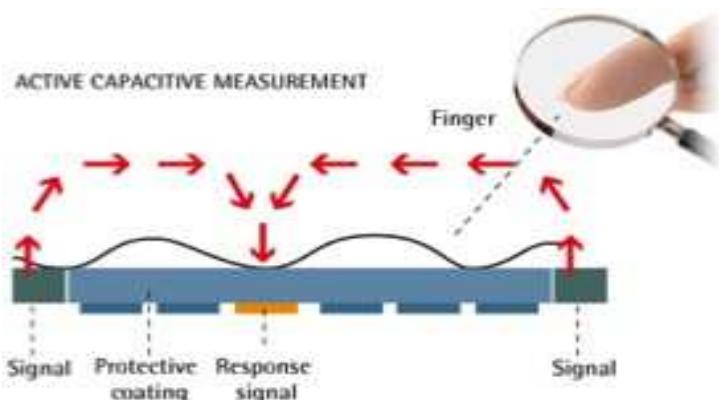


Figure 6 : un capteur capacitif

### e) Le capteur du champ électrique :

Le capteur de champ-électrique fonctionne avec un champ-électrique et le mesure au-delà de la couche extérieure de la peau où l'empreinte digitale commence. La technologie de champ-électrique peut être utilisée dans des conditions extrêmes, c'est-à-dire même si le doigt est sale ou sec. [4]



**Figure 7 : un capteur du champ électrique de type Authentec AES4000**

### f) Le capteur de pression :

Le principe du capteur de pression réside dans le fait que lorsqu'un doigt est placé au-dessus de la zone du capteur, seuls les bosses de l'empreinte digitale entrent en contact avec les rayons piezo du capteur. Les creux en contraste n'ont aucun contact avec les cellules du capteur. [4]

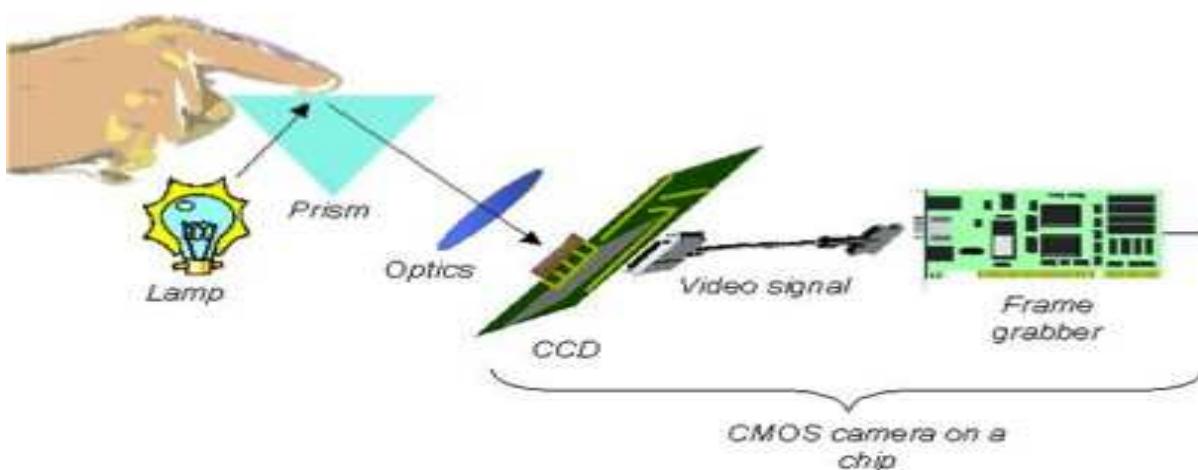


**Figure 8 : Le capteur de pression**

### g) Le capteur optique :

Ces capteurs s'assimilent à des minis-caméras : Ils sont composés d'un appareil-photo DTC (dispositif à transfert de charge) (CCD en anglais). Le doigt est apposé sur une platine en plastique dur ou en quartz, qui est en vis-à-vis de la mini caméra. Il résiste très bien aux fluctuations de température, mais est gêné par une lumière ambiante trop forte.

De plus il est assez volumineux. Son coût est intéressant, et il est intrinsèquement protégé contre les décharges électrostatiques. Il permet d'avoir des images précises et nettes.



**Figure 9 : le principe de fonctionnement du capteur optique**

La majorité des capteurs d'empreintes digitales optiques exploitent la modification de l'indice de réflexion de la surface d'un prisme lorsque les reliefs du doigt sont en contact avec cette dernière. Le principe de fonctionnement de ce type de capteur est représenté de manière simplifiée sur la Figure 9. [4]

## La carte microcontrôleur (Arduino Uno R3)

### I. INTRODUCTION

L' "Arduino" est une carte basée sur les microcontrôleurs. Il y en a plus de types d' "Arduino" comme Chaque type a ses caractéristiques, et ses utilités, qui ont plusieurs avantages :

- **Pas cher.**
- **Matériel Open source.**

**C'est-à-dire : que tous les personnes qui utilise ces cartes ont la possibilité de les modifier et de les contrôlées soulons leurs besoins.**

- **Logiciel Open Source.**

**C'est-à-dire :qu'on peut développer le langage d' "Arduino".**

- **Un environnement de programmation clair et aisés à l'utilisation.**

Dans notre projet on s'intéresse à la carte "Arduino uno".

Un microcontrôleur est un circuit intégré qui rassemble les éléments essentiels d'un ordinateur:

- Processeur.
- Mémoires (mémoire morte pour le programme, mémoire vive pour les données).

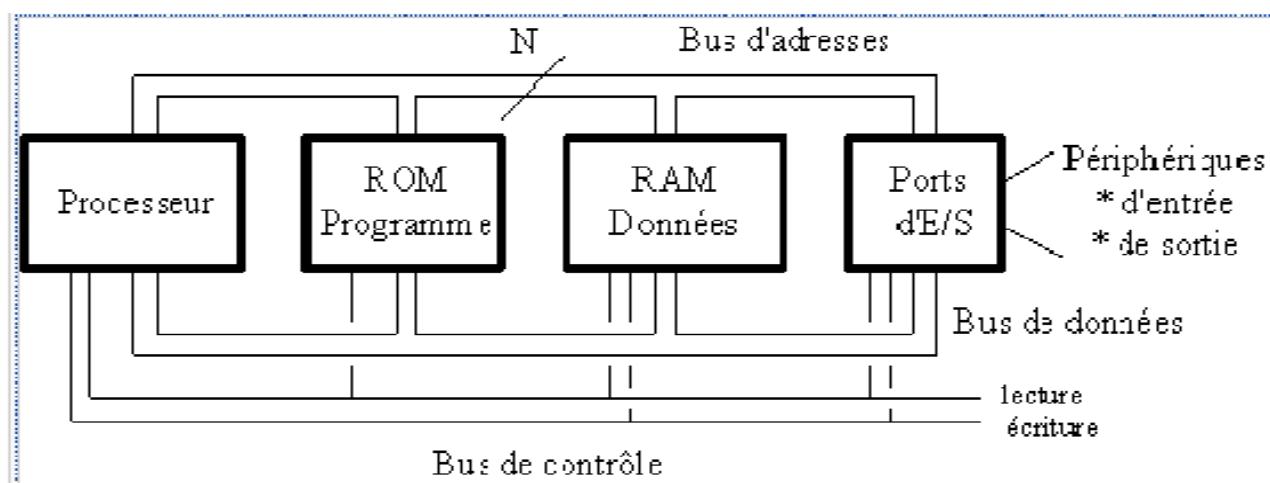


Figure 10 : Structure d'un système à microprocesseur

Les microcontrôleurs se caractérisent par un plus haut degré d'intégration, une plus faible consommation électrique, une vitesse de fonctionnement plus faible (de quelques mégahertz jusqu'à plus d'un gigahertz).

Les microcontrôleurs sont fréquemment utilisés dans les systèmes embarqués, comme les contrôleurs des moteurs automobiles, les télécommandes, les appareils de bureau, l'électroménager, les jouets, la téléphonie mobile, etc.

## II. DESCRIPTION DE LA CARTE "ARDUINO UNO R3 "(PARTIE MATERIEL)

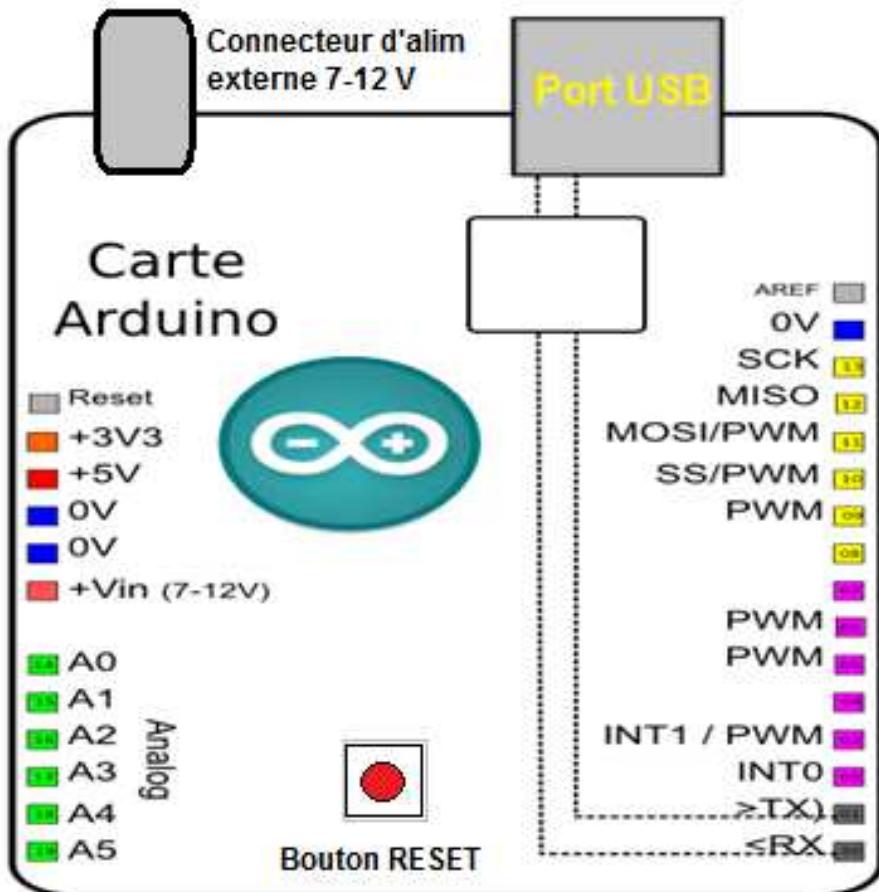
### 1. Le Microcontrôleur « ATMEGA 328 »

La carte "Arduino Uno R3" utilise comme mémoire l'ATmega 328 [9] contient :

- La mémoire **FLASH** contient 32 Ko, c'est vrai que ce nombre est très petit mais suffisant pour écrire plus de choses pour programmer la carte.
- La partie (**BOOT LOADER**) a presque de 0.5 Ko, la partie responsable de la programmation de la carte.
- La mémoire **SRAM** contient 2 Ko.
- La mémoire **EEPROM** contient 1 Ko.

## 2. Les Broches de la carte

Chaque carte arduino caractérise par un nombre des entrées et de sorties (LES BROCHES).



**Figure 11 : les broches de la carte "Arduino uno R3"**

- **Les broches numérotées de 0 à 13 (les broches jaunes, roses, grises) :** sont soit des entrées soit des sorties digitales, dont 6 disposent en sorties PWM (largeur d'impulsion modulée).  
(RX) pour recevoir et (TX) pour transmettre.
- **AREF (grise) :** Tension de référence utilisée pour les entrées analogiques qui est différent du 5V.
- **les 6 broches d'entrées analogiques(en vert) :** peuvent être utilisé comme des entrées ou des sorties numériques.
- **GND (bleu) :** la Masse.
- **VIN (7V <--- >12V) :** sont utilisée pour alimenter la carte.

- **5V "tension régulée"(en rouge)** : qui est obtenus d'une tension provenir de la tension d'alimentation VIN à travers le régulateur qui la transforme en une tension stable (régulée).
- **3.3V (en rouge)** : source d'alimentation pour des certains circuits externes qui nécessitant cette tension au lieu du 5V, avec une courant maximale de 50mA.
- **RESET (broche grise)** : on utilise le niveau bas de 'reset' pour redémarrer le microcontrôleur, mais en pratique il est utilisé pour ajouter un bouton de réinitialisation.
- **port(USB) (en blanc)** : pour faire connecter la carte a l'ordinateur.
- **bouton (RESET)** : pour la réinitialisation.

### III. PROGRAMMATION DE L' "ARDUINO" (PARTIE LANGAGE)

Le fonctionnement des cartes "Arduino" il nécessite une programme selon notre besoin, Le langage "Arduino" est basé sur les langages C et C++.

La méthode d'enregistrer le programme :[5]

#### 1) Téléchargement du logiciel

##### Sur Windows

- Télécharger la version Windows du logiciel "Arduino" sur: [10] (50 Mo environ)
- Installer le logiciel.
- Dé zipper le pilote FTDI USB Drivers.zip.
- Brancher l' "Arduino" et pointer l'installeur Windows vers le pilote.
- Et voilà ! la carte est prête à accueillir un programme Utilisateur.

## 2) Désigner le bon port Série (USB-Série)

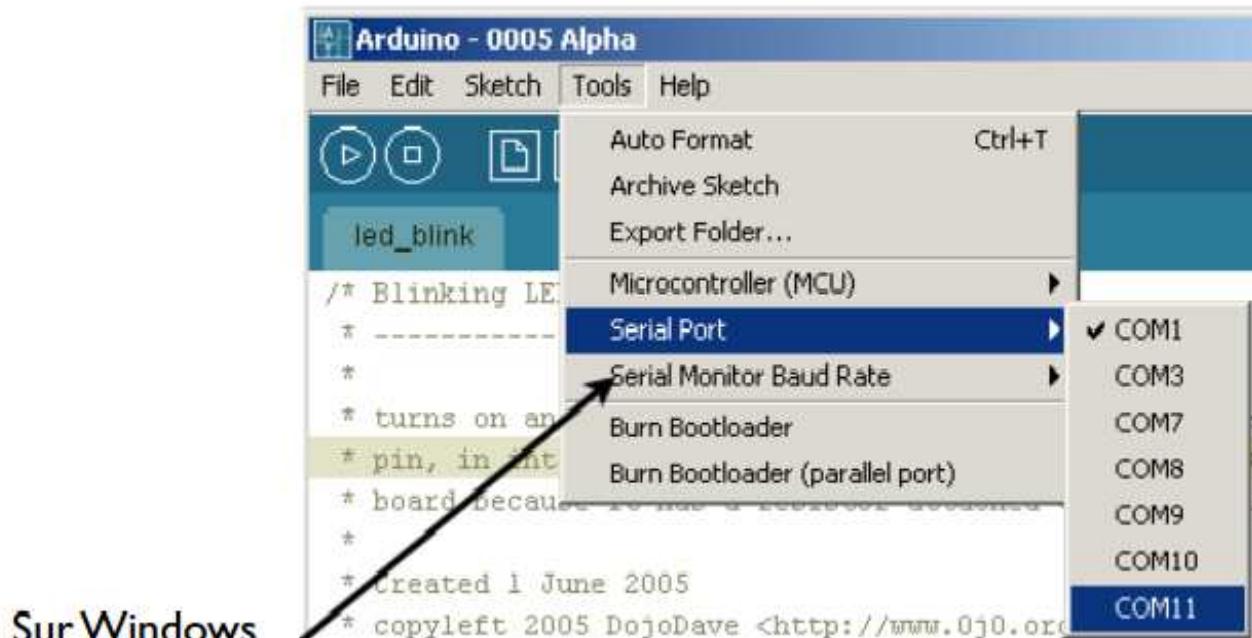


Figure 12

Description de la fenêtre du logiciel :



Figure 13 : La fenêtre de logiciel de la programmation d' "Arduino "

Alors pour cette raison, les électroniciens sont développées des programmes aider à communiquer avec la carte, Les programmes "Arduino" peuvent être divisés en trois partie principales: la structure, les valeurs (variables et constantes) et les fonctions.

## 1. Structure

Pour tous les programmes, il est nécessaire de poser les deux fonctions de base suivant :

- ❖ **void setup ()** : la fonction d'initialisation et de configuration de l'état des broches (Entrées ou sorties) on utilisant la fonction pinMode (nom, état).  
Exemple : pinMode (12, output)
- ❖ **Void loop ()** : dans laquelle on mit nos instructions.

➤ **Les structure de contrôle:** if /if...else /for /while / do ....wihle /continue /break...

➤ **Les opérateurs arithmétiques:** +,= , / , \* , % , - .

➤ **Opérateurs de comparaison :**

- ❖ == (Egal à)
- ❖ != (Différent de)
- ❖ < (Inférieur à)
- ❖ (Supérieur à) ;
- ❖ <= (Inférieur ou égal à)
- ❖ >= (Supérieur ou égal à)

➤ **Opérateurs composés :**

- ❖ ++ (incrémentation)
- ❖ -- (décrémentation)
- ❖ += (addition composée)
- ❖ -= (soustraction composée)
- ❖ \*= (multiplication composée)
- ❖ /= (division composée).

## 2. Variables et constantes :

Dans le programme d' "Arduino" il y a des **Constantes prédefinies** :

- ❖ **HIGH / LOW** : (haut/bas)
- ❖ **INPUT / OUTPUT** :( entrée/sortie)
- ❖ **TRUE / FALSE** :(vrai/Faux)
- ❖ **constantes décimales**

### ➤ *Types des données :*

- ❖ char (caractère)
- ❖ int (entier)
- ❖ long (réel long)
- ❖ void (fonctions)
- ❖ Les tableaux de variables
- ❖ float (réel).....

### ➤ *Conversion des types de données :* ces fonctions sont utilisées pour convertir vers un nombre réel, entier, réel long ou vers un caractère....

- ❖ char ()
- ❖ byte ()
- ❖ int ()
- ❖ long ()
- ❖ float ()

## 3. Fonctions :

### ➤ **Entrées ou Sorties Numériques :**

- ❖ **pinMode (broche, état)** : écrire un état sur une broche.
- ❖ **digitalWrite (broche, valeur)** : écrire un état sur une broche numérique.
- ❖ **digitalRead (broche)** : lire un état sur une broche numérique.
- ❖ **unsigned long pulseIn(broche, état)** : lire une impulsion sur une broche numérique.

### ➤ **Entrées analogiques :**

- ❖ **analogRead (broche)** : lire la valeur d'une broche ana.
- ❖ **analogWrite(broche, valeur)** : écrire une valeur analogique sur les broches 9, 10 ou 11 (PWM ).

➤ **Gestion du temps :**

- ❖ **unsigned long millis ()** : temps de fonctionnement du programme.
- ❖ **delay (ms)** : attente, en millisecondes.
- ❖ **delayMicroseconds (us)** : attente, en microsecondes.[9]

**4. Bibliothèques :**

Les librairies peuvent être incluses dans notre programmes à l'aide de l'instruction " #include " et on peut utiliser des fonctions élaborées avec l'carte "Arduino" (communication série par exemple). Les librairies nous permettent surtout d'interfacer et d'utiliser notre carte "Arduino" avec toutes sortes de matériel (afficheur LCD, clavier matriciel, etc.).

On dispose de plusieurs bibliothèques, nous nous limiterons à La librairie SoftwareSerial:

La librairie de communication série présentée ici a été développée pour permettre des communications série sur d'autres broches numériques de la carte "Arduino", en utilisant un programme (le code de la librairie) pour répliquer la même fonction de communication série (d'où le nom de Software Serial pour communication série logicielle).

➤ **Limitations**

En raison du fait qu'elle n'est pas basée sur un module matériel, cette librairie a quelques limitations :

- ❖ **seules les vitesses jusqu'à 9600 (bit/seconde).**
- ❖ **l'instruction Serial.read : attendre jusqu'à ces toutes les données arrive**
- ❖ **Seules les données reçues tant que l'instruction Serial.read() est active seront reçues. Les données reçues à d'autres moments ne seront pas prises en compte.**[1]

➤ **Les fonctions de la librairie:**

- ❖ **SoftwareSerial(rx Broche, tx Broche)**
- ❖ **begin():**Fixer la vitesse pour la communication série
- ❖ **read():**Lire un caractère sur la broche en réception sur le port série logiciel
- ❖ **print():**Afficher dans une fenêtre Terminal les données émises par la broche d'émission du port série logiciel.
- ❖ **println():**Afficher dans une fenêtre Terminal les données émises par la broche d'émission du port série logiciel, suivi d'un saut de ligne.[1]

## La partie de la réalisation

### I. INTRODUCTION

Au début, on mentionne que le capteur utilisé est SEN04172P

Le Capteur d'empreinte digitale est un capteur optique basé sur le principe décrit à la page 6. Il rendra la détection d'empreintes digitales et de vérification ajoutant. Nous pouvons enregistrer jusqu'à 162 empreintes digitales. Ces empreintes seront stockées sous forme digitale dans la mémoire flash embarquée. Il ya une LED rouge dans la lentille qui s'allume durant la prise de photo.

#### 1. Spécification :

- ❖ **Type de module :** SEN04172P
- ❖ **Tension d'alimentation :** 3,6 ~ 6,0 V
- ❖ **Courant d'emploi (Max):** 120 mA
- ❖ **Capacité de stockage:** 162 modèles
- ❖ **Vitesse de transmission:** 9600, 19200, 28800, 38400, 57600(bit/seconde)  
(Valeur par défaut est 57600)
- ❖ **Température de travail:** -20 ~ 50 °C. [6]



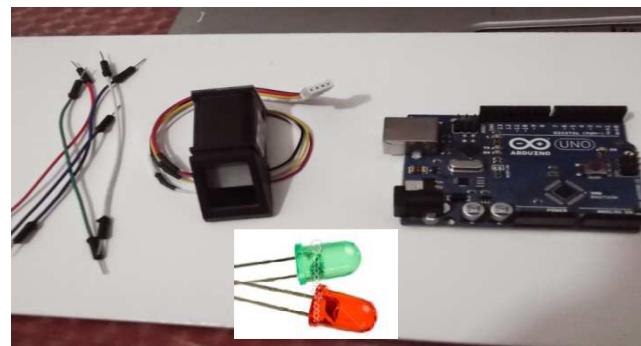
**Figure 14 : le Capteur d'empreinte digitale SEN04172P**

Ce capteur vient généralement avec un câble spécial pour le connecter, ses couleurs sont :

- ❖ **Rouge:** alimentation 5V
- ❖ **Noir:** masse (GND)
- ❖ **Blanc:** fil Rx (récepteur)
- ❖ **Vert "Parfois Jaune":** fil Tx (Transmetteur)

#### 2. Les pièces utilisées :

- ❖ carte "Arduino Uno"
- ❖ capteur d'empreinte digitale
- ❖ fils des cavaliers
- ❖ Deux LED (verte et rouge)



**Figure 15: Les pièces utilisées**

## II. PROGRAMMATION DE L'EMPREINTE

En va télécharger le logiciel "SFGDemo" qui sert à stocker les empreintes la mémoire du capteur. [8]

### 1. Première étape

Nous allons d'abord stocker les empreintes digitales que nous voulons pour les identifier à l'avance, et cela se fait à travers les étapes suivantes:

- ❖ maintenir un programme vide "Blink" sur "Arduino"

```
//Blink code
//Code for finger print sensor
voidsetup() { }

voidloop() { }
```

- ❖ Nous relions directement le capteur à (Tx/D0, Rx/D1) comme dans la figure 16 :

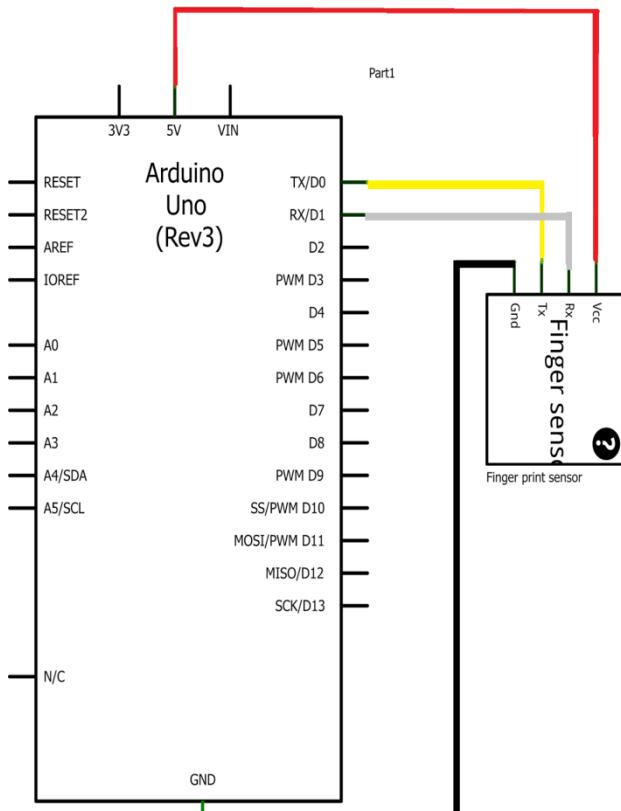


Figure 16

- ❖ Démarrez le logiciel SFGDemo et cliquez sur Ouvrir le périphérique dans le coin inférieur gauche. Sélectionnez le port COM utilisé par l' "Arduino". Nous sélectionnons le numéro de port correct comme dans la figure 17 :

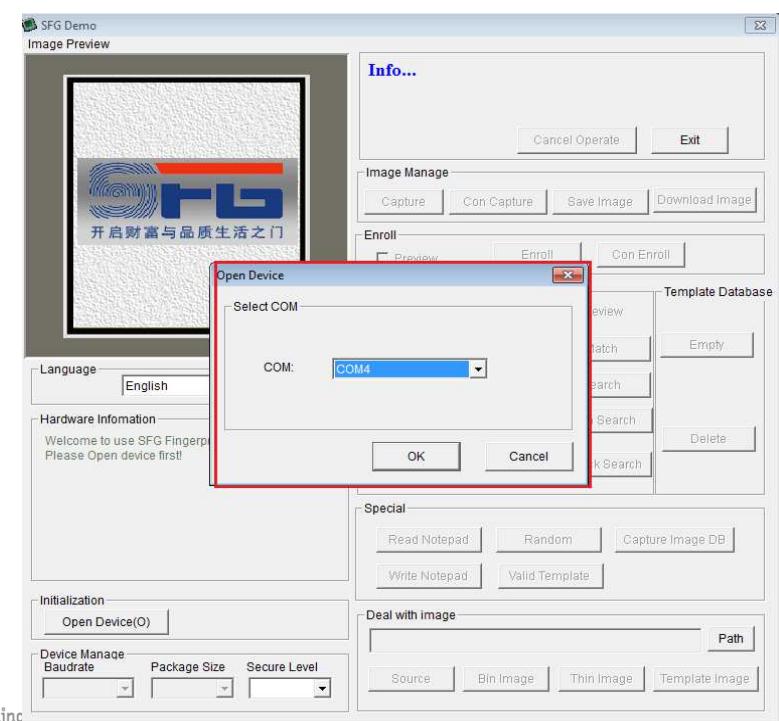


Figure 17

Après on appuyant sur "OK", le logiciel "SFGDemo" donnera des données et des informations sur le capteur, et même les empreintes digitales stockées précédemment " si déjà ajouté " apparaît sur l'écran (figure 18).

Pour l'instant, nous allons ajouter une nouvelle empreinte que le capteur la reconnaîtra plus tard.

Nous allons appuyant sur ' Enroll 'pour montrer notre écran, cet écran nous demande de localiser le lieu de stockage de l'empreinte "de 0-162 empreinte"

Après avoir localisé le stockage de l'empreinte, on appuie sur "OK", puis mettre le doigt sur le capteur, un couleur rouge apparaîtra sur le capteur, gardez votre main sur le capteur et un message

apparaît sur l'écran, vous guide à travers les étapes.

Une méthode rapide pour nous assurons que notre empreinte est bien stockée. On appuie sur "search"(figure 19), puis ont mis le doigt sur le capteur. Le logiciel nous donne si empreintes digitales existe et où il est stocké.

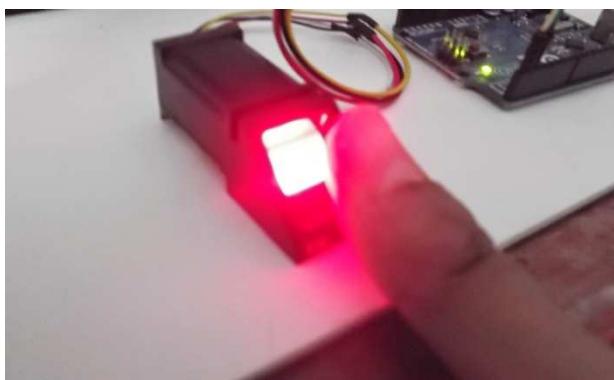


Figure 19

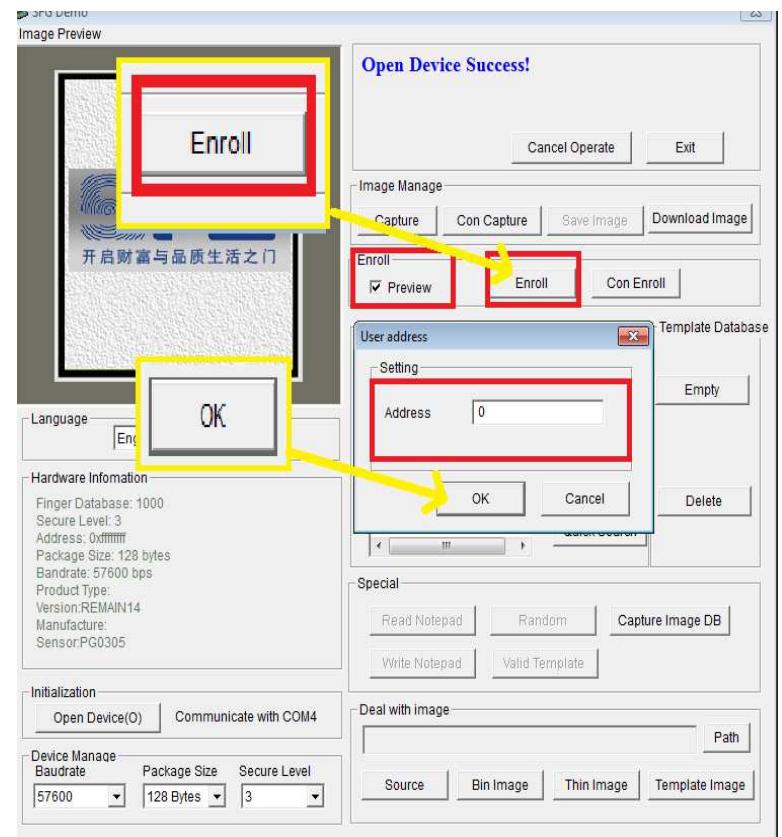


Figure 18

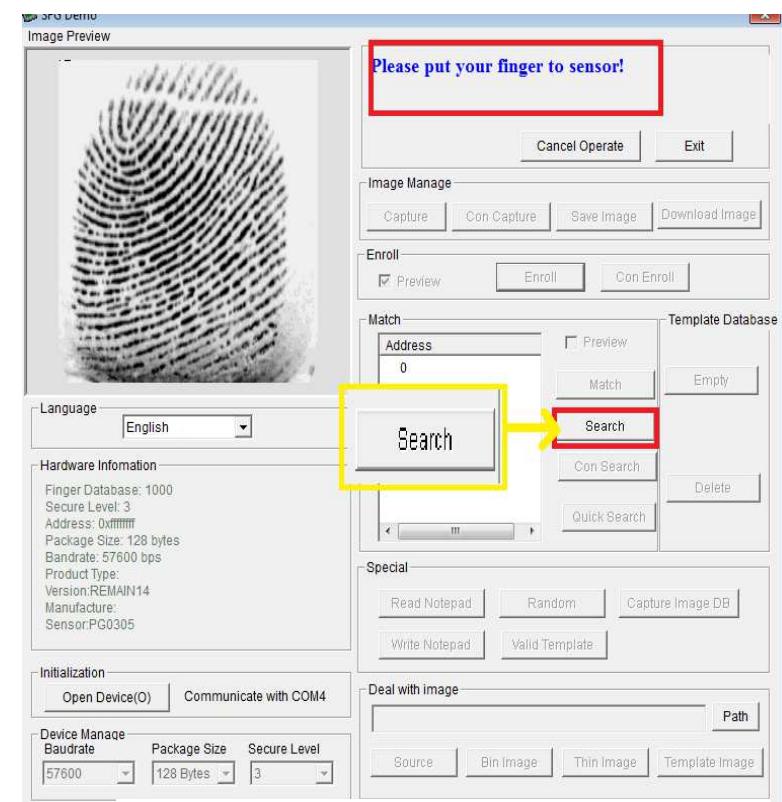


Figure 20

## 2. La deuxième étape:

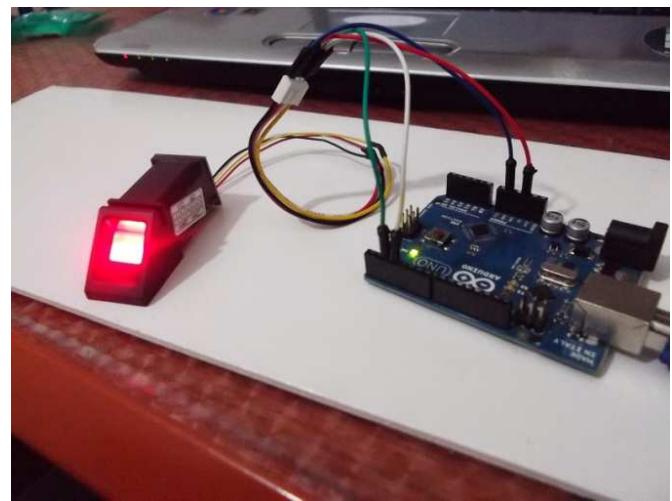
On relie le capteur directement avec "Arduino" (figure 21).

Ensuite, on compare les empreintes digitales qui sont relevées par ce capteur. Ces dernières utilisées pour faire des emplois différents, exemple:

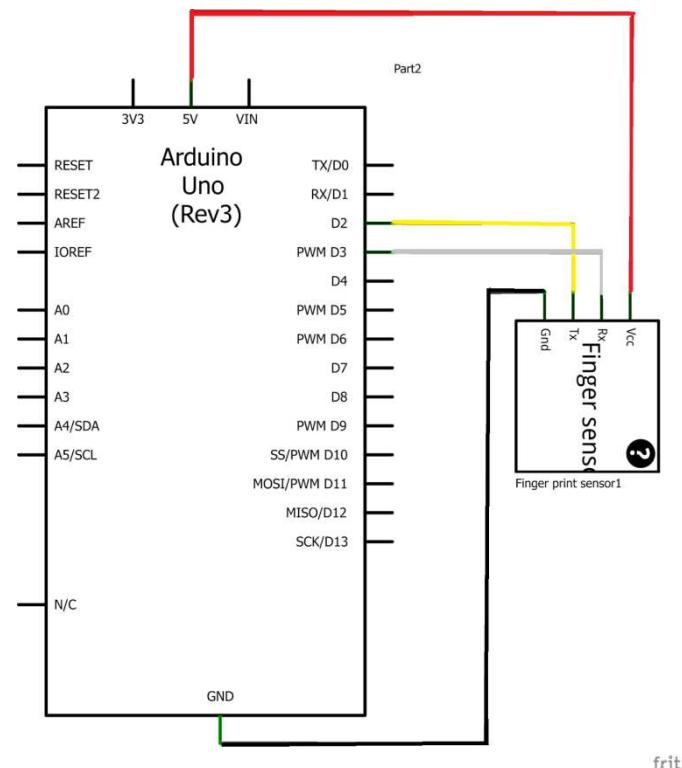
"Moteur tournant, contrôle d'accès d'une base de données, permettant de recevoir des ordres de cette personne ... etc."

La connexion va changer comme dans la figure 22.

Nous aurons besoin de télécharger la bibliothèque "Adafruit-Fingerprint-Sensor-Library-master", et placé dans le chemin des bibliothèques dans "Arduino". [7]



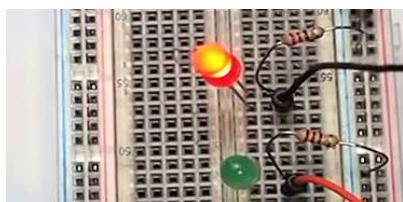
**Figure 21:** le capteur d'empreinte digitale relier avec l'arduino



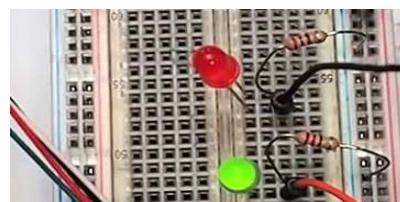
fritzing

**Figure 22**

Avant d'exécuter le programme écrit au dessous, on va brancher les deux Led, le rouge pour montrer que le système est fermé et l'autre (verte) pour montrer que le système est ouvert.

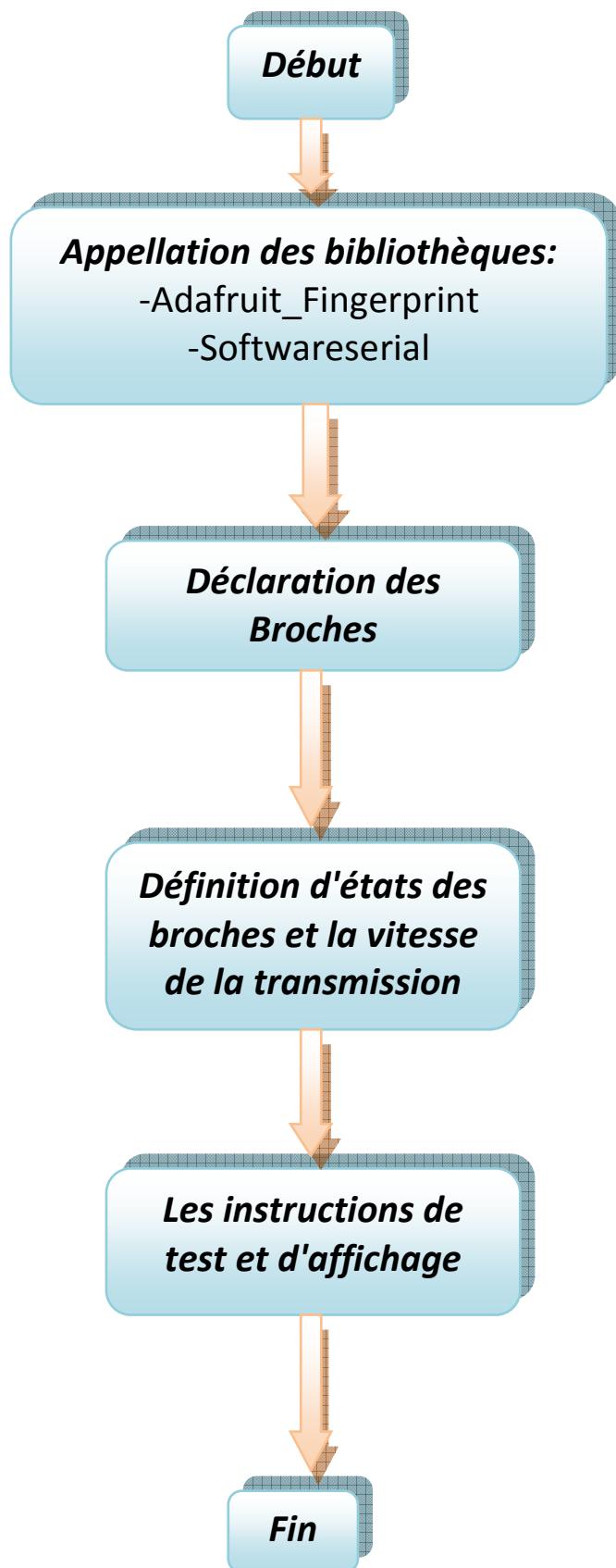


**Figure 23**



**Figure 24**

### L'organigramme du programme du projet



## Le programme utilisé dans ce Projet

```

1) #include <Adafruit_Fingerprint.h>
2) #include <SoftwareSerial.h>
3) int getFingerprintIDez();
4) // pin #2 is IN from sensor (GREEN wire)
5) // pin #3 is OUT from arduino (WHITE wire)
6) SoftwareSerial mySerial(2, 3);
7) Adafruit_Fingerprint finger = Adafruit_Fingerprint(&mySerial);
8) void setup()
9) {
10) Serial.begin(9600);
11) Serial.println("fingertest");
12) pinMode(11,OUTPUT);
13) pinMode(12,OUTPUT);
14) // set the data rate for the sensor serial
     port
15) finger.begin(57600);
16) if (finger.verifyPassword()) {
17) Serial.println("Found fingerprint sensor!");
18) } else {
19) Serial.println("Did not find fingerprint sensor :(");
20) while (1);
21) }
22) Serial.println("Waiting for valid finger...\"");
23) }

24) void loop()           // run over and over
   again
25) {
26) getFingerprintIDez();
27) delay(50);          //don't ned to run this at
   full speed.
28) digitalWrite(12,HIGH);
29) }
30)
31) // returns -
   1 if failed, otherwise returns ID #
32) int getFingerprintIDez() {
33) uint8_t p = finger.getImage();
34) if (p != FINGERPRINT_OK) return -1;

35) p = finger.image2Tz();
36) if (p != FINGERPRINT_OK) return -1;

37) p = finger.fingerFastSearch();
38) if (p != FINGERPRINT_OK) return -1;

39) // found a match!
40) digitalWrite(12,LOW);
41) digitalWrite(11,HIGH);
42) delay(4000);
43) digitalWrite(11,LOW);
44) delay(500);
45) digitalWrite(12,HIGH);

46) Serial.print("Found ID #"); Serial.print(finger.fingerID);
47) Serial.print(" with confidence of "); Serial.println(finger.confidence);
48) return finger.fingerID;
}

```

## Conclusion

Au cours de ce projet on a vu que la sécurité de l'accès à les sociétés, entreprises présente un vrai problème. Il nous obligera de trouver des moyens pour sécuriser ces endroits.

Depuis plusieurs années, plusieurs techniques sont élaborées pour contrôler l'accès aux sociétés, parmi ces techniques :

- **L'accès par un code ou un mot de passe.**
- **L'accès par un badge ou des cartes puce.**
- **L'accès biométriques (Empreinte digitale, L'œil, Visage ...).**

La biométrie par l'empreinte digitale est la technologie la plus employée à travers le monde.

Et on voit fleurir des solutions de plus en plus abordables et performantes.

Pour réaliser cette technique on ait utilisé une carte microcontrôleur "Arduino" qui possède un espace de programmation qu'est très claire et simple. Cette carte sert à contrôler un capteur d'empreinte digitale. Ce dernier nous aide pour stocker des empreintes à l'aide du logiciel "SFGDemo" qu'on va utiliser après.

## LES ABRIVIATIONS

**Caméra DTC(ou CCD) :** (dispositif à transfert de charge)

Le fonctionnement d'un détecteur CCD(ou) peut être ainsi résumé :

- Chaque pixel de la matrice CCD correspond à un élément semi-conducteur en sandwich dans un condensateur électrique.
- Un photon incident crée un photoélectron, lorsqu'il apporte à un électron du matériau semi-conducteur l'énergie nécessaire pour franchir le seuil énergétique (gap).
- Les photoélectrons sont stockés dans le puits de potentiel qu'est le pixel convenablement polarisé.
- La lecture de ces photoélectrons est commandée par polarisation via des transistors à effet de champ. Elle a lieu soit directement, un obturateur cachant la source, soit par transfert de trame. Dans ce dernier cas, une moitié de la surface du CCD est réservée à la collecte du signal; l'autre moitié n'est jamais éclairée, mais recueille les photoélectrons de la partie réceptrice, avant la lecture complète et le transfert des charges vers l'étage d'amplification.

**SRAM :** (**S**tatic **R**andom **A**ccess **M**emory) est un type de mémoire vive utilisant des bascules pour mémoriser les données.

**EEPROM :** (**E**lectrically-**E**rasable **P**rogrammable **R**ead-**O**nly **M**emory)(ou mémoire morte effaçable électriquement et programmable) est un type de mémoire morte.

## REFERENCES

- [1] [http://www.mon-club-elec.fr/pmwiki\\_reference\\_arduino/pmwiki.php?n>Main.Librairies](http://www.mon-club-elec.fr/pmwiki_reference_arduino/pmwiki.php?n>Main.Librairies)
- [2] <http://policescientifique-tpe.e-monsite.com/pages/3-les-empreintes-digitales.html>
- [3] <http://biometrie-tpe68.e-monsite.com/pages/introduction/les-differentes-familles-de-capteurs.html>
- [4] <http://www.biometrie-online.net/technologies/empreintes-digitales>
- [5] <http://www.craslab.org/interaction/files/LivretArduinoCRAS.pdf>
- [6] <http://www.adafruit.com/products/751>
- [7] [http://www.genotronex.com/2014/04/blog-post\\_25.html](http://www.genotronex.com/2014/04/blog-post_25.html)
- [8] <http://www.adafruit.com/datasheets/SFGDemoV2.0.rar>
- [9] <http://arduino.cc/>
- [10] <http://www.arduino.cc/en/Main/Software>