الجمهورية الجزائرية الديمقراطية الشعبية
République Algérienne Démocratique et Populaire
وزارة التـــعليـــــــم العـــــالـــــــــي والبـــحـــث العلـمي
Ministère de l'Enseignement Supérieure et de la Recherche Scientifique

جامعة وهران2 محمد بن احمد
Universitéd'Oran2 Mohamed Ben Ahmed

معـــهد الصيانـــة والامـــن الصناعـــي
Institut de Maintenance et de Sécurité industrielle

Département de Maintenance en instrumentation

Mémoire

En vue de l'obtention du diplôme de

Master

Filière : *Génie industrielle*

Spécialité *: Génie industrielle*

Thème :

Conception et développement d'un Système d'identification et de vérification d'individus

Présenté et soutenu publiquement par :

Regabna Sarrah

Mehdi Khayra

Devant la commission du jury composée de : 26/11/2020

Nom et prénom	***Grade***	***Etablissement***	***Qualité***
Mr.GHOUARI Adel Benmohamed	*MCB*	*IMSI/Univ Oran 2*	***Président***
Mr.CHENNOUFI Mohammed	*MCB*	*IMSI/Univ Oran 2*	***Encadreur***
Mr.LALAOUI Med Amine	*MAA*	*IMSI/Univ Oran 2*	***Examinateur***

2019/2020

Remerciement

Nous tenons tout d'abord à remercier ***Allah*** le tout puissant et miséricordieux, qui nous a donné la force et la patience d'accomplir ce Modeste travail.

En second lieu, nous tenons remercier très chaleureusement *Mr.Chennoufi Mohamed,* Directeur adjoint chargé des études qui nous a permis de bénéficier de son encadrement. Les conseils qu'il nous a prodigué, la patience, la confiance qu'il nous a témoignés ont été déterminants dans la réalisation de notre travail.

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

Nous souhaitant adresser nos remerciements les plus sincères aux personnes qui nous ont apporté leur aide et qui ont contribué à l'élaboration de ce mémoire.

Un clin d'œil à nos familles et nos amis qui par leurs prières et leurs encouragements, on a pu surmonter tous les obstacles.

Enfin, nous tenons également à remercier toutes les personnes qui ont participé de près ou de loin à la réalisation de ce travail.

DEDICACES

A mes chers parents, pour tous leurs sacrifices, leur amour, leur tendresse, leur soutien et leurs prières tout au long de mes études.

A mes chers frères Arbi, Kaled, Iyad.

Mes chères sœurs Nada, Zina, Borouj.

A mon très cher fiancé Aboubakre tes sacrifices, ton soutien moral, ta gentillesse sans égal, ton profond attachement m'ont permis de réussir mes études. Sans ton aide, tes conseils et tes encouragements ce travail n'aurait vu le jour.

A Sarra, Zohra, Manar et Belkis.

A tous les êtres chers don leur soutien m'a été indispensable et qui attendant avec impatience ma réussie, en espérant être à la hauteur de leurs attentes.

Sarrah

Je dédie ce mémoire

A mes chers parents ma mère et mon père pour leur patience, leur amour, leur soutien et leurs encouragements.

À ma sœur Fatima, qui m'a soutenu dans ma carrière universitaire.

A mes frères Ahmed, Mohamed et Abdelkader.

À mon fiancé Mounir, qui a planté la volonté et la patience dans mon cœur.

Mes neveux Nourhane, Malak, Alouch et Hamza.

Sans oublier mes amies et tous les professeurs que ce soit du primaire, du moyen, du secondaire ou de l'enseignement supérieur.

Khayra

Table des matières

Chapitre II : Les empreintes digitales

Chapitre III : Conception et réalisation

Chapitre IV : Résultats expérimentaux

Liste des figures et des tableaux :

Les abréviations

ID Identité.

ADN Acide désoxyribonucléique.

FRR False Rejet Rate ou Taux de Faux Rejet.

FAR False Acceptation Rate ou Taux de Fausse Acceptation

CCD Dispositif Charge Couplé.

EEPROM Electrically Erasable Programmable Read-Only Memory (Mémoire de lecture seule programmable effaçable électriquement)

LED Light emitting diodes (diode électroluminescente).

CN nombre de connexion.

PIN pesonal identificatino number (numéro d'identification personnel).

USB universal serial bus.

IDE integrated développement environnement**.**

BDD base de données.

SRAM static random-access memory (mémoire statique à accès aléatoire).

Introduction générale

De nos jours on parle de plus en plus de l'insécurité dans divers secteurs ainsi que des moyens informatiques à mettre en œuvre pour contrer cette tendance. La vérification et l'identification des individus est l'un des moyens permettant d'assurer cette sécurité. L'être humain se sert quotidiennement de son système visuel pour identifier les personnes de façon automatique, bien que le processus mis en jeu soit complexe.

l'homme a mis en place des moyens de vérification d'identité qui sont liés, soit à ce que possède une personne telle qu'une carte d'identité ou un passeport, soit à ce que sait cette personne, c'est le cas du mot de passe ou un code PIN. Néanmoins, ces éléments peuvent être oubliés, volés ou falsifiés. Pour contourner ces limitations, un autre moyen de sécurité a été développé qui permet d'utiliser, non pas l'information qu'un individu possède ou connaît, mais une information (propre) intrinsèque à cette personne. Cette nouvelle façon d'identification des individus est ***la biométrie.***

Dans notre travail, nous avons choisi « l'empreinte digitale » comme modalité d'identification de par son ancienneté, sa mise à l'épreuve et son coût abordable.

Parmi toutes ces techniques, l'utilisation de l'empreinte digitale comme un moyen d'identification et d'authentification, est celle qui est la plus courante. La force de ce procédé tient du fait que l'utilisation de l'empreinte digitale est généralement plus facile d'acception par la communauté, et qu'elle est une des plus efficaces. La raison principale de l'utilisation d'empreinte dans le système identification ou vérification est que l'empreinte est unique et reste invariable avec l'âge.

Organisation du mémoire : Ce mémoire est structuré en quatre chapitres :

Dans le premier chapitre, nous traitons de notions fondamentales à l'effet d'introduire les principales définitions de la biométrie, les fonctionnements de ses systèmes et les mesures de leurs performances, et les techniques biométriques existantes.

Le deuxième chapitre mettra l'accent sur la modalité la plus ancienne et la plus mature d'identification biométrique qui est l'empreinte digitale, nous étudierons les propriétés de cette modalité ainsi que la structure complète d'un système de reconnaissance par empreinte digitales.

Dans le troisième chapitre, on présentera la conception et la réalisation de notre projet, un système de reconnaissance d'empreinte digitale par une carte Arduino et un capteur optique d'empreinte qui nous permet de créer une base de données spécialement pour nous modifiable et changeable selon notre désir.

Le quatrième chapitre présente les résultats expérimentaux réalisés sur une base de données standard. Nous montrons qu'un choix judicieux des caractéristiques permet une amélioration importante des performances du système.
Ce mémoire est terminé par une conclusion générale mettant en relief les résultats obtenus ainsi que des perspectives à réaliser à long terme.

Chapitre I

La technique de détection et vérification.

I.1. Introduction :

Plusieurs techniques biométriques ont été développées par la recherche scientifique pour identifier les personnes. Ces techniques, généralement appelées *méthodes biométriques*, ont donné naissance à ce qu'on appelle le mot de passe biométrique. L'avantage principal d'un mot de passe biométrique est qu'il ne pourrait pas être volé, oublié ou transmis à une autre personne. En effet, chaque individu possède son propre stique biométrique, permettant de l'identifier. Par conséquent et dans un futur relativement proche, le mot de passe biométrique remplacera le mot de passe conventionnel dans toutes les applications nécessitant un niveau élevé de sécurité.

I.2. Les différentes techniques de la biométrie :

Les techniques biométriques se divisent en deux groupes selon la coopération ou non de l'individu :

. Techniques intrusives : Ces techniques requièrent un contact physique avec l'individu pour l'identifier, tel que les empreintes digitales, la rétine, l'ris ou la forme de la main. Leur usage est généralement mal accepté.

. Techniques non intrusives : Ces techniques ne requièrent pas la coopération de l'individu en question. Leur application peut se faire à distance en utilisant des capteurs qui ne nécessitent pas de contact directe avec l'utilisateur (visage, démarche,…).La biométrie permet l'identification ou l'authentification d'une personne sur les bases de données reconnaissables et vérifiables qui lui sont propres.

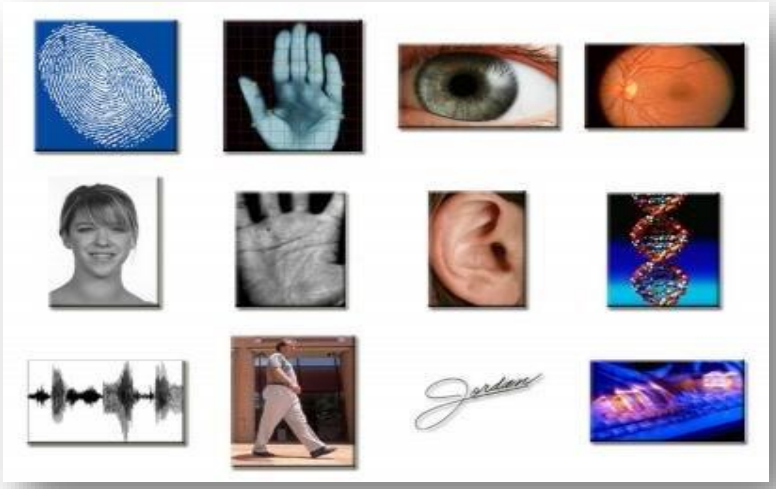


Figure I.1 : Différentes modalités biométriques [1]

On peut classer ***les techniques biométriques*** en trois catégories :

1- Celles basées sur l'analyse de traces biologiques : ce type de biométrie se fait à l'aide de l'ADN d'une personne, de son sang, ou de sa salive…

2- Celles basées sur l'analyse comportementale : se base sur l'analyse de certains comportements d'une personne comme le tracé de sa signature, l'empreinte de sa voix, sa démarche et sa façon de taper sur le clavier.

3- Celles basées sur l'analyse morphologique : est basée sur l'identification de traits physiques particuliers qui, pour toute personne, sont uniques et permanents. Cette catégorie regroupe la reconnaissance de la forme du visage, de la forme de la main, des empreintes digitales, de la rétine et de l'iris de l'œil.

Ces éléments ont l'avantage d'être stables dans la vie d'un individu et ne subissent pas autant les effets du stress par exemple, que l'on retrouve dans l'identification comportementale.

Toutefois, dans un système biométrique pratique (à savoir, un système qui utilise la biométrie à des fins personnelles de reconnaissance), il y a un certain nombre d'autres questions qui devraient être considérées, y compris:

-La performance, qui se réfère à la précision de la reconnaissance et de la vitesse possible ainsi que les ressources nécessaires pour obtenir cette précision de la reconnaissance et de la vitesse désirées. La performance se réfère également au fonctionnement et aux facteurs environnementaux qui influent sur la précision et la vitesse.

- ***L'Acceptabilité,*** qui indique la mesure dans laquelle les gens sont prêts à accepter l'utilisation notamment d'un identifiant biométrique (caractéristique) dans leur vie quotidienne.

- ***Le contournement,*** ce qui reflète la façon dont le système peut facilement être dupe en utilisant des méthodes frauduleuses.

Les technologies les plus fréquemment utilisées sont les suivantes :

I.2.1. Analyses biologiques :

I.2.1.1. L'odeur corporelle : Chaque personne dégage une odeur qui lui est particulière.

Les systèmes biométriques qui exploitent cette technologie analysent les composantes chimiques contenues dans l'odeur pour ensuite les transformer en données comparatives.

I.2.1.2. L'A.D.N. (Support matériel de l'hérédité):

Présent dans les cellules du corps, il est spécifique d'un individu à un autre et permet de l'identifier de manière certaine à partir d'un simple fragment de peau, d'une trace de sang ou d'une goutte de salive.

Actuellement, le temps requis pour une analyse et le coût associé à celle-ci restreignent son utilisation dans des domaines autres que celui de l'identification judiciaire. Cependant, ce procédé biométrique fait l'objet de recherche intensive puisqu'il représente la technologie d'identification par excellence avec une marge d'erreur bien en dessous des autres moyens biométriques.

I.2.1.3. La reconnaissance de la thermographie faciale :

Une caméra infrarouge capte la chaleur émise par la peau. Contrairement à la reconnaissance faciale, on peut donc l'utiliser même dans l'obscurité ou de mauvaises conditions de visibilité. Mais les conditions de prise de vue peuvent conduire à des erreurs.

I.2.2. Analyses morphologiques :

I.2.2.1. Les empreintes digitales :

Un système biométrique utilisant l'empreinte digitale comme moyen d'identification ou de vérification ne procède pas de la même façon, ce n'est pas l'image de l'empreinte digitale qui sert de point de comparaison, mais l'ensemble des données biométriques qui est tiré à partir des minuties de l'empreinte digitale. Les minuties représentent les fins de crêtes, les bifurcations, les lacs, les Lots et les points qui composent l'empreinte digitale. La combinaison des minuties

est quasi infinie. L'acquisition des données est faite par un capteur électronique de type optique, thermique, capacitif ou à ultrasons. Cette dernière est considérée comme la plus fiable, mais aussi la plus coûteuse (voir figure I.2).

Figure I.2 : La reconnaissance de l'empreinte digitale [2]

I.2.2.2. La Géométrie de la main :

La reconnaissance de la forme de la main est considérée comme l'ancêtre des technologies biométriques. A la fin des années soixante, *Robert P. Miller* déposa un brevet pour un appareil permettant de mesurer des caractéristiques de la main et de les enregistrer pour comparaison ultérieure, l'utilisateur place sa main sur un gabarit. Le tout est éclairé par une lumière infrarouge et l'image est captée par une caméra digitale. Près d'une centaine de caractéristiques sont extirpées de l'image et converties en données stockées en mémoire, lors de la phase d'enrôlement ou comparées lors de la phase d'identification. Ces données concernent la longueur, la largeur et l'épaisseur de la main, de même que la forme des articulations et la longueur inter articulations.

Figure I.3 : La reconnaissance de la main [3]

I.2.2.3. La reconnaissance de l'iris:

C'est une technologie fiable ; et semble être beaucoup plus précise que certains autres moyens biométriques. Ceci s'explique par le fait que notre iris comporte énormément de caractéristiques pouvant varier d'un individu à l'autre. L'iris se compose de vaisseaux sanguins et ceux-ci sont disposés différemment d'un individu à un autre. Chaque œil est unique. Il est prouvé que la probabilité de trouver deux iris identiques est inférieure à l'inverse du nombre d'humains ayant vécu sur terre.

Une fois que l'image de la configuration des vaisseaux sanguins est obtenue par le système biométrique, le fonctionnement est quasi identique à celui du système analysant l'empreinte digitale. La grosseur des vaisseaux, leur positionnement et les bifurcations qui les caractérisent font partie des éléments, les minuties, qui seront étudiés par le système dans le but d'en dégager un algorithme particulier. La comparaison avec le fichier référence pourra s'ensuivre. Le point faible de ce type de système utilisant l'œil à des fins d'identification ou de vérification est qu'il éprouve beaucoup de difficultés à lire l'image de l'œil d'une personne aveugle ou d'un individu ayant un problème de cataracte.

Figure I.4: la reconnaissance de l'iris [4]

I.2.2.4. La reconnaissance de visage :

Le développement de systèmes biométriques basés sur la reconnaissance de la forme du visage est des plus récents. En 1982, deux chercheurs *Hay* et *Young* affirment que l'humain, pour reconnaître un visage, utilise les caractéristiques globales et locales qui le composent.

Des recherches plus avancées furent effectuées afin de voir si cette capacité de reconnaissance pouvait être reproduite informatiquement. C'est à partir des travaux du professeur *Teuvo Kohonen (1989),* chercheur en réseaux neuronaux de l'Université d'Helsinki, et des travaux de Kirby et Sirovich (1989) de l'Université Brown du Rhode Island, que fut mis au point un système de reconnaissance du visage nommé : **EIGENFACE[5]** (voir la figure I.5).

L'image du visage est captée par une caméra. Le sujet peut se présenter volontairement devant celle-ci ou encore, son image peut être capturée à son insu pour en dégager certaines particularités. Selon le système utilisé, l'individu doit être positionné devant l'appareil ou peut être en mouvement à une certaine distance. Les données biométriques qui sont obtenues sont par la suite comparées au fichier référence.

La reconnaissance de visage est basée sur des attributs faciaux mesurables comme l'écartement des yeux, des sourcils, des lèvres, la position du menton et la forme.

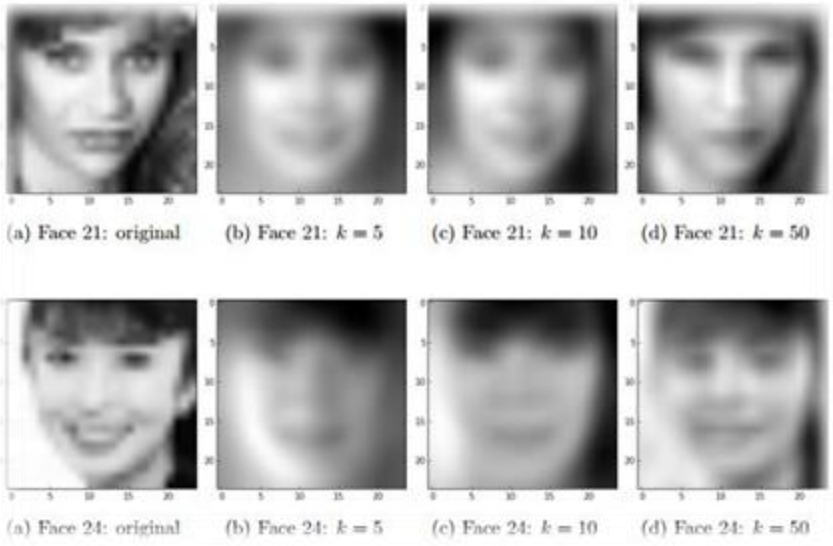


Figure I.5: la reconnaissance de visage [11].

I.2.2.5. La reconnaissance vocale :

C'est en 1962 que Lawrence Kersta, un ingénieur du bel laboratoire, établit que la voix de chaque personne est unique et qu'il est possible de la représenter graphiquement. La voix est constituée de composantes physiologiques et comportementales.

Dans les années 80, plusieurs entreprises développèrent des systèmes de reconnaissance de la voix pour les corps policiers et les agences d'espionnage. Au début des années 90, le gouvernement américain demanda à ces entreprises de mettre au point un système pour le marché commercial.

Initialement, une table de référence de la voix d'une personne doit être construite. Pour ce faire, celle-ci doit lire une série de phrases ou de mots à plusieurs reprises. Plusieurs caractéristiques de la voix sont alors extraites comme le débit, la force, la dynamique et la forme des ondes produites.

Un individu ne parle pas toujours de la même manière, ce qui nécessite l'application d'une méthode permettant d'éliminer certaines de ces variations. Ses caractéristiques formant une empreinte unique sont ensuite traitées par un algorithme et conservées pour une comparaison ultérieure. Il existe cinq principales méthodes de traitement de la voix : dépendante du sujet, indépendante du sujet, discours discontinu, discours continu et discours naturel.

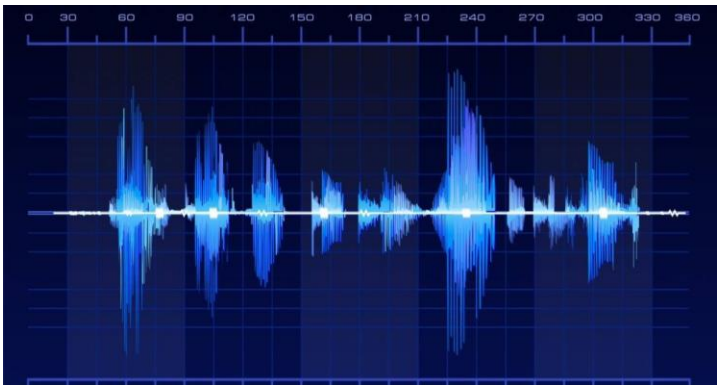


Figure I.6: la reconnaissance vocale [7]

I.2.3. Analyse comportementale:

I.2.3.1. La reconnaissance de la signature:

Des 1929, Osborne établit que l'écriture dépend de plusieurs facteurs donc non seulement limiter la forme de l'écriture mais aussi tenir compte de ces facteurs liés notamment à la conditions environnantes et à la dextérité musculaire.

Par la suite, diverses techniques de reconnaissance de la signature furent mises au moins au bénéfice notamment des banques et des corps policiers. Les systèmes de reconnaissance de l'écriture, analysent les caractéristiques spécifiques d'une signature comme la vitesse, la pression sur le crayon, le mouvement, les points et les intervalles

Figure I.7 : une signature scannée

De temps où le crayon est levé. L'utilisateur de cette technologie signe généralement avec un stylo électronique (voir la figure I.7) sont enregistrées pour comparaison ultérieure. Certains systèmes ne font qu'enregistrer l'image statique de la signature pour comparaison.

I.2.3.2. La reconnaissance de la démarche :

Il s'agit de reconnaître un individu par sa façon de marcher et de bouger (vitesse, accélération, mouvements du corps…), en analysant des séquences d'images. La démarche serait en effet étroitement associée à la musculature naturelle et donc très personnelle. Mais des vêtements amples, par exemple, peuvent compromettre une bonne identification.

I.2.3.3. La reconnaissance de la dynamique de la frappe au clavier :

Le système est basé sur la dynamique de frappe au clavier, il ne nécessite aucun équipement particulier, chaque ordinateur disposant d'un clavier. Il s'agit d'un dispositif logiciel qui calcule le temps ou un doigt effectue une pression sur une touche et le temps où un doigt est dans les airs (entre les frappes).

Cette mesure est capturée environ 1000 fois par seconde. La séquence de frappe est prédéterminée sous la forme d'un mot de passe. Initialement l'utilisateur doit composer son mot de passe à quelques reprises afin que soit constitué un gabarit de référence.

I.3. Architecture générale d'un système biométrique :

I.3.1. Qu'est-ce que la biométrie ?

La biométrie peut être définie comme étant "la reconnaissance automatique d'une personne en utilisant des traits distinctifs". Une autre définition de la biométrie est "toutes caractéristiques physiques ou traits personnels automatiquement mesurables, robustes et distinctives qui peuvent être utilisées pour identifier un individu ou pour vérifier l'identité prétendue d'un individu" [26].

I.3.2. Les Caractéristiques biométriques :

Une caractéristique biométrique est une donnée contenant l'essentiel d'informations permettant de différencier deux individus. Pratiquement n'importe quelle caractéristique physiologique ou comportementale peut être considérée comme une caractéristique biométrique, dans la mesure Où elle est :

Universelle : existe chez tous les individus.

Unique : différente pour chaque individu.

Permanente : stable dans le temps.

Enregistrable : atteignable.

Mesurable : une technologie de capteur existe.

Utilisable : acceptation par l'utilisateur.

Non imitable : difficilement copiable.

I.3.3. Système biométrique :

Un système biométrique est essentiellement un système de reconnaissance de formes qui fonctionne en acquérant des données biométriques à partir d'un individu, extrayant un ensemble de caractéristiques à partir des données acquises, et comparant ces caractéristiques contre la signature dans la base de données. Il sert à vérifier l'identité d'une personne à l'aide d'une ou plusieurs modalités qui lui sont propres [28].

I.3.3.1. Identification :

Est une comparaison "un à un", dans lequel le système valide l'identité d'une personne en comparant les données biométriques saisies avec le modèle biométrique de cette personne stocké dans la base de données du système.

La vérification est réalisée via un numéro d'identification personnel, un nom d'utilisateur, ou bien une carte à puce.

I.3.3.2. Authentification :

Est une comparaison "un à N", dans lequel le système reconnaît un individu en l'appariant avec un des modèles de la base de données. La personne peut ne pas être dans la base de données. Ce mode consiste à associer une identité à une personne [28].

I.4. Les module biométriques :

Il existe toujours au moins deux modules dans un système biométrique: le module d'apprentissage et celui de reconnaissance [6].

Le troisième module est le module d'adaptation (voir la figureI.8). Pendant l'apprentissage, le système va acquérir une ou plusieurs mesures biométriques qui serviront à construire un modèle de l'individu. Ce modèle de référence servira de point de comparaison lors de la reconnaissance. Le modèle pourra être réévalué après chaque utilisation grâce au module d'adaptation [9].

I.4.1. Module d'apprentissage :

Au cours de l'apprentissage, la caractéristique biométrique est tout d'abord mesurée grâce à un capteur; on parle d'acquisition ou de capture. En général, cette capture n'est pas directement stockée et des transformations lui sont appliquées. En effet, le signal contient de l'information

inutile à la reconnaissance et seuls les paramètres pertinents sont extraits. Le modèle est une représentation compacte du signal qui permet de faciliter la phase de reconnaissance, mais aussi de diminuer la quantité de données à stocker.

Il est à noter que la qualité du capteur peut grandement influencer les performances du système. Meilleure est la qualité du système d'acquisition, moins il y aura de prétraitements à effectuer pour extraire les paramètres du signal. Cependant, les capteurs de qualité sont en général coûteux et leur utilisation est donc limitée à des applications de haute sécurité pour un public restreint. Le modèle peut être stocké dans une base de données.

I.4.2. Module de reconnaissance

Au cours de la reconnaissance, la caractéristique biométrique est mesurée et un ensemble de paramètres est extrait comme lors de l'apprentissage. Le capteur utilisé doit avoir des propriétés aussi proches que possibles du capteur utilisé durant la phase d'apprentissage. Si les deux capteurs ont des propriétés trop différentes, il faudra en général appliquer une série de prétraitements supplémentaires pour limiter la dégradation des performances. La suite de la reconnaissance sera différente suivant le mode opératoire du système : Identification ou vérification.

En mode identification, le système doit deviner l'identité de la personne. Il répond donc à une question de type : ***« Qui suis-je ? »***. Dans ce mode, le système compare le signal mesuré avec les différents modèles contenus dans la base de données (problème de type 1 : N).

En général, lorsque l'on parle d'identification, on suppose que le problème est fermé, c'est-à-dire que toute personne qui utilise le système possède un modèle dans la base de données.

En mode vérification, le système doit répondre à une question de type :

« Suis-je bien la personne que je prétends être ? ».

L'utilisateur propose une identité au système et ce dernier doit vérifier que l'identité de l'individu est bien celle proposée. Il suffit donc de comparer le signal avec un seul des modèles présents dans la base de données (problème de type 1 : 1). En mode vérification, on parle de problème ouvert puisque l'on suppose qu'un individu qui n'a pas de modèle dans la base de données (imposteur) peut chercher à être reconnu.

Identification et vérification sont donc deux problèmes différents. L'identification peut-être une tâche redoutable lorsque la base de données contient des milliers, voire des millions d'identités,

tout particulièrement lorsqu'il existe des contraintes de type « temps réel » sur le système. Ces difficultés sont analogues à celles que connaissent par exemple les systèmes d'indexation de documents multimédia.

I.4.3. Module d'adaptation :

Pendant la phase d'apprentissage, le système biométrique ne capture souvent que quelques instances d'un même attribut afin de limiter la gêne pour l'utilisateur. Il est donc difficile de construire un modèle assez général capable de décrire toutes les variations possibles de cet attribut. De plus, les caractéristiques de cette biométrie ainsi que ses conditions d'acquisition peuvent varier. L'adaptation est donc nécessaire pour maintenir voire amélioré la performance d'un système utilisation après utilisation. L'adaptation peut se faire en mode supervisé ou non-supervisé mais le second mode est de loin le plus utile en pratique.

Si un utilisateur est identifié par le module de reconnaissance, les paramètres extraits du signal serviront alors à ré-estimer son modèle. En général, le taux d'adaptation dépend du degré de confiance du module de reconnaissance dans l'identité de l'utilisateur. Bien en tendu, l'adaptation non-supervisée peut poser problème en cas d'erreurs du module de reconnaissance. L'adaptation est quasi indispensable pour les caractéristiques non permanentes comme la voix [10].

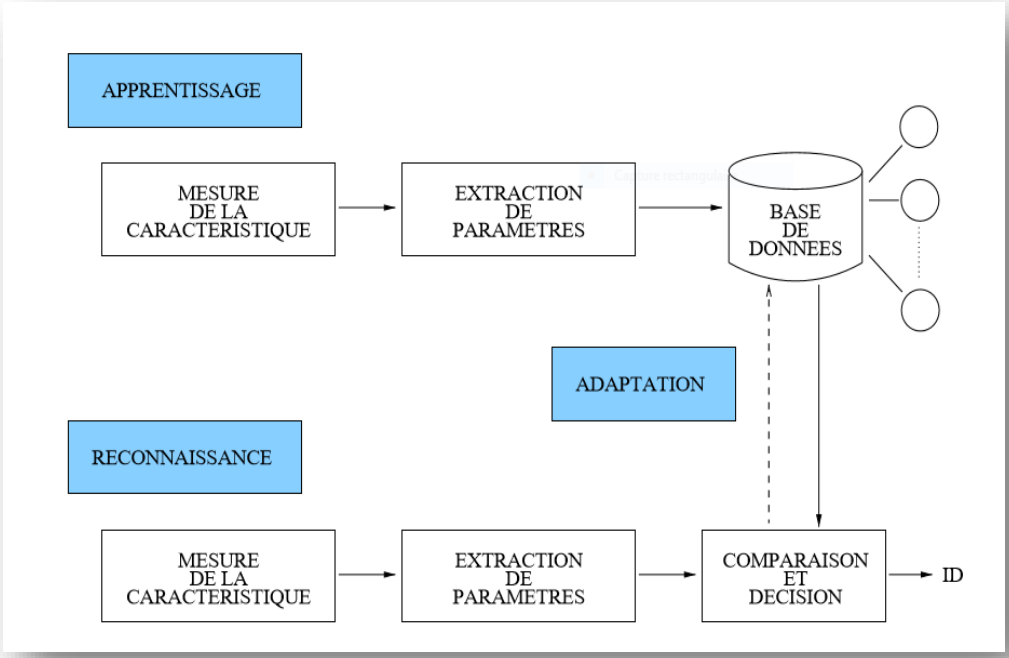


Figure : I.8: Architecture d'un système biométrique [13]

I.3.5. Les performances des systèmes biométriques :

Les performances d'un système biométrique sont données par la mesure de deux taux d'erreurs : le **FRR** (False Rejet Rate ou Taux de Faux Rejet) et le **FAR** (False Acceptation Rate ou Taux de Fausse Acceptation). Le FRR ou le TFR (Taux de Faux Rejets) : estime le pourcentage d'utilisateurs valides qui ne seront pas reconnus par le système. Le FAR ou le TFA (taux de fausse acceptation) : estime le pourcentage d'utilisateurs non connus qui seront faussement reconnus par le système.

Le paramétrage d'un système consiste à trouver le bon équilibre entre ces deux taux, le FAR augmentant lorsque le FRR diminue, et inversement. Un contrôle d'accès très sécurisé aura un FAR très bas, pour garantir qu'aucune personne non autorisée n'accède au site, mais, en contrepartie le FRR sera élevé, ce qui signifie que des utilisateurs valides se verront refuser l'accès. Les autres mesures de performance sont les temps d'encodage de l'empreinte et de mise en correspondance. Là encore, ces valeurs peuvent varier considérablement d'une application à une autre. Un troisième paramètre FER *(False Equal Rate)* mesure le taux d'échec à l'enrôlement. Il traduit la probabilité d'absence d'une caractéristique biométrique pour un individu dans une population, donne un point sur lequel le T.F.A. est égal au T.F.R. [12]

La figure I.9 illustre le FRR et le FAR à partir de distributions des scores authentiques et imposteurs.

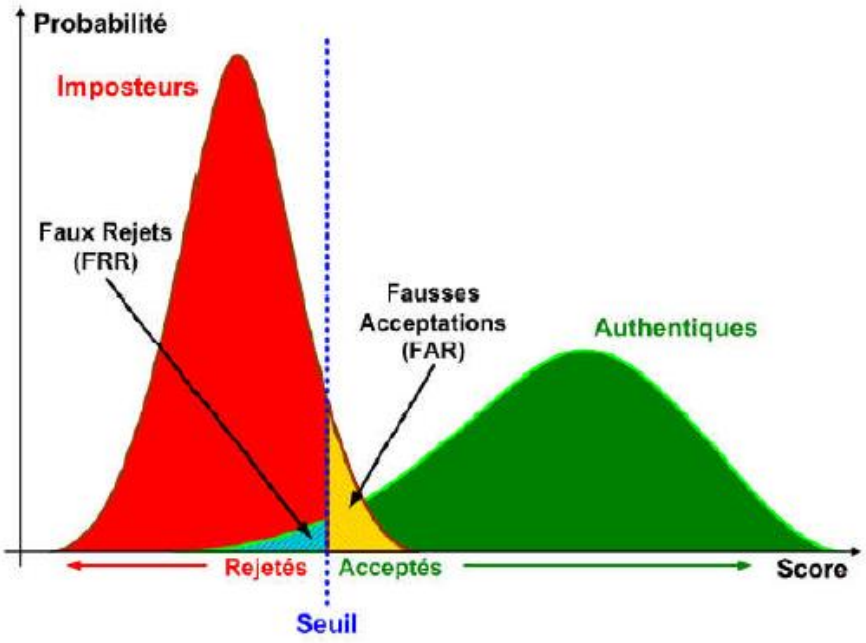


Figure I.9 : Illustration du FRR et du FAR [12].

I.3.6. Applications de la biométrie :

L'application de la reconnaissance biométrique peut être divisée en trois groupes :

■ ***Applications commerciales*** : comme connexion aux réseaux informatiques, sécurisation des données électroniques, e-commerce, accès internet, distributeur automatique de billets, carte de crédit, contrôle d'accès physique, cellulaire, gestion des rapports médicaux et l'apprentissage à distance.

■ ***Applications gouvernementales*** : comme les cartes d'identité nationale, établissements pénitentiaires, permis de conduire, sécurité sociale, contrôle des frontières et des passeports .

■ ***Applications médico-légales*** *:* comme l'identification de cadavres, investigation criminelle, identification terroriste, détermination de paternité et des enfants disparus.

I.4.Conclusion :

Dans ce chapitre, nous avons présenté un survol sur la biométrie en définissant les systèmes biométriques et nous avons cité quelques techniques biométriques de la détection et la vérification et enfin l'architecture générale de ce système biométrique.

Parmi les modalités utilisées dans la reconnaissance biométrique, nous avons trouvé que la texture des minuties de l'empreinte digitale est le trait le plus intéressant à cause de leur précision et leur stabilité. De même, l'utilisation de l'empreinte digitale suscite de plus en plus l'intérêt de la communauté scientifique car elle présente plusieurs challenges et verrous technologiques.

Chapitre II

Les empreintes digitales

II.1 Introduction :

Plusieurs caractéristiques humaines ont été exploitées par la biométrie pour l'identification et la vérification automatique des individus, Les empreintes digitales sont un outil d'identification rapide, fiable et moins onéreux que certains autres. L'utilisation de l'empreinte digitale comme moyen d'identification d'une personne n'est pas nouvelle.

C'est la technique biométrique la plus ancienne et la plus mature. Les empreintes ont formellement été acceptées comme identificateur de personnes valide dès le début du siècle. Elles ont d'abord étaient utilisées dans les milieux juridiques, avant de devenir une technique d'authentification effective.

II.2 Historique :

L'histoire des empreintes est longue, nous donnons ici un bref aperçu [30] ; Les empreintes n'ont pas été décrites sur les manuscrits jusqu'au 17eme siècle.

• En 1686, Marcello Malpighi un professeur d'anatomie à l'université de Bologne (Italie) décrit les crêtes papillaires dans son traite.

• En 1888, le Britannique Galton un anthropologue anglais et cousin de Charles Darwin démontre la permanence du dessin papillaire de la naissance à la mort ainsi que son inaltérabilité. Cet arrangement particulier des lignes papillaires forme des points caractéristiques nommés minuties ou points de Galton qui sont à l' origine de l'individualité des dessins d'empreintes. En se basant sur ces calculs, la probabilité pour que les empreintes de deux individus différents se correspondent est de 1 sur 64 Billions.

• En 1901, les empreintes furent introduites pour l'identification de criminels en Grande Bretagne. Les observations de Galton et leur révision par Edward Henry ont été utilisées. Cela marque le fondement du système de classification de Henry. L'avènement de l'ordinateur et les progrès récents réalisés dans le domaine de la reconnaissance des formes ont aidé à développer les systèmes d'identification automatiques. Ces systèmes ont considérablement amélioré la productivité opérationnelle des agences de loi et ont réduit le cout d'employer et de former les experts d'empreintes digitales.

II.3. Caractéristiques des empreintes :

Une empreinte digitale est constituée d'un ensemble de lignes localement parallèles formant un motif unique pour chaque individu (Figure II.1), on distingue **les *stries*** (ou crêtes, ce sont les lignes en contact avec une surface au toucher) et **les *sillons*** (ce sont les creux entre deux stries). Les stries contiennent en leur centre un ensemble de ***pores*** régulièrement espacés.

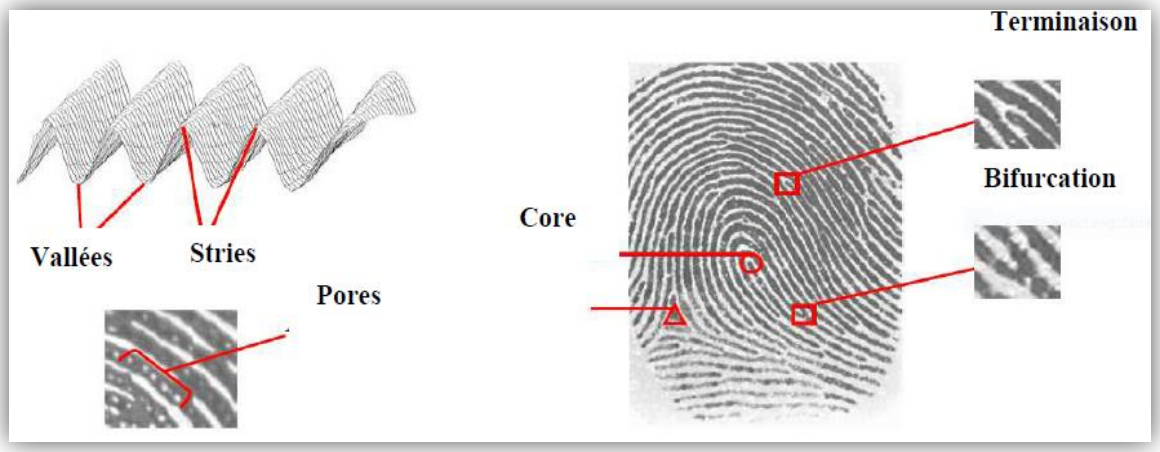


Figure II.1: Caractéristiques d'une empreinte digitale [29]

Chaque empreinte possède un ensemble de points singuliers globaux (les *centres* et les *deltas*) et locaux (les *minuties*). Les centres correspondent à des lieux de convergences des stries, tandis que les deltas correspondent à des lieux de divergence [15]. Plusieurs études ont montré l'existence de seize types de minuties différentes (Figure II.2) mais en général les algorithmes ne s'intéressent qu'aux bifurcations et terminaisons qui permettent d'obtenir les autres types par combinaison.

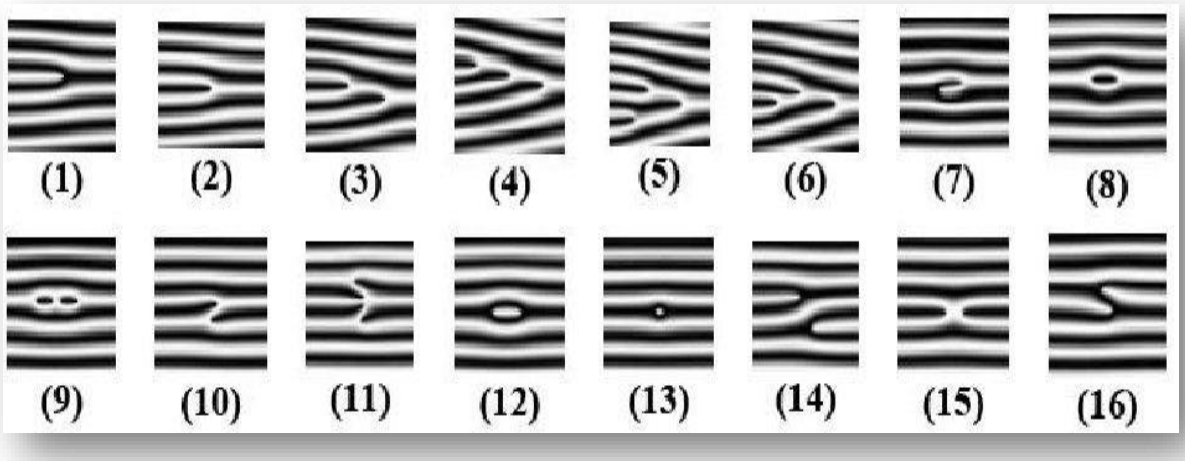


Figure II.2: Les différents types de minuties [15].

1. Terminaison
2.bifurcation simple
3. bifurcation double
4.bifurcation triple 1
5. bifurcation triple 2
6.bifurcation triple 3
7. crochet
8. Boucle simple
9. boucle double
10.pont simple
11. pont jumeau
12.intervalle
13. point isolé
14.Traversée
15. croisement
16.tête bêche

La position et le nombre de centres et de deltas permettent de classifier les empreintes en catégorie selon leur motif général, on distingue principalement trois grandes familles (voir Figure II. 3) :

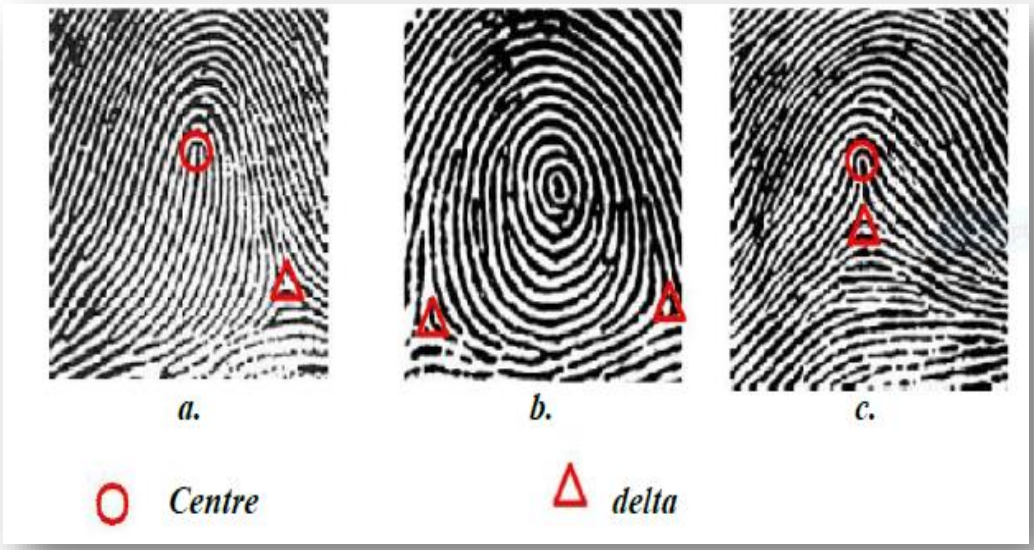


Figure II.3: Les trois principales classes d'empreintes, boucle (a), spire (b), Arche (c) [15].

Avec :

a. La Classe Boucle :

Les lignes ont un trajet récurrent et reviennent aux bords dont elles sont parties.

b.la Classe spire :

Les lignes présentent un trajet plus au moins spirale et limité vers les bords du doigt.

c.La Classe Arche :

Les lignes vont d'un bord à l'autre du doigt.

La forme des crêtes du doigt ne changent pas à moins d'accidents telles que des contusions et des coupes sur les bouts du doigt. L'empreinte se forme à partir du troisième mois de la vie fœtale [13], le motif général est influencé par les gènes héréditaires mais l'apparition des minuties est créée accidentellement par des pressions variables et aléatoires sur les surfaces tactiles [15].

De plus l'empreinte une fois formée, ne change plus au cours de la vie de l'individu, cette propriété fait des empreintes digitales une marque biométrique très attrayante.

II.4. Structure d'un système complet de reconnaissance d'empreintes.

II.4.1. Principe général

Un système automatique complet de reconnaissance d'empreintes digitales est une chaîne de processus qui à partir du doigt d'un utilisateur en entrée renvoie un résultat en sortie, permettant ainsi à l'utilisateur d'accéder ou non à des éléments nécessitant une protection.

La réalisation d'un tel système a fait l'objet de très nombreuses recherches et des méthodes très différentes de traitement ont été proposées [16]. Cependant ces systèmes répondent toujours à la même structure (Figure II.4).

La première phase permet d'obtenir une image de l'empreinte de l'utilisateur (*acquisition*), laquelle va subir un prétraitement pour extraire l'information utile de l'image (*signature*) suivi éventuellement d'un traitement supplémentaire permettant d'éliminer de possibles fausses informations qui se seraient glissées entre temps dans la chaîne de traitement. Ensuite si l'utilisation du système consiste juste à créer une base de données (*stockage*) la signature est éventuellement compressée puis stockée dans la base de données au moyen d'une technique d'archivage (*classification*).

Pour un système d'identification, l'ensemble des empreintes présentes dans la base de données pouvant correspondre à celle de l'utilisateur (modèle identique) sont désarchivées et comparées (*appariement*) une à une avec celle de l'utilisateur, si une éventuelle correspondance est trouvée des informations personnelles concernant l'utilisateur sont renvoyées par le système.

Dans le cas d'un système de vérification il n'y a qu'une seule comparaison et un résultat binaire est renvoyé, permettant l'acceptation ou le rejet de l'utilisateur.

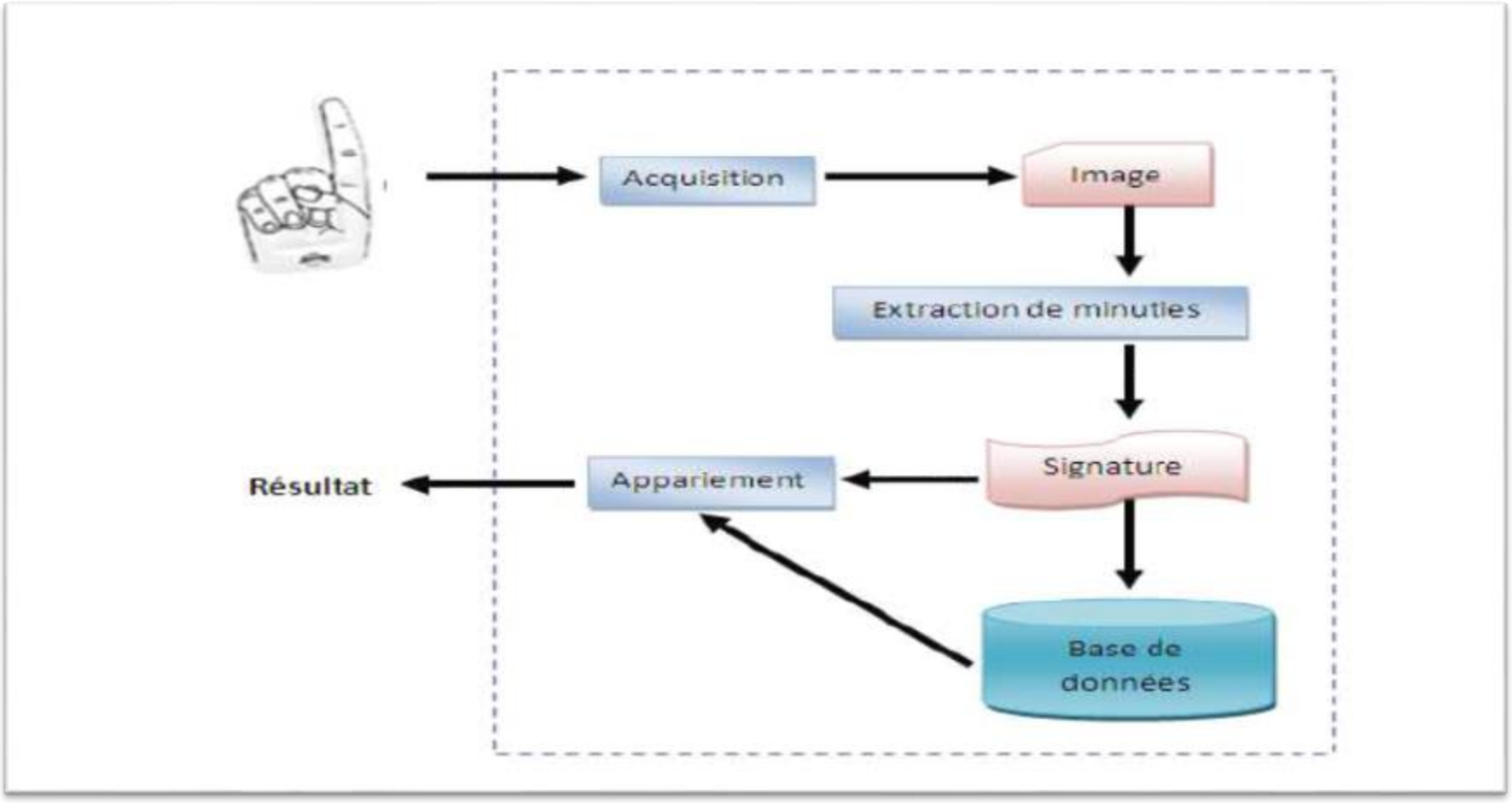


Figure II.4 : *Architecture générale d'un système complet de reconnaissance d'empreintes* [31]

II.4. 2. L'acquisition de l'empreinte :

L'acquisition d'empreinte s'agit de capturer les images numériques d'empreintes qui consistent à trouver les lignes tracées par les crêtes (en contact avec le capteur) et les vallées. La qualité d'image de l'empreinte digitale peut varier selon que la peau du doigt est sale, trop humide ou trop sèche, huileuse ou affligée d'une coupure. La pression que l'on exerce sur le lecteur optique de l'appareil est aussi déterminante quant aux détails qui sont recueillis. Un bon système biométrique tiendra compte de ces facteurs.

Le point commun à toutes les technologies utilisées pour la prise d'image d'une empreinte, est que l'image est constituée à partir des points de contact du doigt sur le capteur.

Les techniques d'acquisition sont diverses on citera [17]**:**

II.4. 2.1. Les capteurs optiques d'empreinte :

La méthode optique est une des méthodes les plus communes. Un appareil-photo CCD (Dispositif Charge Couplé) est utilisé au cœur du capteur optique. Il se compose simplement d'une rangée de diodes sensibles légères appelées photo sites. En général, le doigt est placé sur une surface en verre et le CCD prend la photo. Le système contient une rangée de LED qui illumine les creux et les bosses du doigt. Un prix avantageux constitue l'avantage principal des systèmes optiques ; leur inconvénient est qu'ils sont faciles à détourner. L'autre problème est celui des empreintes latentes : l'empreinte digitale du doigt précédente, qui a été placé sur le capteur, peut rester.

Figure II.5 : deux modèles du capteur optique [17]

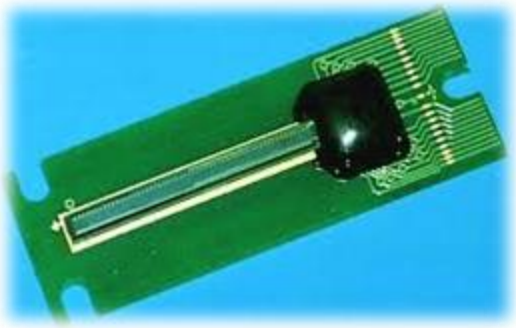
II.4.2.2. Les capteurs électriques-thermique :

La méthode pour reconnaître l'empreinte, consiste à faire glisser le doigt le long du capteur. Ce dernier mesure la différence de température entre les creux de la peau et l'air capturé dans les bosses de l'empreinte digitale. Cette méthode donne une image d'excellente qualité même sur des empreintes de qualité médiocre telles que celles provenant de doigt sec avec peu de profondeur entre les creux et les bosses. La technologie thermique fonctionne également dans des conditions environnementales difficiles, comme lors de températures extrêmes, de taux d'humidité ou de poussière élevé, ou de contamination d'eau. Cette méthode a également l'avantage de nettoyer le capteur, évitant ainsi que les empreintes digitales restent après le passage de chaque personne.

En fait, cette méthode, s'appuyant sur la technologie thermique permet au capteur d'être un des plus résistant par rapport aux autres technologies. L'inconvénient est le chauffage du capteur qui augmente la consommation électrique [27].

II.4.2.3. Capteurs capacitifs :

La méthode capacitive est l'une des méthodes les plus populaires. Comme les autres capteurs, le capteur capacitif reproduit l'image des creux et des bosses qui composent une empreinte digitale. Le capteur capacitif emploie des condensateurs de courant électrique pour mesurer l'empreinte, il se compose d'une rangée de cellules minuscules. Chaque cellule inclut deux plaques conductrices recouvertes par un revêtement protecteur. L'avantage principal de ces capteurs est qu'ils demandent une réelle empreinte digitale. Mais ils rencontrent des difficultés avec les doigts secs et humides[27].



II.4.2.4. Capteurs de champ-électrique :

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

La technologie de champ-électrique crée un champ entre le doigt et le semi-conducteur adjacent qui imite la forme des creux et des bosses de la couche épidermique du doigt. Un amplificateur de sous-Pixel est utilisé pour mesurer les signaux. Les capteurs fonctionnent ensemble afin de rendre une image plus claire correspondant exactement au modèle de l'empreinte digitale. On parvient ainsi à une image plus claire que ce que peuvent donner les technologies optiques ou capacitives. L'inconvénient est la basse résolution d'images et une trop petite zone d'image, ce qui a pour conséquence de générer un haut taux d'erreur.

II.4.3. Le prétraitement de l'image :

Les algorithmes de reconnaissance des empreintes digitales sont sensibles à la qualité des images d'empreintes digitales obtenue lors de l'acquisition [18].

La qualité de ces images dépend de plusieurs facteurs comme ;

- Les substances parasites présentes sur le doigt (encre, graisse, saletés…).
- La personne (cicatrices, métiers manuels, âge….).
- L'environnement où se produit l'acquisition (température de l'air, degré, d'humidité….).
- Les caractéristiques spécifiques du moyen d'acquisition utilisé.
- La profondeur de rides/vallée, etc.

Alors l'étape de prétraitement est nécessaire avant d'effectuer les étapes suivantes.

Typiquement le prétraitement peut se composer de lissage, segmentation et filtrage du domaine spatiale/ fréquence.

II.4.4.L'extraction de la signature :

La plupart des systèmes de reconnaissance des empreintes digitales emploient des minuties comme caractéristiques des empreintes digitales. Alors cette partie présentera les méthodes pour extraire des minuties à partir des empreintes digitales [19].

Un extracteur de minuties cherche des terminaisons de stries et des bifurcations dans les empreintes. Si les stries sont bien déterminées, alors l'extraction de minuties est une tâche

relativement simple. Cependant, dans la pratique, il n'est pas toujours possible d'obtenir une carte parfaite de stries. Donc la performance des algorithmes d'extraction de minuties dépend fortement de la qualité des images des empreintes digitales d'entrée [25].

II.4.5. Le stockage et la phase d'appariement

Pour les systèmes disposant de grosses bases de données, l'identification peut poser problème en temps de calcul si la signature d'entrée doit être comparée avec les signatures présentes dans la base donnée. C'est pourquoi, un processus de classification et de dé-classification est nécessaire pour limiter les temps de recherche [21].

Lorsqu'une image est stockée, un groupe spécifique lui est attribué en fonction de ses caractéristiques. Lors de l'identification, on désarchive l'ensemble des signatures de la base correspondant au groupe de l'empreinte nécessitant l'identification. Puis chacune des images désarchivées est comparée avec celle de l'utilisateur. Ceci permet de réduire sensiblement le temps de recherche en limitant le nombre d'images à comparer. Parmi les différentes techniques existantes on distingue principalement : l'extraction des singularités de l'image (la position des centre et delta permet de déterminer la classe de l'empreinte) [22].

La phase d'appariement est l'étape critique du système, elle reçoit en entrée deux signatures issues de deux acquisitions différentes d'empreinte et renvoie en sortie un résultat binaire indiquant si oui ou non les deux signatures proviennent de la même empreinte [23].

II.5.Représentation de l'empreinte digitale

Les systèmes de reconnaissance par empreinte digitale rencontrent des problèmes lors du processus d'acquisition de l'empreinte digitale tels que (les blessures, les coupures, les bleus … etc.). L'objectif étant de représenter l'empreinte digitale de façon invariante qui ne s'altère pas avec le temps. Cette représentation peut être globale prenant en compte toute l'image ou, locale c'est-à-dire constituée d'un ensemble de composantes dérivée chacune d'une région restreinte sur l'empreinte [24].

II.5.1 Représentation en image :

Cette représentation prend en considère toute l'image et l'appariement se réalise par la corrélation ; La mise en correspondance consiste à rechercher la magnitude du pic dans l'image de corrélation. Sa position indique la translation entre les images et sa valeur informe sur le degré de similarité. Les phénomènes de translation et de rotation dégradent l'exactitude de cette corrélation. Les méthodes de corrélation locale peuvent surmonter le problème de distorsion mais l'inconvénient principal de cette représentation est la taille conséquente de l'image à sauvegarder durant l'inscription [24].

II.5.2.Représentation avec les descripteurs de texture :

Les algorithmes basés sur les descripteurs de texture utilisent les informations telles que (la moyenne et la variance) et l'appariement s'effectue en calculant la distance euclidienne entre deux descripteurs [24]. Une mauvaise qualité risque de fausser la localisation des points. De plus, la performance est inferieure comparée aux appariements basés sur les minuties.

II.5.3 Représentation en minuties :

Les minuties représentent des discontinuités locales et marquent les positions où la crête se termine ou bifurque. Chaque minutie peut être décrite par un nombre d'attributs tels que: [24]

• Le type de minutie : Bifurcation ou Terminaison.

• La position de la minutie dans l'image : coordonnées (x, y)

• La direction du bloc local associé à la strie θ (voir la figure 6)

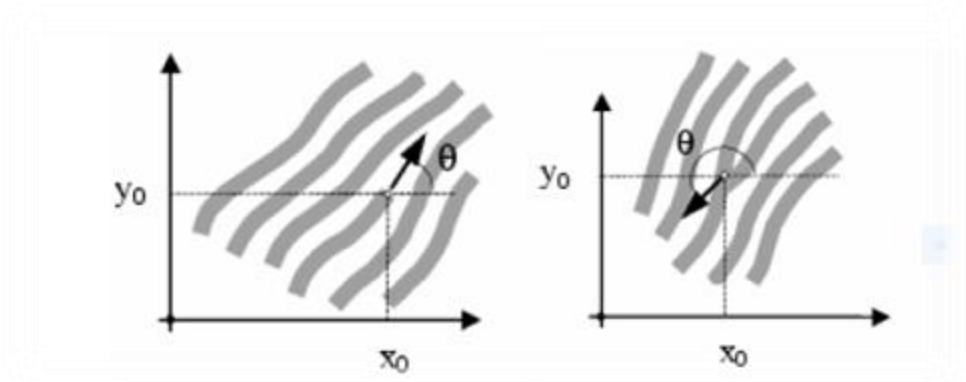


Figure II.6 : les caractéristiques principales des minuties [14].

La figure suivante est une représentation en minuties d'une empreinte :

Figure II.7: Exemple d'une représentation d'une empreinte par sa carte minuties [20]

L'appariement basé sur les minuties dépasse rarement 1ko contrairement aux algorithmes basés sur la corrélation qui nécessitent des gabarits de grande taille. De plus, Les points minuties sont invariants et ils ne sont pas influencés par les phénomènes de translation et rotation contrairement aux attributs de texture qui dépend de la qualité de l'image.

II.5.3.1.la détection des minuties :

Les minuties constituent les attributs les plus utilisés pour l'identification par empreinte digitale. Les systèmes existants se basent sur les détails des minuties.

La détection de minuties est une étape importante dans le processus de reconnaissance de L'empreinte digitale qui se base principalement en l'appariement de minuties. Plusieurs méthodes ont été développées pour réaliser une extraction de minuties efficace. Les méthodes d'extraction de minuties peuvent être classées dans la littérature en deux grandes catégories: celles qui se basent sur la binarisation de l'image et celles qui travaillent directement sur l'image en niveau de gris [32].

III.6.1.Prétraitement des images d'empreinte:

Basé sur la nature des bases de données proposées par [8] qui contient les différentes empreintes on observe que tous les images nécessitent un traitement. Le premier objective est de chercher à regrouper les images est les transformer en se basant sur la même dimension afin de faciliter la comparaison.

Figure II.8 : Echenillant des bases de données proposent par [8]

La 2ème étape est de faire un prétraitement au niveau d'image, ce traitement a pour objectif d'améliorer la qualité de l'image contre le bruit lié à la mesure de perturbation.

La figure 4 nous donne une idée sur le processus suivi dans phase du prétraitement.

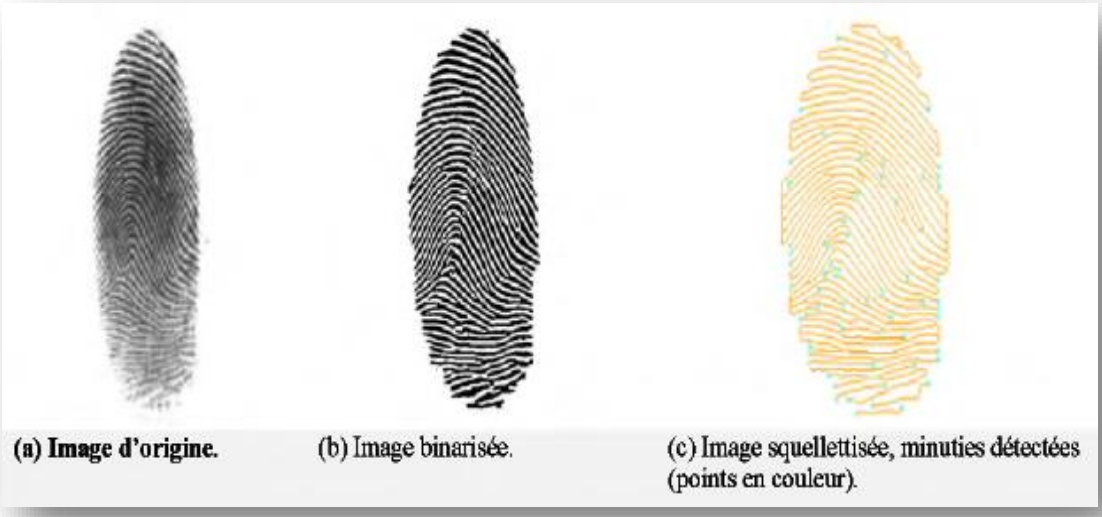


Figure II.9 : Traitement d'une empreinte digitale [8]

A) La binarisation:

La binarisation consiste à transformer une image à plusieurs niveaux en une image en noir et blanc (deux niveaux seulement). C'est le moyen privilégié pour isoler des objets. Par suite, une image binaire peut être représentée par une matrice booléenne dont chaque élément signifie Vrai (1 =blanc) ou Faux (0 = noir). [10] La binarisation d'empreintes digitales est une technique pour produire une image de type 1 -bit, avec 0 comme crêtes qui sont teintées de noir et de 1 les vallées qui sont teintées de blanc [9] (voir figureII.10).

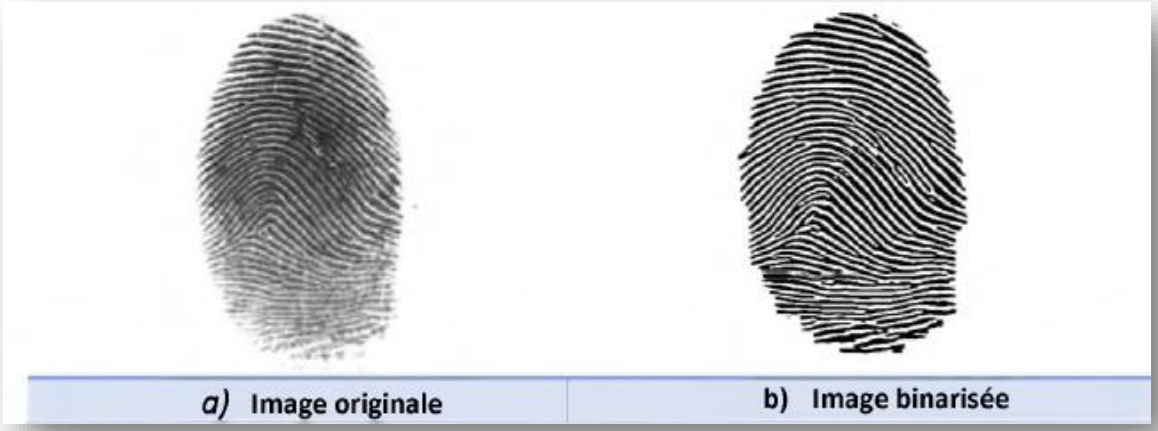


Figure II.10: Exemple d'opération du binarisation [8]

B) La squelettisation:

Un algorithme d'amincissement (ou shrinking algorithme) consiste en la suppression jusqu'à stabilité de points simples, le résultat obtenu s'appelle un noyau homotypique. Si la suppression est réalisée de façon séquentielle alors la topologie est préservée ; cela par la définition même d'un point simple. Si le processus est modifié de façon à ce que certains points simples soient préservés durant le processus de suppression, il est alors possible de Lignes d'épaisseur 1 pixel. La méthode nécessite l'emploi successif de 8 masques. On effectue sur l'image une succession de passes; on arrête lorsque le résultat entre deux passes successives est inchangé. Une passe consiste en l'application successive, sur toute l'image de chacun des 8 masques (le point central sur le point courant à traiter). Les 8 masques correspondent aux transformations suivantes : si la situation de gauche est rencontrée, alors on remplace le pixel traité par 0.

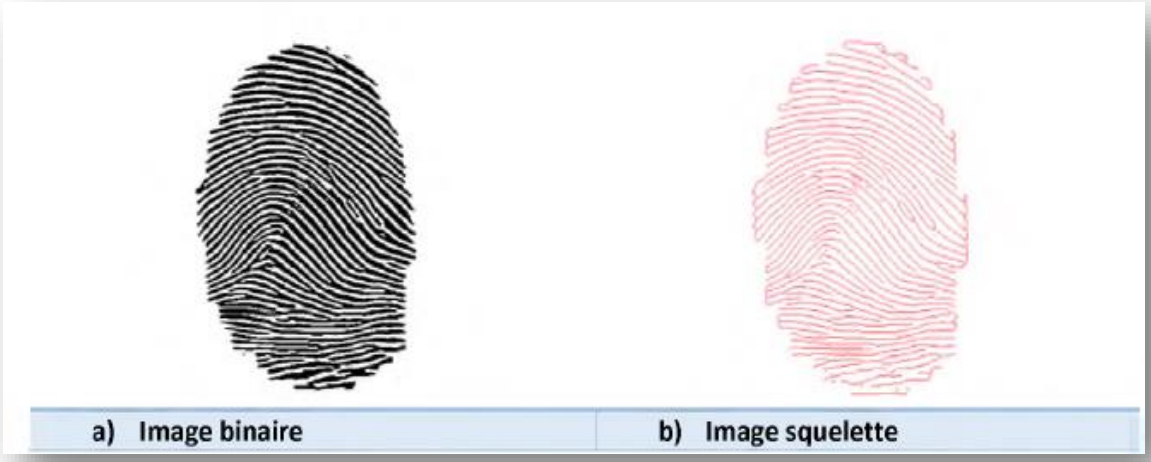


Figure II.11 : Exemple d'opération de squelettisation [8]

III.6.2. Extraction des minuties

Après avoir obtenu l'image traitée, on doit trouver dans cette dernière les minuties les plus intéressantes de l'image. L'extraction des minuties consiste à calculer le nombre de connexion CN de chaque pixel blanc avec ses 08 voisins.

Le couple (x, y) dénote un pixel sur une crête amincie et P0, P1,…P7 dénotent ses 8 voisins, ainsi le nombre de connexion :

$CN = (\sum_{i=0}^{7} PI)$

• si un pixel est sur une crête amincie, alors il prend la valeur 0 et sinon il prend la valeur1.

• Un pixel (x, y) est une fin de crête (Terminaison) si CN =1

• Un pixel (x, y) est une Bifurcation si CN >2.

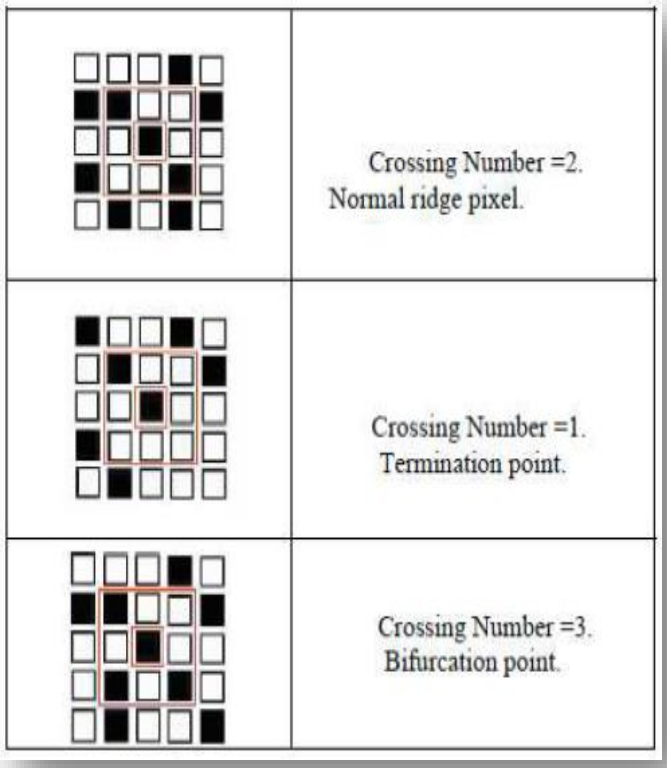


Figure II.12 : le CN et le type des minuties.

II.7.Conclusion :

Dans ce chapitre nous avons fait un tour d'horizon sur les caractéristiques principales de l'empreinte digitale ainsi, nous avons décrit la structure globale d'un système de reconnaissance par empreinte digitale et décrit les différentes représentations possibles de l'empreinte digitale. Nous avons montré aussi que la représentation en minuties était avantageuse et la plus aboutie.

Chapitre III

Conception et réalisation

III.1.Inroduction :

Dans ce chapitre, Nous présenterons l'environnement de travail pour réaliser notre système. Notre travail consiste à réaliser un système qui a pour but la reconnaissance d'empreintes digitales (identification et authentification d'empreintes), nous présentons la démarche et toutes les étapes qui nous ont permis d'atteindre notre objectif.

III.2. La partie Hardware

Afin de mener à bien ce projet, il a été mis à notre disposition un ensemble de matériels dont les caractéristiques sont les suivants:

III.2.1.Un ordinateur DELL DESKTOP-6ASV6MJ avec les caractéristiques suivantes:

- Processeur Intel® Core™ i5-5300U d'une fréquence de 2.29 GHz.
- Une mémoire vive d'une capacité de 8 GO.
- Une carte graphique Intel® HD Graphiques 5500.
- Windows 10.

III.2. 2. La carte Arduino (Uno):

Ce sont des cartes électroniques programmables (donc dotées d'un processeur et de mémoire) sur lesquelles nous pouvons brancher des capteurs de température, d'humidité, de vibration ou de lumière, une caméra, des boutons, des potentiomètres de réglage, des contacts électriques…etc. Il y a aussi des connecteurs pour brancher des LED, des moteurs, des relais, des afficheurs, un écran…
Une carte Arduino est un cerveau qui permet de rendre intelligent des systèmes électroniques et d'animer des dispositifs mécaniques [36].

III.2. 2. 1. Avantages de la carte Arduino UNO:

Il y a de nombreuses cartes électroniques qui possèdent des plateformes basées sur des microcontrôleurs disponibles pour l'électronique programmée. Tous ces outils prennent en charge les détails compliqués de la programmation et les intègrent dans une présentation facile à utiliser. De la même façon, le système Arduino simplifie la façon de travailler avec les microcontrôleurs tout en offrant à personnes intéressées plusieurs avantages cités comme suit:

Le prix (réduits) : les cartes Arduino sont relativement peu coûteuses comparativement aux autres plates-formes. La moins chère des versions du module Arduino peut être assemblée

- à la main, (les cartes Arduino préassemblées coûtent moins de 2500 Dinars). Multi plateforme : le logiciel Arduino, écrit en JAVA, tourne sous les systèmes d'exploitation Windows, Macintosh et Linux. La plupart des systèmes à microcontrôleurs sont limités à Windows.
- Un environnement de programmation clair et simple : l'environnement de programmation Arduino (le logiciel Arduino IDE) est facile à utiliser pour les débutants, tout en étant assez flexible pour que les utilisateurs avancés puissent en tirer profit également.

- Logiciel Open Source et extensible : le logiciel Arduino et le langage Arduino sont publiés sous licence open source, disponible pour être complété par des programmateurs expérimentés. Le logiciel de programmation des modules Arduino est une application JAVA multi plateformes (fonctionnant sur tout système d'exploitation), servant d'éditeur de code et de compilateur, et qui peut transférer le programme au travers de la liaison série (RS232, Bluetooth ou USB selon le module).

- Matériel Open source et extensible : les cartes Arduino sont basées sur les microcontrôleurs Atmel ATMEGA8, ATMEGA168, ATMEGA 328, les schémas des modules sont publiés sous une licence créative Commons, et les concepteurs des circuits expérimentés peuvent réaliser leur propre version des cartes Arduino, en les complétant et en les améliorant. Même les utilisateurs relativement inexpérimentés peuvent fabriquer la version sur plaque d'essai de la carte Arduino, dont le but est de comprendre comment elle fonctionne pour économiser le coût.

III.2. 2. 2. Description de la carte :

La carte "Arduino Uno " utilise comme mémoire l'Atmega 328 [36] contient **:**

- La mémoire FLASH contient 32 Ko, c'est vrais 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.

La majorité des cartes Arduino ont **ces composants** en communs :

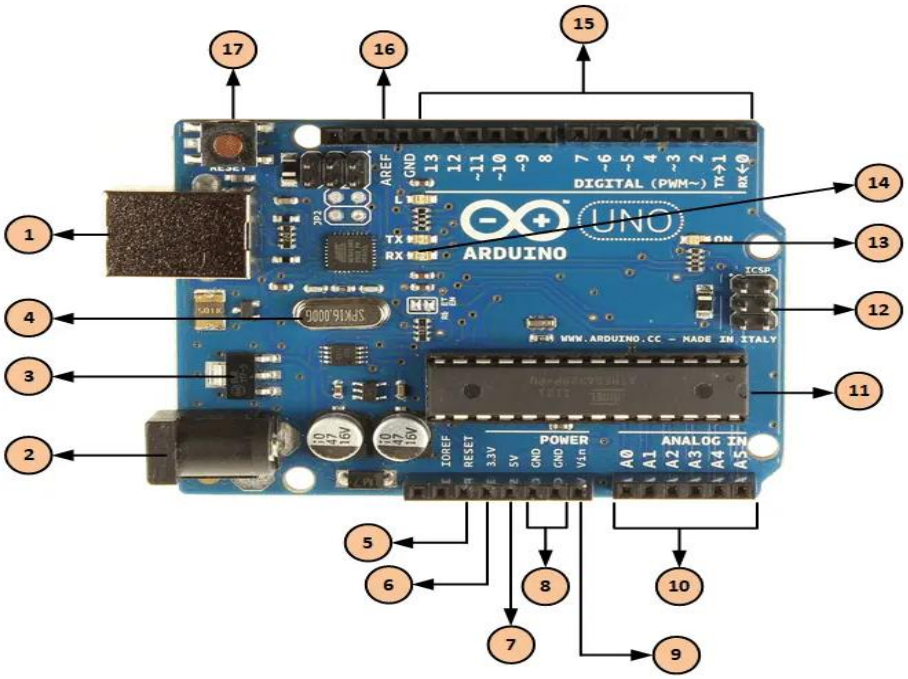


Figure III.1 : Description de la carte Arduino. [33]

1	**Alimentation / Programmation par USB** La carte Arduino peut être alimentée avec un câble USB relié à votre ordinateur. Tout ce dont vous avez besoin, c'est de connecter votre carte Arduino à votre ordinateur avec le câble USB type A/B.
2	**Alimentation via connecteur Jack DC** Diamètre interne 2.1mm, externe 5.5mm La carte Arduino peut être directement alimentée par ce connecteur Jack DC. Ce connecteur (2) est relié au régulateur de tension intégré à la carte. L'alimentation via ce connecteur (2) doit être comprise entre 5 et 12 V.
3	**Régulateur de tension** La fonction du régulateur de tension (3) est de contrôler la tension d'alimentation de l'Arduino pour la stabiliser à la bonne tension du microcontrôleur et de chaque élément de la carte. La tension de stabilisation est de 5 Volts sur les cartes UNO.
4	**Oscillateur à quartz** Un oscillateur à quartz est un élément électronique qui a la particularité de posséder un quartz à l'intérieur qui vibre sous l'effet piézoélectrique. Les propriétés électromécaniques du quartz sont telles qu'on arrive à faire vibrer le quartz à une fréquence très précise. Cet élément aide l'Arduino UNO à calculer les données de temps. Sur le dessus du composant, on peut lire 16.000H9H. Cela signifie que la fréquence est de 16, 000,000 Hertz, soit 16 MHz.

5,17	**Arduino Reset** Vous pouvez redémarrer un Arduino avec un “Reset”. Cela aura pour effet de redémarrer votre programme depuis le début. Vous pouvez redémarrer l’Arduino UNO de deux manières : soit en utilisant le bouton “Reset” (17), soit en connectant un bouton externe sur la broche de la carte Arduino mentionnée “RESET” (5).
6,7,8,9	**Broches (3.3, 5, GND, Vin)** * 3.3V (6) – Broche d’alimentation de tension 3.3 Volts * 5V (7) – Broche d’alimentation de tension 5 Volts * La plupart des composants destinés à fonctionner avec Arduino fonctionnent bien en 3.3 Volts ou 5 Volts. *GND (8) (Grounds / Masse) – Il y a plusieurs broches de ce type présentes sur la carte Arduino, elles sont toutes communes et peuvent être utilisées comme masse (potentiel 0 Volts) pour vos circuits. *Vin (9) – Cette broche permet d’alimenter l’Arduino depuis une source de tension extérieure. Elle est reliée au circuit d’alimentation principale de la carte Arduino.
10	**Broches analogiques** L’Arduino UNO possède 5 broches d’entrées analogiques numérotée de A0 jusqu’à A5. Ces broches permettent de lire un signal analogique d’un capteur comme un capteur d’humidité ou de température. La carte Arduino utilise un convertisseur analogique/numérique (convertisseur CAN) pour permettre la lecture du signal par le microcontrôleur. Un signal sera converti sur 10 bits. La valeur pourra être lue sur une échelle 1024 points.
11	**Microcontrôleur principal** Chaque carte Arduino possède son propre microcontrôleur (11). Vous pouvez le considérer comme le cerveau de la carte Arduino. Le microcontrôleur sur l’Arduino est légèrement différent d’une carte à l’autre. Les microcontrôleurs sont généralement de la société ATMEL. Vous devez savoir quel est le microcontrôleur de votre carte avant de charger un nouveau programme depuis l’IDE Arduino. Cette information est disponible directement sur le composant. Pour plus de détails sur la construction et les fonctions du microcontrôleur, vous pouvez vous référer à la fiche technique (data sheet).
12	**Connecteur ICSP** Avant tout, le connecteur ICSP (*In-Circuit Serial Programming*) est une connectique AVR comprenant les broches MOSI, MISO, SCK, RESET, VCC et GND. Il s’agit d’un connecteur de programmation. Ce connecteur permet entre autre de programmer directement le microcontrôleur sur les couches les plus basses (boot loader, code ASM…). C’est aussi un port appelé port SPI (Serial Peripheral Interface), qui permet de dialoguer avec d’autres composants SPI (écrans, capteurs, etc…). On ne va pas se préoccuper de ce connecteur au début des tutoriels.
13	**Indicateur LED d’alimentation** Ce voyant doit s’allumer lorsque vous branchez votre Arduino sur une source d’alimentation pour indiquer que votre carte est correctement alimentée. Si cette lumière ne s’allume pas, il y a un problème avec votre alimentation, et je ne parle pas de nourriture ici.

14	**LEDs TX et RX** Sur votre carte, vous trouverez deux indicateurs : TX (émission) et RX (réception)**.** Ils apparaissent à deux endroits sur la carte Arduino UNO. Tout d'abord, sur les broches numériques 0 et 1, pour indiquer les broches responsables de la communication série. Deuxièmement, les LEDs TX et RX (13). Le voyant TX clignote à une vitesse variable lors de l'envoi des données série. La vitesse de clignotement dépend de la vitesse de transmission utilisée par la carte. RX clignote pendant le processus de réception. La vitesse de transmission s'exprime en bauds, soit l'équivalent du bits/seconde si le signal est binaire.
15	**Entrées/Sorties numériques** La carte Arduino UNO possède **14 broches d'Entrées / Sorties numériques** (15), dont 6 peuvent fournir une sortie PWM (Pulse Width Modulation). Ces broches peuvent être configurées pour fonctionner comme des broches numériques d'entrée pour lire des valeurs logiques (0 ou 1) ou numériques. Elles peuvent également être utilisées comme des broches de sortie pour piloter différents modules comme des LEDs, des relais, etc. Les broches étiquetées "~" peuvent être utilisées pour générer des PWM.
16	**Broche AREF** AREF est l'acronyme anglais de "référence analogique". Cette broche est parfois utilisée pour définir **une tension de référence externe** (entre 0 et 5 Volts) comme limite supérieure pour les broches d'entrée analogiques.

Tableau III.1 : les composants d'une carte Arduino [34]

III.2. 3.Capteur Lecteur Empreinte Digitale FPM10A :

3.3 V/5 V picro-contrôleur de puissance ; FPM10A lecteur d'empreintes digitales Module de capteur optique pour Arduino verrouille l'interface de Communication série.

Figure III.2 : Capteur de lecteur d'empreintes digitales FPM10A [35]

III.2. 3.1.Description de l'article : [35]

Tension d'alimentation: cc 3.6 ~ 6.0 V / 3.3 V / Courant d'alimentation: < 120 mA

Courant de crête: < 140mA / Temps d'image d'empreinte digitale: < 1.0 secondes

Taille de fenêtre: 14 millimètres 18mm / Mode correspondant: mode Match (1 : 1)

Mode de recherche: (1 : N) / Fichier de Signature: 256 octets

Fichiers de modèle: 512 octets / Capacité de stockage: 200 maximums

Niveau de sécurité: cinq (de bas à haut: 1, 2, 3, 4, 5)

Faux taux d'acceptation (FAR): < 0.001% (niveau de sécurité 3)

Taux de faux rejet (FRR): < 1.0% (niveau de sécurité 3)

Temps de recherche: < 1.0 secondes (1:500, moyenne)

Interface PC: UART (niveau logique TTL) ou USB2.0 / USB1.1

Débit en bauds de Communication (UART): (9600 ╳ N) bps où N = 1 ~ 12 (valeur par défaut N = 6, soit 57600bps)

Environnement de travail:

Température: -20 ℃ - 50 ℃

Humidité Relative: 40% RH-85 % Hr (sans condensation).

Environnement de stockage:

Température: -40 ℃ - 85 ℃

Humidité Relative: < 85% H (sans condensation)

Dimensions (L épaisseur W épaisseur H):

Fente: capteur d'empreinte digitale: 56 millimètres 20 millimètres 21.5mm

-Un: 56 millimètres 20 millimètres 21.5mm

III.3. La partie Software :

Pour écrire, modifier ou injecter les programmes (le code) dans le microcontrôleur, on utilisera une interface graphique appelée IDE (Integrated Development Environment).

III.3.1.Téléchargement et l'installation du logiciel :

Sur Windows

- ✓ Télécharger la version Windows du logiciel "Arduino" sur [37].
- ✓ Installer le logiciel.
- ✓ Installer le driver USB de la carte Arduino :

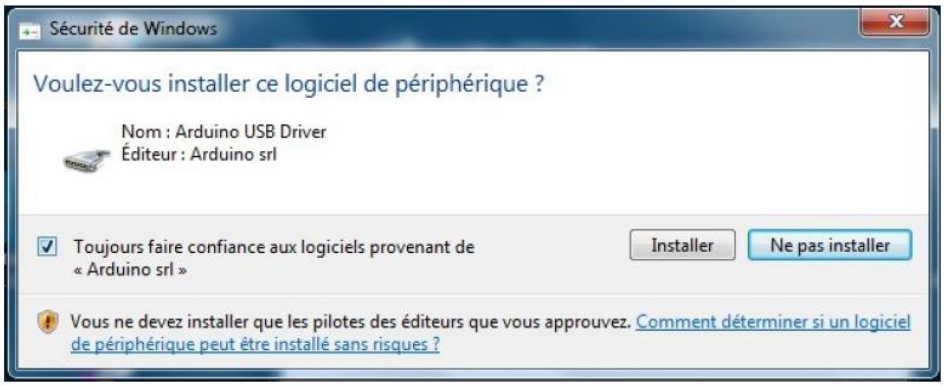


Figure III.3: fenêtre d'installation du logiciel Arduino [39]

- Connecter la carte ARDUINO à l'ordinateur à l'aide du câble USB.
  Une lumière verte s'allumer sur la carte, intitulé «ON».

- Lancer le logiciel et choisir la carte Arduino Uno et le port dans le menu ≪outils ≫.

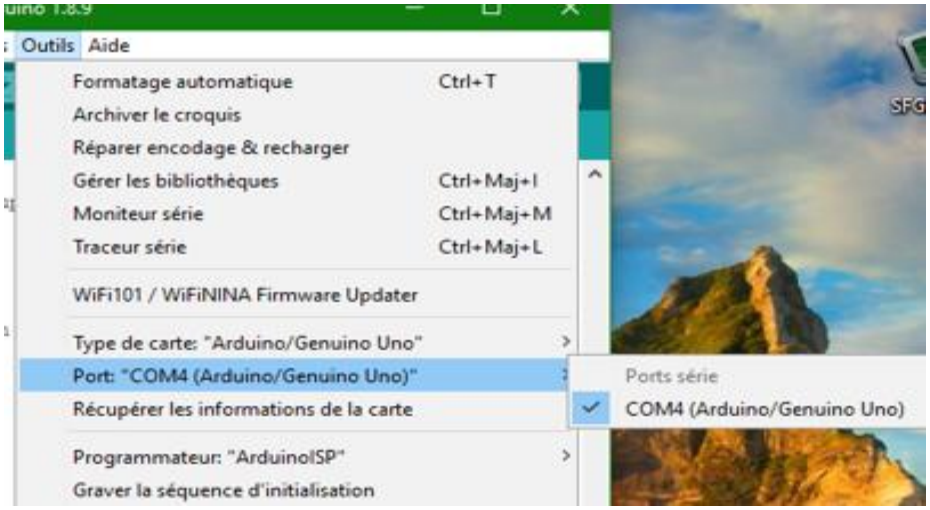


Figure III.4: Illustration comment choisir la carte et le port.

III.3.2.Structure générale du programme (IDE Arduino) :

Comme n'importe quel langage de programmation, une interface souple et simple est exécutable sur n'importe quel système d'exploitation Arduino basé sur la programmation en C.

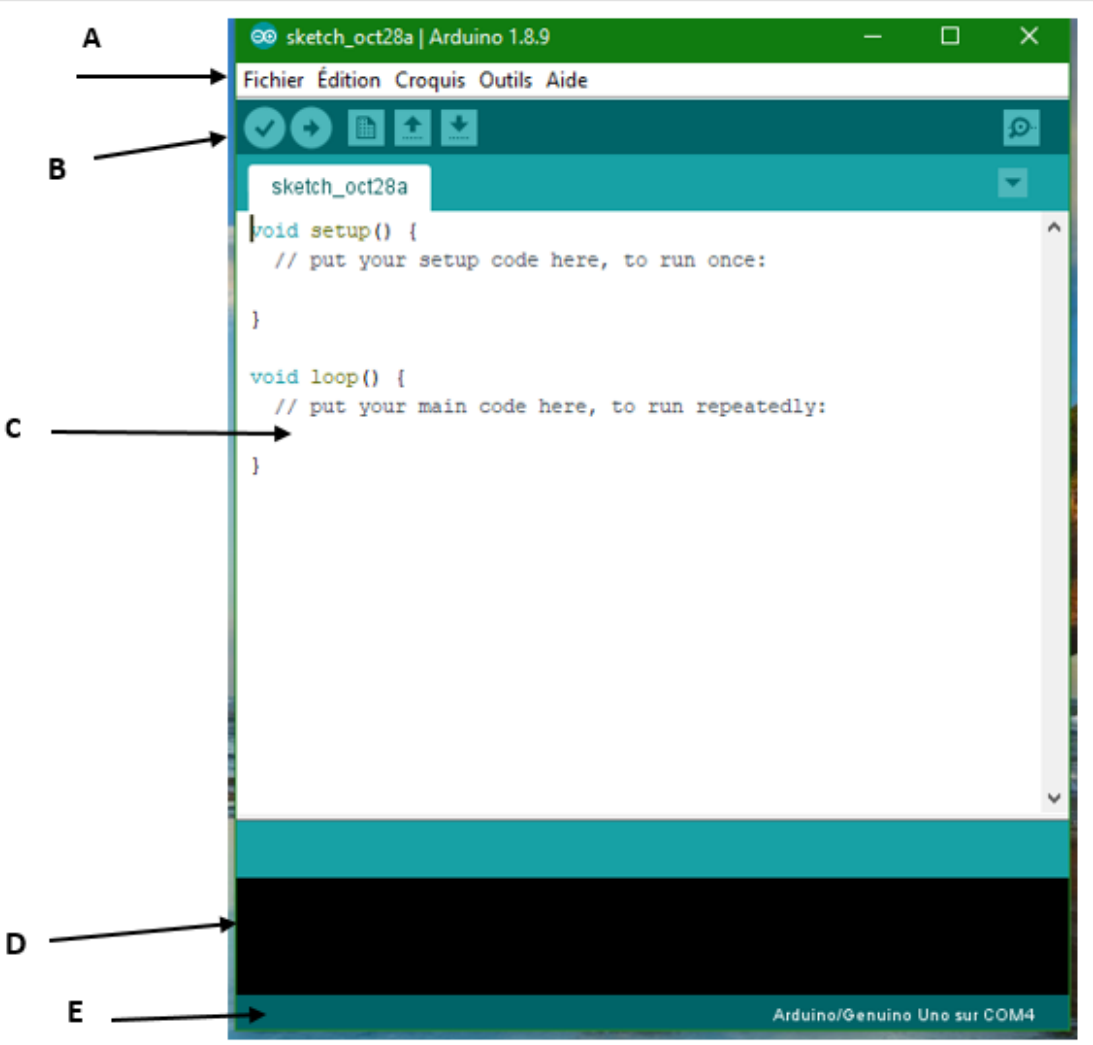


Figure III.5: Interface IDE Arduino

A : Barre de menus.

B : Barre d'outils (raccourcis les plus utiles de fonctions se trouvant dans les menus) : Vérifier le code | injecté le code dans le microcontrôleur | Nouveau code | Ouvrir | Enregistrer | Moniteur série (permet d'afficher des messages textes reçus de la carte Arduino et/ou d'envoyer des caractères vers la carte Arduino).

C : Zone d'écriture du programme.

D : Zone de notification lors de l'injection où sera indiqué ce qui se passe et les Eventuels messages d'erreur.

E : Barre de notification du port de communication employé (utile si on utilise plusieurs Arduino[35].

La compilation permet de voir les éventuelles erreurs dans le programme avant de le charger au niveau de la carte et le téléversement permet de charger le programme après correction des erreurs sur la carte en la connectant à l'ordinateur via un câble USB, celui qui ressemble à celui de l'imprimante.

(a)

(b)

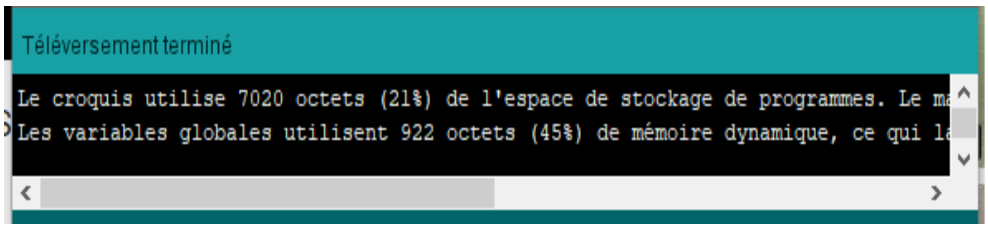


Figure III.6: (a) Statut d'un programme bien compilé (b) Statut d'un programme bien téléversé

III.3.2.1.Programmation de l' Arduino :

Un programme utilisateur Arduino est une suite d'instructions élémentaires sous forme textuelle, ligne par ligne. La carte lit puis effectue les instructions les unes après les autres, dans l'ordre défini par les lignes de code.Ce programme est structuré est trois grandes parties essentielles à savoir : la structure, les valeurs (variables et constantes) et les fonctions.

A. Structures :

Il est impératif de définir deux fonctions de bases ci-dessous :

Void setup () qui fera l' initiation et la configuration des états des broches

Void loop () s' en charge de nos instructions.

Mais aussi nous dénombrons d'autres structures telles que celles de contrôles (if, else, for, White, break,…), de :

- Comparaisons

= =	**Egale**
! =	**Différent**
< , >	**Inferieure supérieure**
<= , >=	**Inferieur ou égale, supérieur ou égale**

Tableau III.2 : Symboles des structures de comparaison

- Composés

++ ,- -	Incrémentation, décrémentation
+=, - =	Addition composée, soustraction composée
*=	Multiplication composée
/=	Division composée

Tableau III.3 : Symboles des structures composées

B. Variables :

Comme dans les autres langages de programmation, nous avons ici aussi des constantes prédéfinies à savoir :

- **HIGH, LOW** : Haut, Bas
- **INPUT, OUTPUT** : Entrée, Sortie
- **TRUE, FALSE** : Vrai, Faux
- **Constantes décimales**

Les différents types de données et leurs conversions sont les suivantes :

- **Void** : fonctions **Char** : caractère
- **Long** : réel long **Int** : entier
- **Float** : réel

*C. **Fonctions :***

Dans cette partie nous allons voir deux types d'entrées/sorties en fonction de l'utilisation à savoir:

- **Entrées/sorties analogiques de A0-A5**
  - AnalogRead (broche) : elle permet la lecture d'une broche analogique.
  - AnalogWrite (broche, valeur) : Elle permet quant à elle l'écriture d'une valeur sur une broche analogique généralement les 9, 10 ou 11.

- **Entrées/sorties numériques de D0-D13 :**
  - digital Read (broche) : sert à la lecture de l'état assigné à la broche.
  - digitalWrite (broche, valeur) : assignation d'une valeur sur une broche.
  - pin Mode (broche, état) : définition par écriture d'un état sur la broche.
  - unsigned long pulseIn (broche, état) : lecture d'une impulsion au niveau de la Broche.
- **Temps :**
  - Delay (ms) : temps d'attente en millisecondes.
  - delayMicroseconds (us) : attente en millisecondes.
  - Unsigned long millis () : période d'activité du programme.

III.3.2.2.Bibliothèque :

Dans cette partie nous décrirons comment choisissons-nous la bibliothèque d'Arduino cela correspond à notre montage, elle permet de faire la communication entre la carte et le programme (qui est un code).

- Choisir gérer les bibliothèques dans le menu ≪outils ≫ et La fenêtre suivante apparaît :

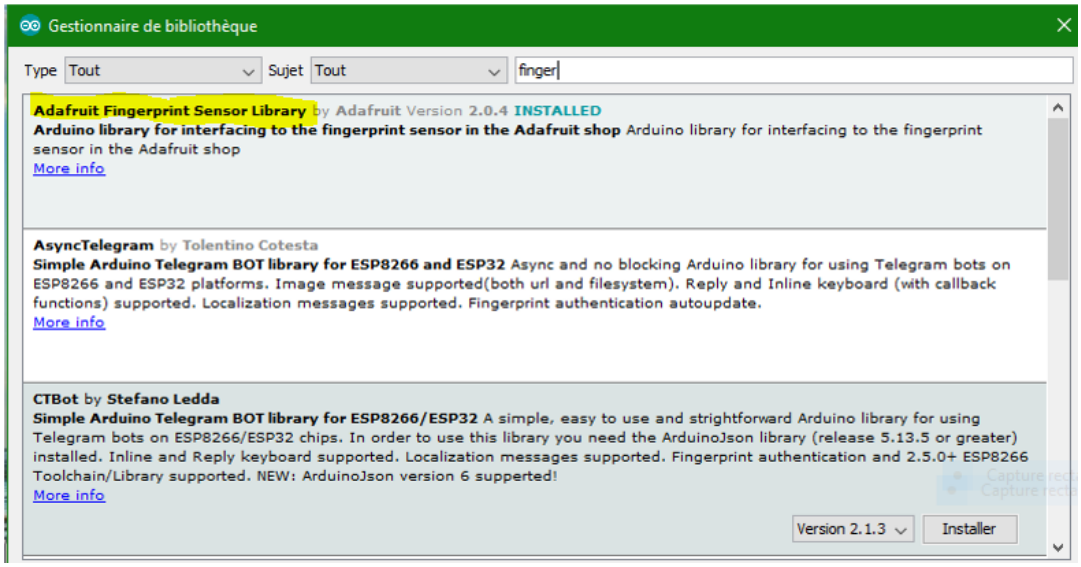


Figure III.7 : gestionnaire de bibliothèque d'Arduino.

III.4.La partie réalisation :

Réalisation du montage :
Le but du montage de démonstration sera de tout simplement entrer les empreintes digitales dans notre système, qui sont stockées dans la base de données pour une identification et une authentification ultérieure.
Pour réaliser ce montage :

- Une carte UNO (et son câble USB).
- Lecture d'empreinte digitale FPM 10A.
- fils des cavaliers.

III.4.1.Principe de fonctionnement

L'ordinateur est relie au microcontrôleur par un port USB qui sert à transmettre les informations ainsi qu'à alimenter en électricité la carte Arduino. Les données extérieurs des capteurs sont envoyées au microcontrôleur .S'il faut effectuer une action, l'ordinateur envoie une instruction au microcontrôleur qui, via un relais, agit sur l'appareil électrique. L'ordinateur peut être remplacé par un smartphone. De plus, les données relevées par les capteurs peuvent être visibles sur un écran.

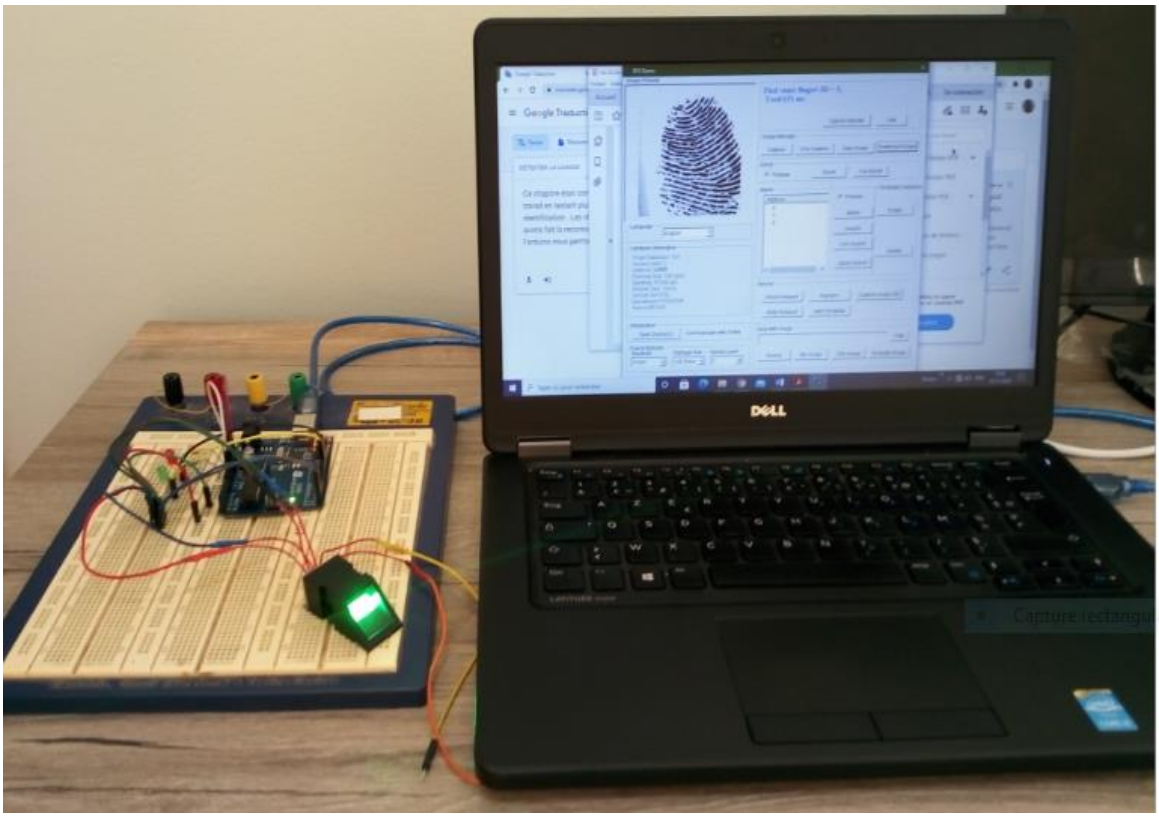


Figure III.8 : le montage réaliser au cours de ce projet

- Maintenant on placer notre capteur avec l'Arduino comme dans la figure III.9.

  Vert: alimentation 5V avec alimentation 5v.

  Rouge : masse (GND) avec GND.

  Marron: file Rx (récepteur) avec Tx.

  Bleu: fils Tx (Transmetteur) avec Rx.

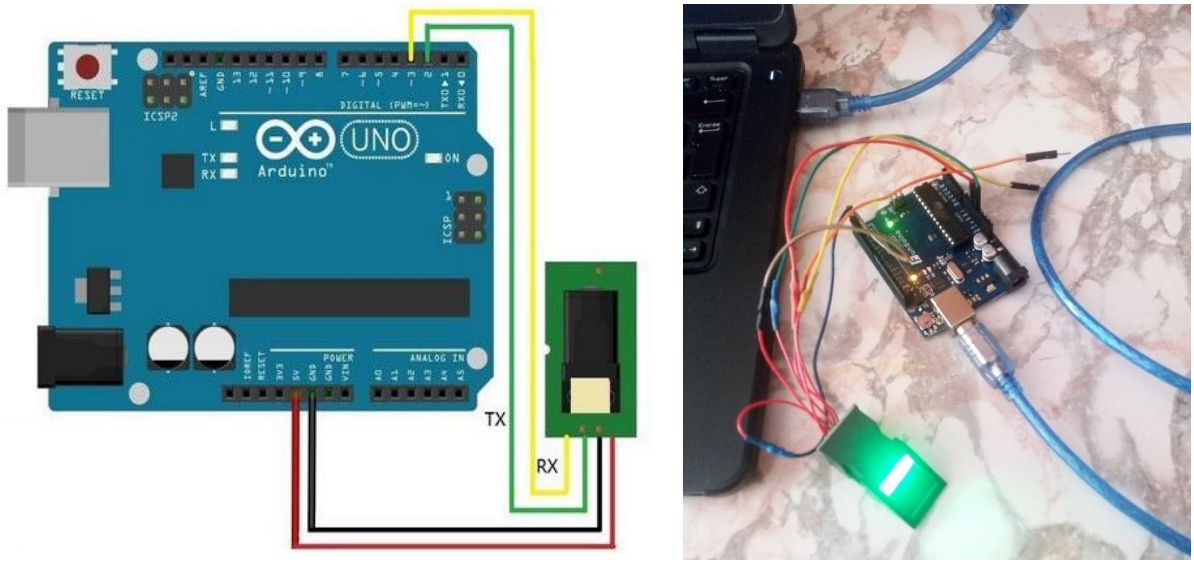


Figure III.9 : Schémas du montage du Lecture avec la carte Arduino.

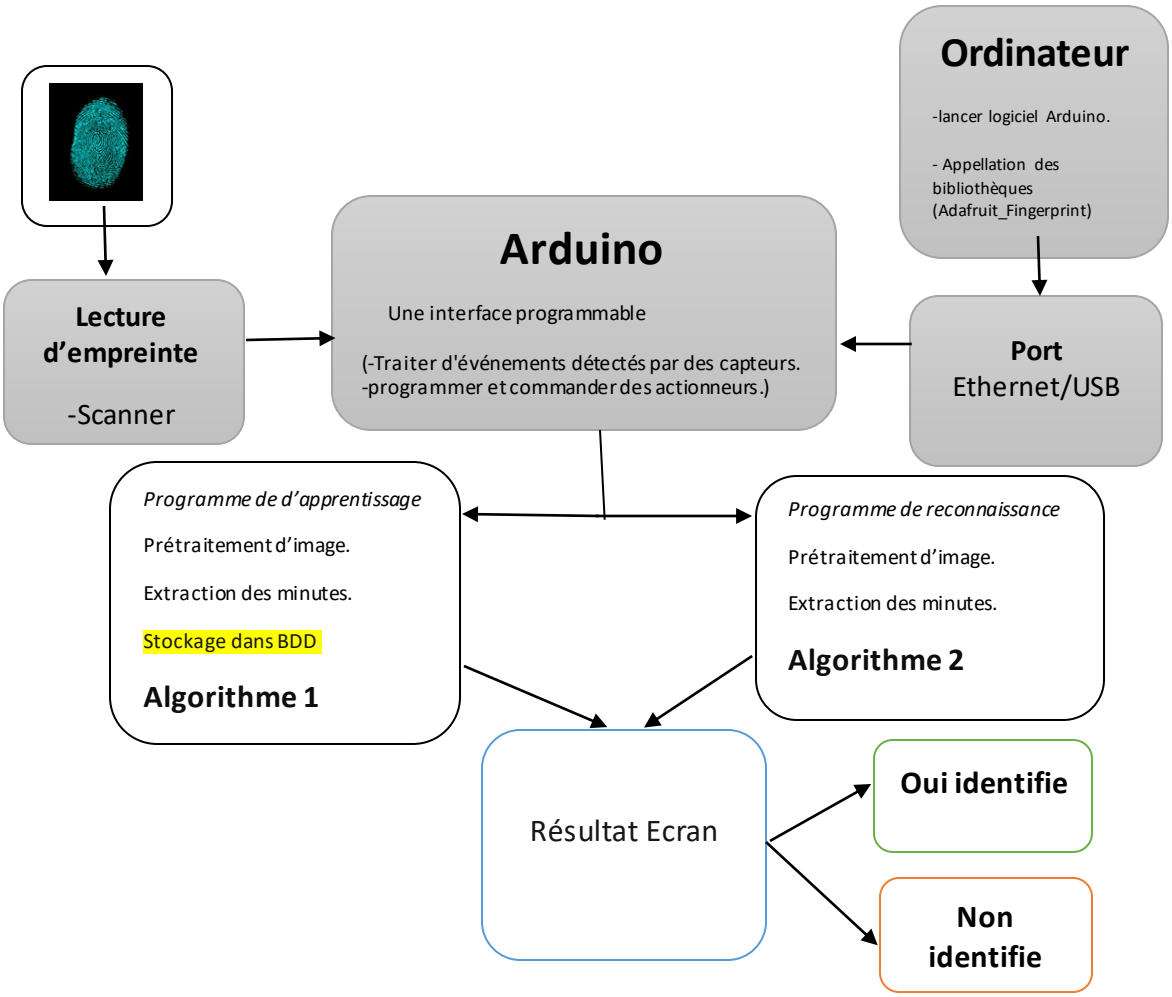


Figure III.10 : L'organigramme du programme du projet.

III.4.2.Algorithme (1) d'apprentissage (ANNEXE 1)

```
Début {
#Initialisation des ports
Si { (vérification.motdepasse ())
Alors
    Ecrire ("capteur empreinte trouvé") ;
Sinon
Ecrire (" capteur empreinte non trouvé ") ;}
Finsi
Void () {
#lecture_du_paramètre_du_capteur
#niveau_de_securité                    #capacité_d'empreinte
Ecrire ("lire le numéro") ;
Id==lire(1)
Ecrire ("Id, 1") ;
P==-1;
Tantque (P!=empreinte_OK)
P=empreinte.obtenir l'image ()
Switch (p)
Cas (empreinte OK):
Ecrire ("image prise") ;
Cas (pas d'empreinte) ;
Ecrire (".") ;              Cas (chute image) :
Ecrire ("erreur d'image") ;
Pause ;
Défaut ;
Ecrire ("erreur inconnue") ; }
Fin.
```

III.4.3.Algorithme (2) de reconnaissance : (ANNEXE 2)

```
Debut          {
Si     {(empreinte.  Verficationmotdepasse)
Alors      écrire ("capteur  empreinte  trouvé") ;
Sinon            écrire ("capteur  empreinte  non trouvé) ;
Tantque  (1)                                  delay(1) ;        }
#lecture_de_parametres_du_capteur
Si (empreinte.  Nombre de  modèles  ==0)
{       Ecrire ("  le capteur ne contient  aucune  donnée  d'empreinte  digitale") ;}
Sinon      lire  (en attente   validation  d'empreinte)
Ecrire ("le  capteur  contient  de donnée  d'empreinte") ;  }
Void ()   {
Obtenir  l'empreinte  digitale  Id ();        delay (50);   }
P=recherche  empreinte  ();
Si  (p==empreinte  OK)              {Ecrire  ("Empreinte  reconnue  avec  succès") ;   }
Sinon
        Si (P==empreinte  récepteur  de paquets)
{       Ecrire ("erreur  de communication  ") ;
Retour  P ;          }
Sinon          si (P==empreinte  non trouvé)
{       Ecrire ("Les  empreintes  digitales  ne correspondent  pas '") ;
Retour  P ;}
Sinon ecrire ("erreur  inconnu") ;
Retour  P ;}
#Trouvé_correspondance_  (match)
Lire  (trouvé  Id#)                 Ecrire  ("Id, #") ;
Lire  (avec la  confiance  de)          Ecrire  ("emprenite.confiance") ;
Retour  empreinte.empreinte  Id ;                    }
Fin.
```

III.5.Conclusion :

Dans ce chapitre, nous avons expliqué les deux parties essentielles de l'Arduino; (la partie matérielle et la partie de programmation) plus précisément. Nous avons également expliqué le principe de fonctionnement de la carte Arduino sans oublier ses caractéristiques. Après nous avons décrit la partie implémentation de notre application pour la reconnaissance des empreintes digitales. Il reste maintenant la validation et le déploiement de cette dernière.

Chapitre IV

Les résultats expérimentaux

IV. 1.Introduction:

Dans ce chapitre, nous présentons les résultats expérimentaux recueillis au cours de notre travail développé dans le cadre de la reconnaissance des empreintes digitales. On va créer notre base de données avec logiciel ‘Arduino ‘ et à l'aide du logiciel "SFGDemo" qu’on va utiliser pour montrer les images des empreintes digitales.

IV.2.L’apprentissage des empreintes digitales (enroll) :

Ouvrir **l’Arduino** –fichier –exemples –adafruit fingerprint sensor library – enroll.

La figure IV.1 présente l’interface d’enroll le code détaillé (ANNEXE 1) et commenté correspondant au montage :

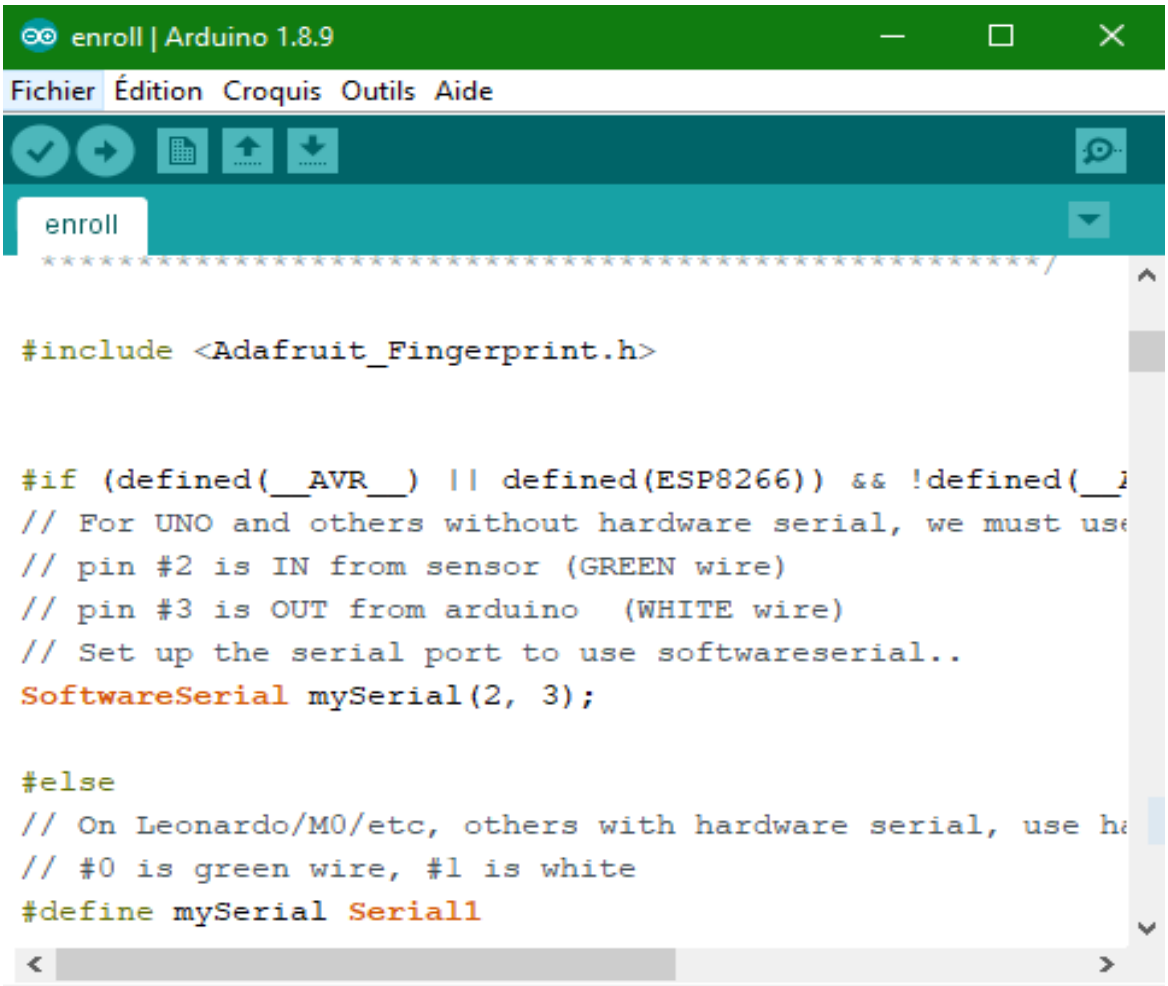


Figure IV.1 : interface enroll.

Le but de notre code va être de :

Inscrire ID (de 0 jusqu'‘ a 127) ; attendre d'un doigt valide pour inscrire sous le numéro déjà proposer ; prendre l'image et convertir la si il y a un problème dans la prise d'image le programme demande de retirer le doigt et placer à nouveau le même doigt enfin stocker dans la base des données.

Après avoir vérifié et téléverser le programme vers la carte Arduino, en ouvreront le moniteur série (onglet ‘outil'), puis en sélectionnant la bonne vitesse de communication (ici 9600 baud) (voir figure IV.2).

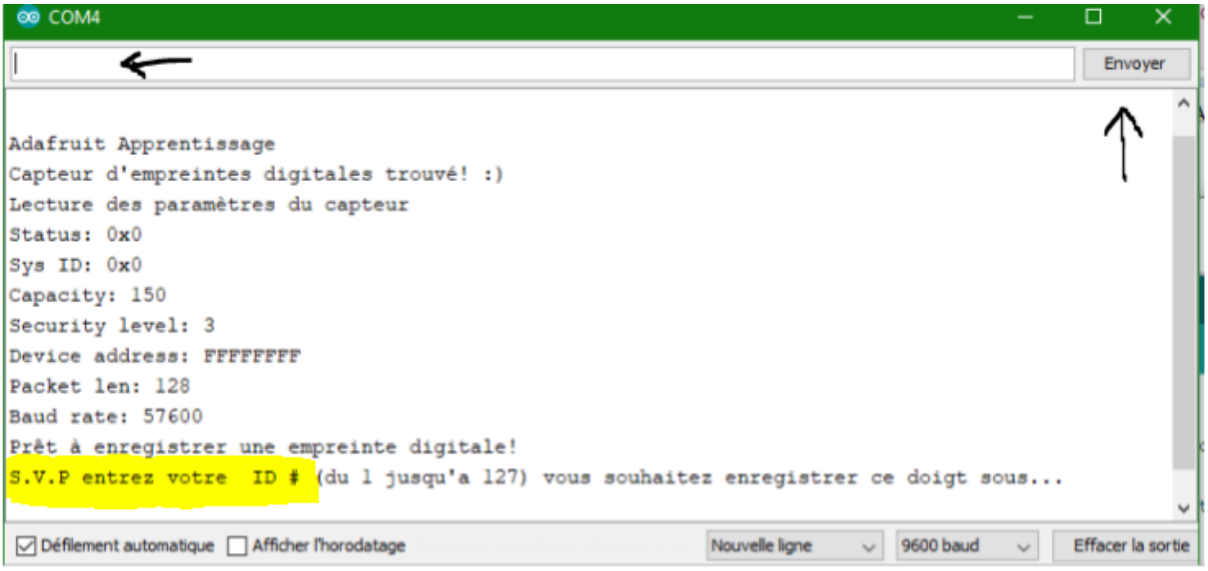


Figure IV.2 : Interface du moniteur série.

Nous allons maintenant enregistrer notre ID et l'envoyer (voir figure IV.2); le programme l'enregistrera. Ensuite après avoir entré l'empreinte digitale ; nouvel affichage sera comme la figure IV. 3 illustre.

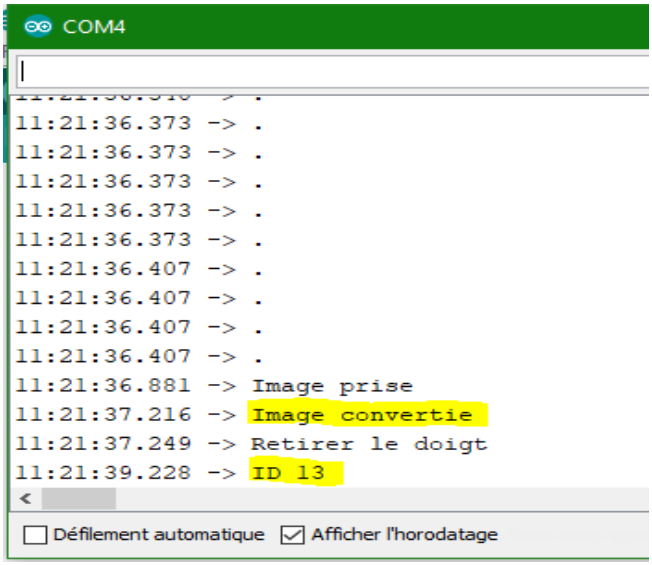


Figure IV.3 : l'enregistrement d'ID.

En cas l'image n'est pas bien prise ou déformée, le système vous demandera de placer le même doigt pour la deuxième fois. (Voir figure IV.4)

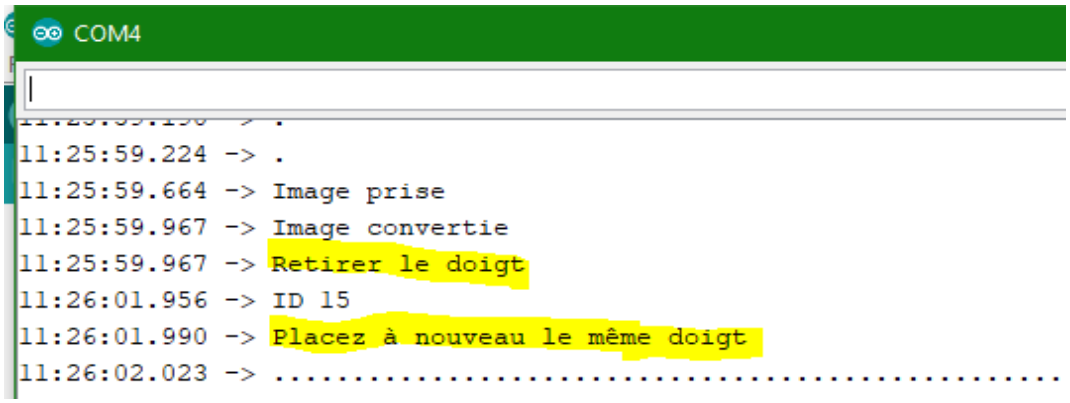


Figure IV.4 : le cas où l'image pas bien prise**.**

IV.3.La reconnaissance des empreintes digitales (fingerprint) :

Pour nous assurons que notre BDD (base de données) est bien fonctionné, nous suivons les mêmes étapes que nous avons suivre dans la partie d'apprentissage. La figure IV.5 présente l'affichage lorsque le capteur ne détecte pas une empreinte, le défilement automatique reste jusqu'une empreinte a capturée.

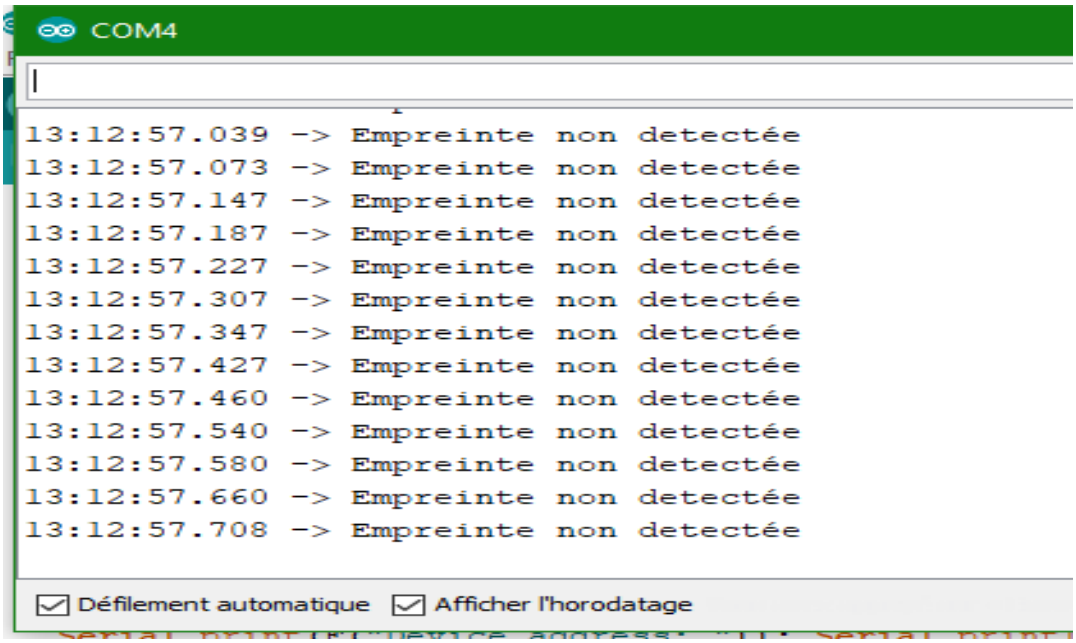


Figure IV.5 : défilement automatique.

Une fois que le système détecte et reconnaît l'empreinte digitale, il donne le numéro d'identification(ID) avec sa confiance et plus cette dernier est grande, meilleure est la correspondance (voir figure IV.6).

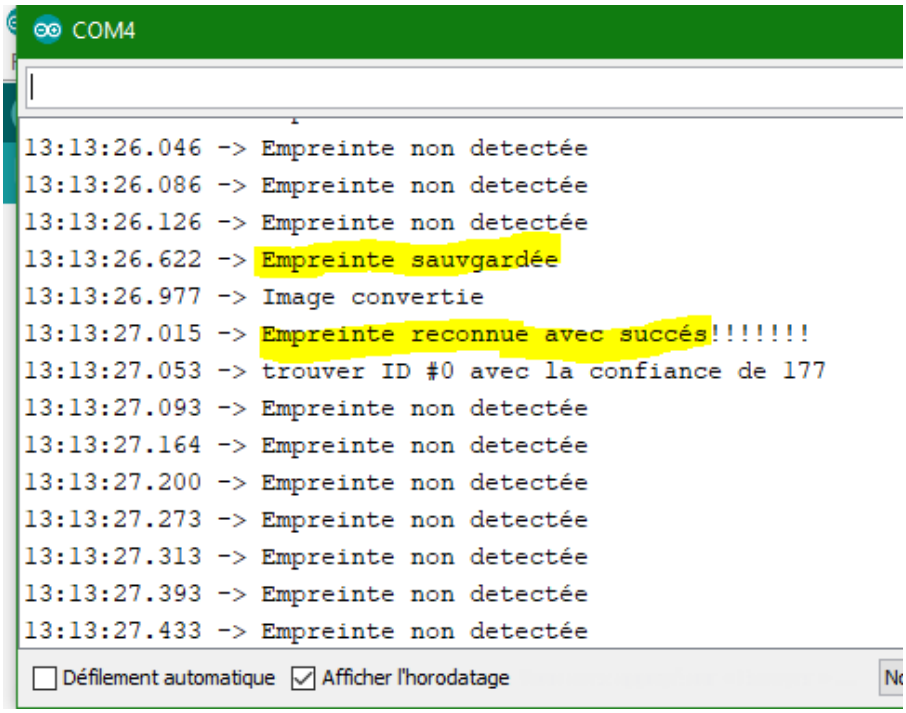
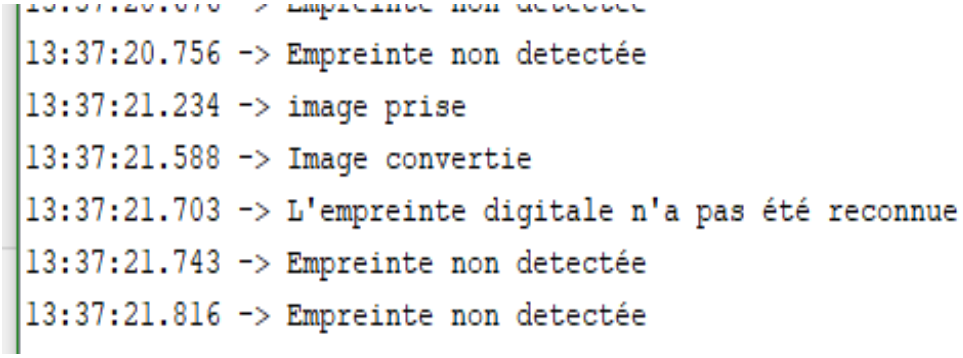


Figure IV.6 : la reconnaissance avec succès.

Et lorsqu'une empreinte digitale non enregistrée est saisie précédemment, le système ne le reconnait pas comme la figure suivante montre :



FigureIV.7: échec de reconnaissance.

IV.4.Le logiciel « SFGDemo » :

Le but d'utilisation le SFG :

Notre objectif du ce logiciel en bref, c'est afficher l'image de l'empreinte digitale et conserver la dans la base de donner.

IV.4.1.Le principe de fonctionnement du SFG :

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

Maintenir un programme vide "Blink" sur "Arduino"

```
//Blink code

//Code for finger print

Void setup () {}

Void loop () {}
```

Nous relions directement le capteur à (Tx/D1, Rx/D0) d'Arduino comme dans la figure IV.8.

Apres on va démarrer le logiciel SFGDemo et cliquer sur Ouvrir le périphérique dans le coin inférieur gauche. Nous sélectionnons le port correct COM 4.

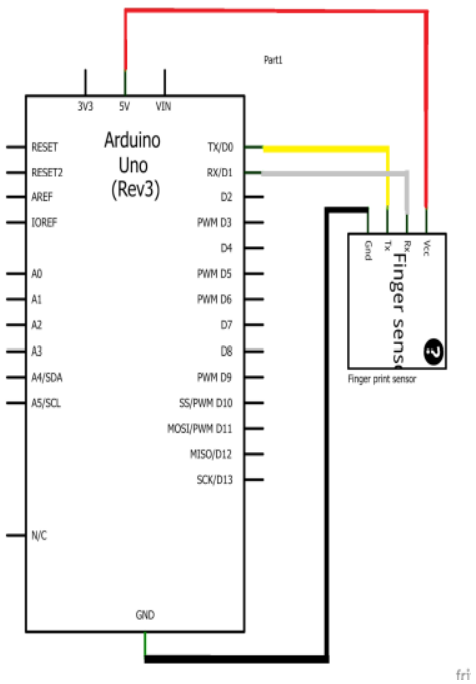


Figure IV.8 : schéma du montage de SFG.

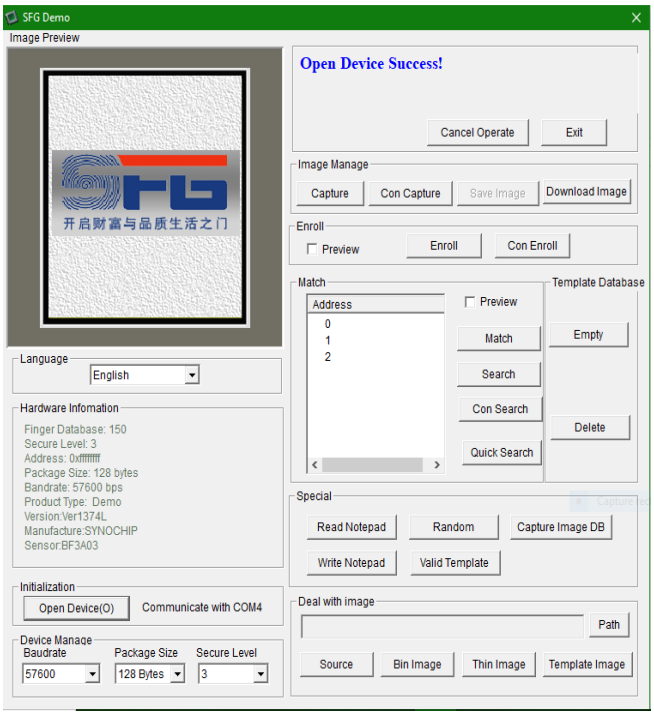
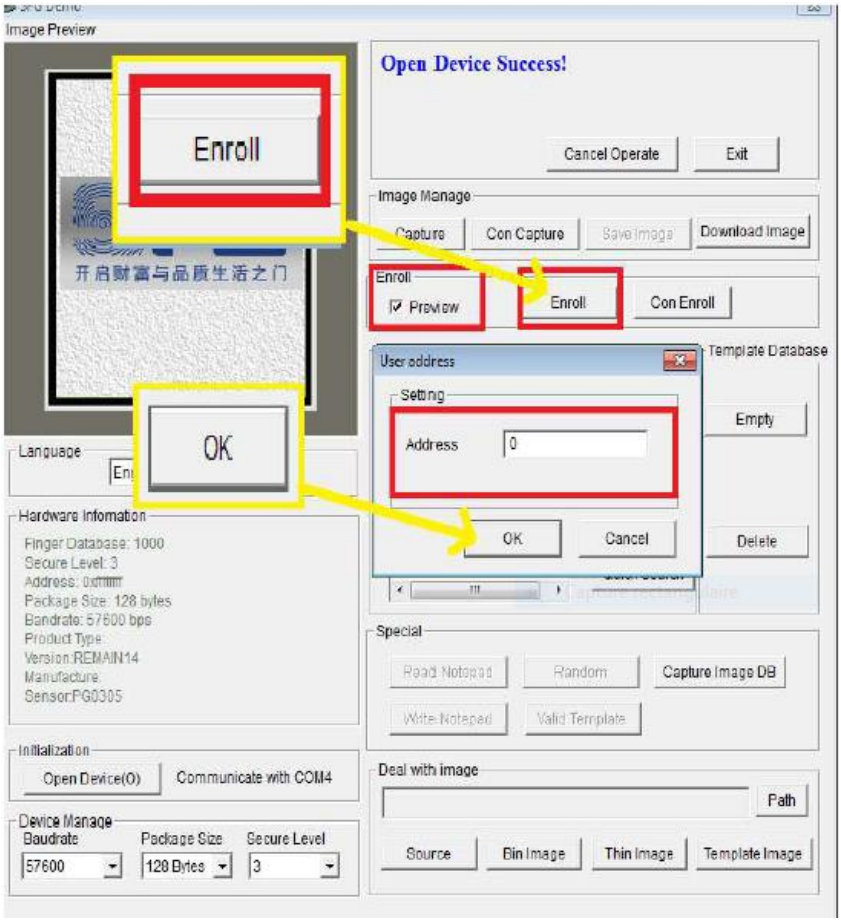


Figure IV.9 : interface le SFG.

Après on appuyant sur "OK", l'information sur le capteur, et même les déjà ajouté " apparaît sur Pour l'instant, nous allons ajouter une nouvelle empreinte que le capteur la reconnaître 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 on mettre le doigt sur le capteur, garder le sur le capteur et un message (Voir la figure IV.10) apparaît sur l'écran, va nous guidons à travers les étapes.

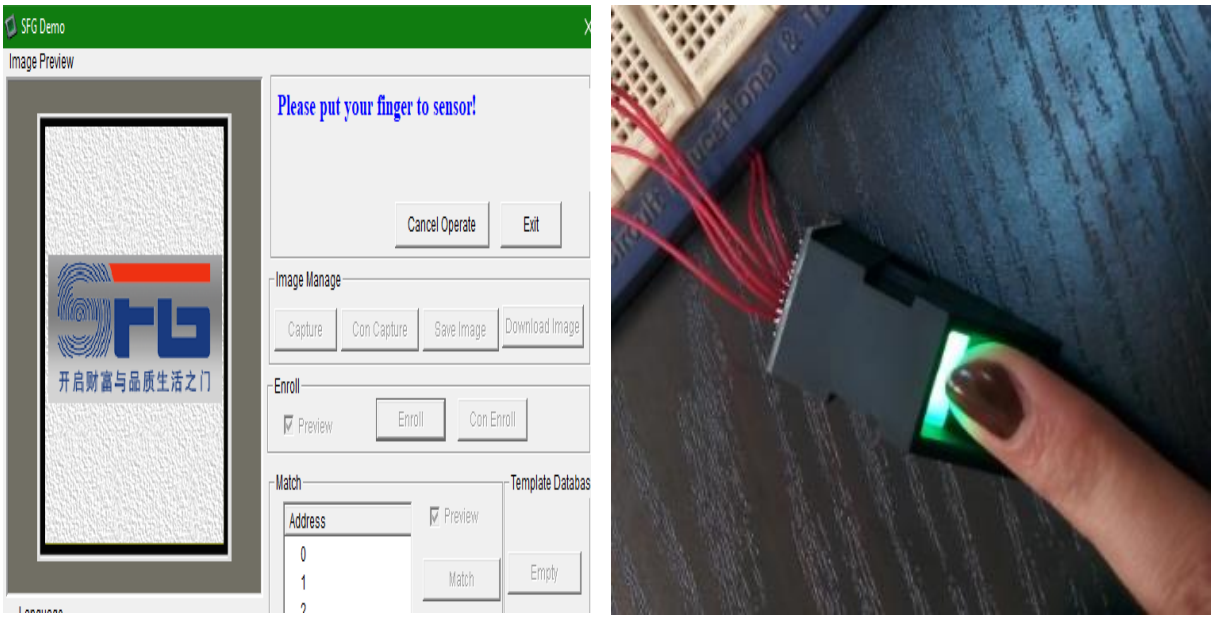


Figure IV.10 :l'entrée d'empreinte.

Après le scan du doigt en quelques secondes, nous pouvons voir un aperçu de l'image de notre empreinte digitale et cela a été confirmé avec succès dans un délai de 435ms comme notre exemple suivant : figureIV.11.

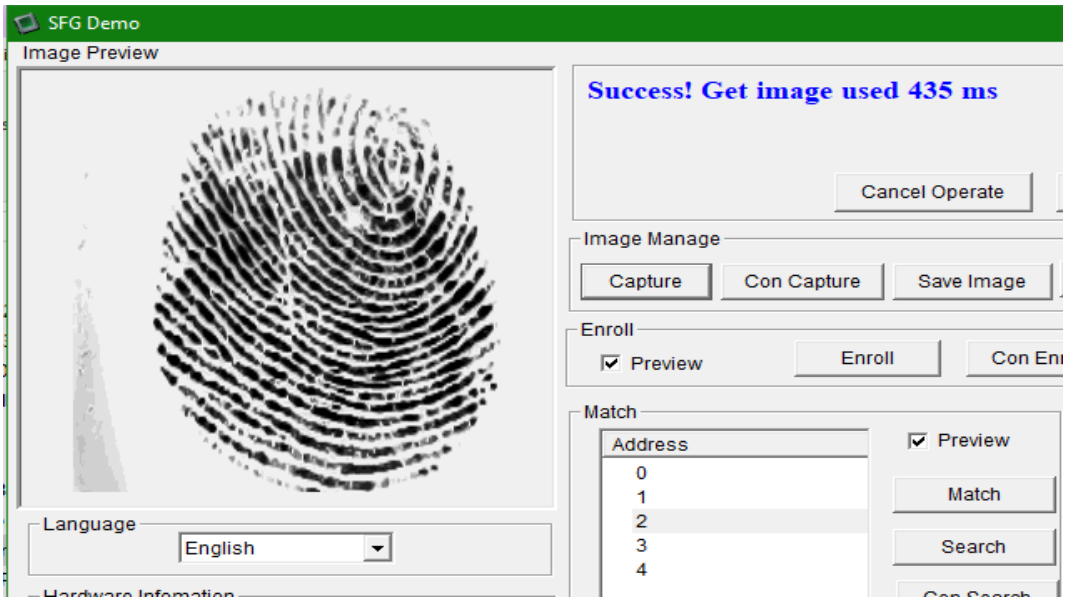
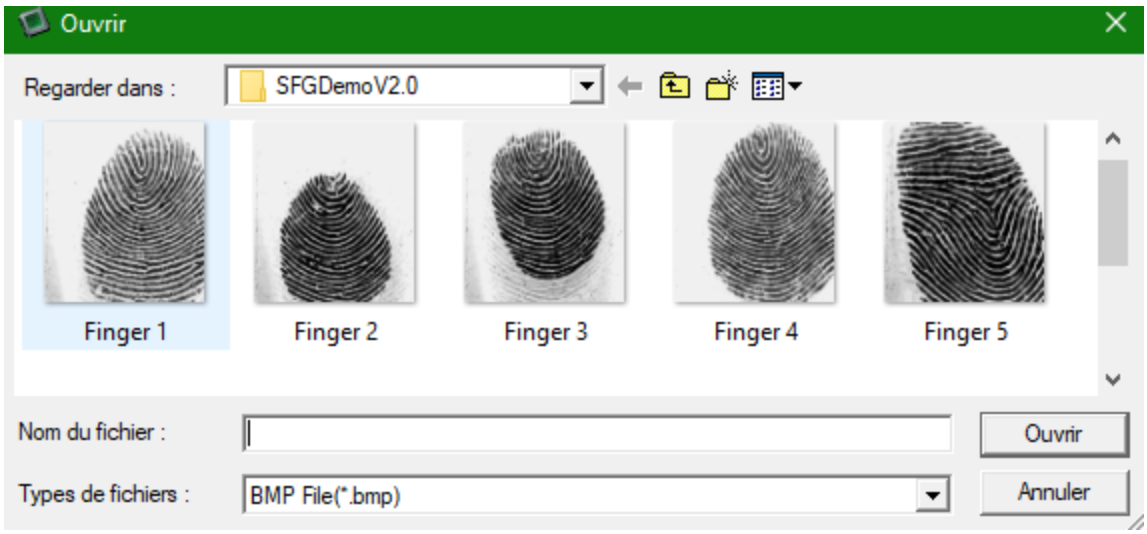


Figure IV.11 : empreinte sauvegardé avec succès.

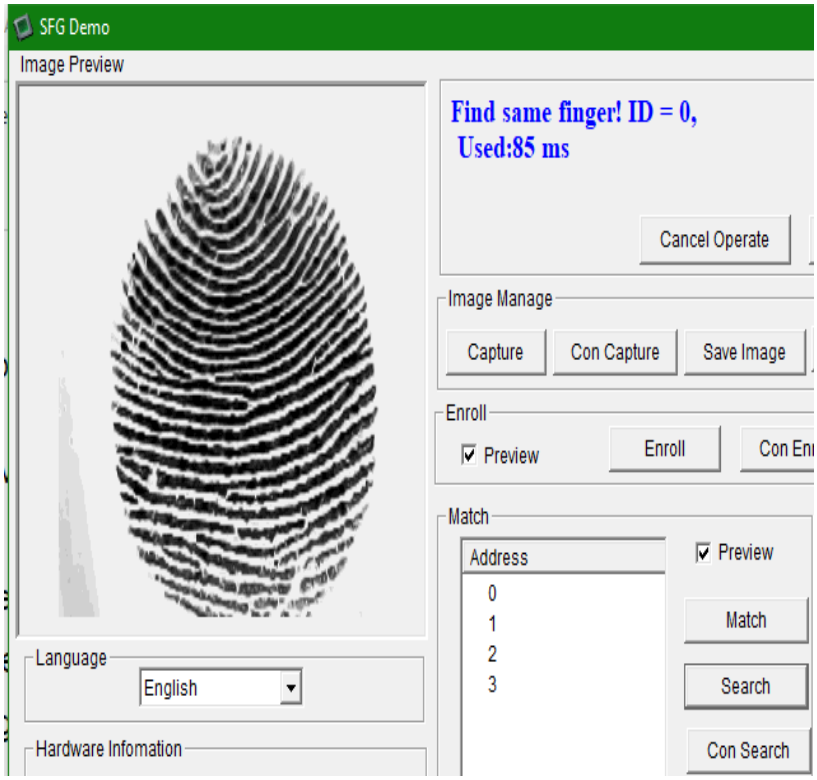
Beaucoup d'options que nous pouvons utiliser sur l'image obtenue dont : l'enregistrement, capturer, téléchargement. La figure suivante présente notre base des données (BDD) laquelle conservé dans un fichier.



FigureIV.12 : Base des données.

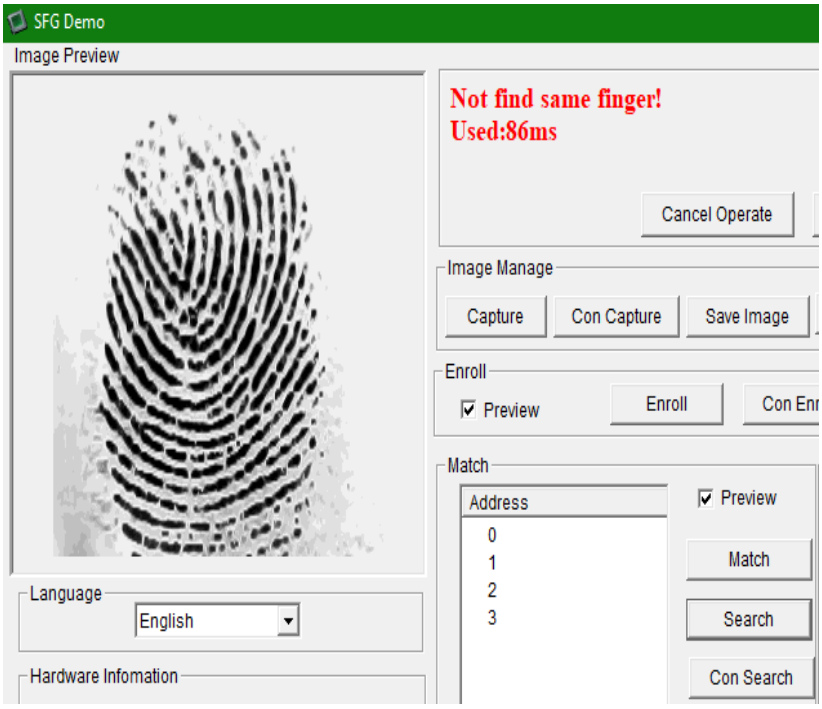
IV.4.2. l'identification :

Pour l'identification, on appuie sur "**search**", puis ont mis le doigt sur le capteur. Le logiciel nous donne si empreintes digitales existe et où il est stocké. La Figure IV.13 **(a)** nous montre que le système a reconnu notre ID.



(a)

(b) représente le cas où l'empreinte digitale n'a pas été entré et que le système ne la pas reconnue lors de la vérification, nous pouvons également enregistrer cette nouvelle empreinte dans la base de donnée



(b)

FigureIV.13: interface search (a) identifier (b) non identifier

IV.4.3. l'authentification:

ET maintenant l'authentification; pour l'objectif du confirmer que cet identifiant appartient à cette personne, nous sélectionnons l'adresse et nous choisissons ''Match'', nous plaçons le doigt sur le capteur pour prendre l'image puis elle est mise en correspondance avec l'identité précédemment définie si la même ; ce message va apparaitre ''Pass ! '' (Voir la figure IV.14).

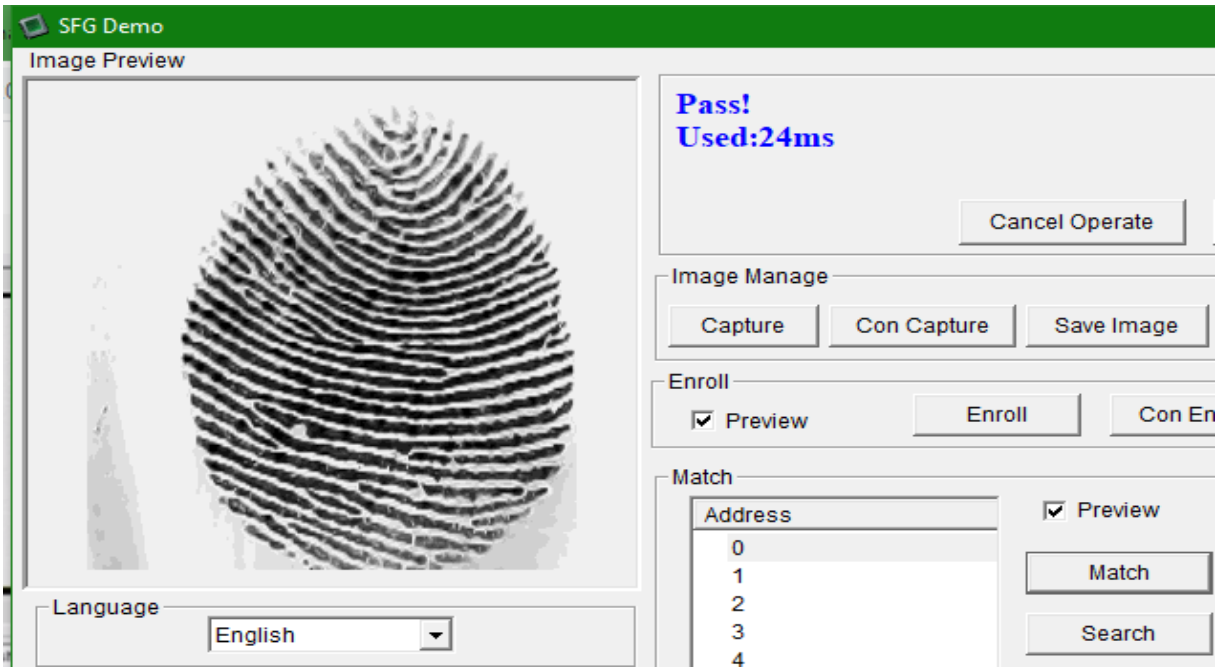


Figure IV.14: appariement d'identité.

Si non ; le message suivant (figure IV.15) nous indique qu'une correspondance a été rejetée (échec).

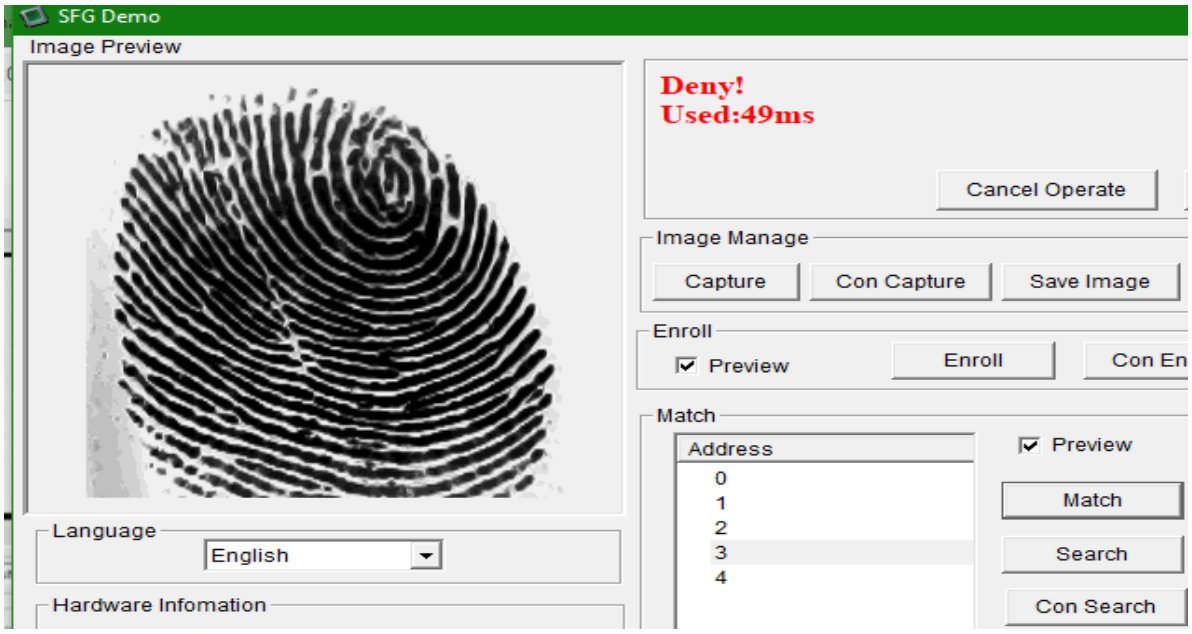
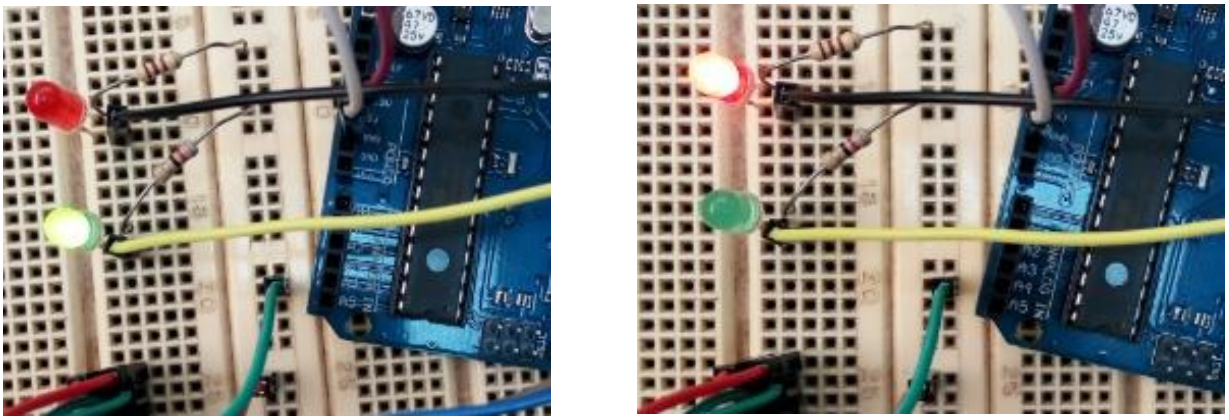


Figure IV.15: correspondance rejetée.

IV. 5.les LEDs :

Nous avons utilisé des LEDs pour indiquer la confirmation ou bien le rejet d'identité, Si l'empreinte est reconnue avec succès la LED vert s'allume sinon la LED rouge va s'allume (empreinte ne reconnue pas).

On peut utiliser ça comme un système du contrôle d'accès par empreinte digitale.



La figure IV.16 : indications des LEDs

IV.6. Conclusion:

Ce chapitre était consacré à l'évaluation des performances de notre travail en testant plusieurs configurations pour assurer la meilleure identification. Les résultats obtenus sont précis et clairs, nous avons fait la reconnaissance de l'empreinte avec succès et en plus l'Arduino permit à nous de développée bien notre système par : contrôler l'affichage et son délai, ajouter des accessoires électroniques comme notre exemple les LEDs.

Conclusion générale

La réalisation maquette issu de notre projet de fin d'études, nous a permis d'aborder différentes caractéristiques biométriques ainsi que leurs performances.

Au cours de ce mémoire, nous avons rappelé les différentes techniques de reconnaissance des empreintes digitales. Nous avons pu constater qu'elle a connu ces dernières années des progrès très importants, permettant désormais de faire face à la variabilité des empreintes digitales entre les individus. Dans ce travail, nous nous sommes focalisés sur la reconnaissance des empreintes digitales par apprentissage.

Nous estimons avoir réalisé un système répondant à l'objectif que nous nous sommes fixés au départ, à savoir la mise en œuvre d'un système permettant la reconnaissance d'individus et le contrôle d'accès ; ce système de reconnaissance d'empreinte digitale par une carte Arduino et un capteur optique d'empreinte nous avons permis de créer une base de données spécialement pour nous modifiable et changeable selon notre désir.

Aussi, ce travail nous a donné une opportunité de nous essayer et de nous familiariser à certains softwares et hardwares, méconnue ou mal maîtriser pour nous jusqu'à lors. C'est là que le système "Arduino" qui a nous donner la possibilité d'assemblé les performances de la programmation et l'électronique, plus précisément.

Enfin, nous espérons que ce projet sera développé dans le futur par l'ajout d'un système de pointage par reconnaissance d 'empreinte digitale ; surtout nos amis de IMSI, nous les encourageons à le faire.

ANNEXE 1

```
#include <Adafruit_Fingerprint.h>
Software Serial mySerial (2, 3);
 Adafruit Fingerprint finger = Adafruit Fingerprint (&mySerial);
 Uint8_t id;
 Void setup ()        {Serial. Begin(9600);
 White (! Serial); // For Yun/Leo/Micro/Zéro/...          Delay(100);
 Serial.println("\n\adafruit enregistrement du capteur d'empreintes digitales ");
 pinMode (11, OUTPUT);
 pinMode (12, OUTPUT);
 finger.begin(57600);
 If (finger.verifyPassword()) { Serial.println ("Capteur d'empreinte trouvé!");   }
 Else {   Serial.println ("Capteur NON trouvé :(");       white (1) {Delay(1) ;}   }
 Serial.println (F ("lecture des paramètres du capteur "));
 finger.getParameters ();
 Serial.print (F ("Status: 0x")); Serial.println (finger.status_reg, HEX);
 Serial.print (F ("Sys ID: 0x")); Serial.println (finger.system_id, HEX);
 Serial.print (F ("Capacity: ")); Serial.println (finger.capacity);
 Serial.print (F ("Security level: ")); Serial.println (finger.security_level);
 Serial.print (F ("Device address: ")); Serial.println (finger.device_addr, HEX);
 Serial.print (F ("Packet len: ")); Serial.println (finger.packet_len);
 Serial.print (F ("Baud rate: ")); Serial.println (finger.baud_rate);       }
 Uint8_t readnumber (void) {  uint8_t num = 0;
 While (num == 0) {      while (! Serial.available());     num = Serial.parseInt ();       }
 return num;     }

 Void loop ()                {
 digitalWrite (12, LOW);
 digitalWrite (11, LOW);

 Serial.println ("enregistrer une empreinte !");
 Serial.println ("S.V.P entrez votre ID # (du 1 jusqu'127) vous souhaitez enregistrer
 ce doigt...");       id = readnumber ();
 If (id == 0) {// ID #0 not allowed, try again!       return;   }
 Serial.print ("Apprentissage ID #");
 Serial.println (id);
 While (! GetFingerprintEnroll());   }
 Uint8_t getFingerprintEnroll () {      int p = -1;
 Serial.print ("En attente de validation du doigt pour s'inscrire comme #");
 Serial.println (id);
 While (p != FINGERPRINT_OK) {
 p = finger.getImage ();
 Switch (p) {
 Case FINGERPRINT_OK:
 Serial.println ("Image prise ");      break;
 Case FINGERPRINT_NOFINGER:
 Serial.println (".");    break;
 Case FINGERPRINT_PACKETRECIEVEERR:
```

```
Serial.println (" erreur de communication "); break;
Case FINGERPRINT_IMAGEFAIL:
Serial.println ("erreur d'imagerie "); break;
Default:
Serial.println ("erreur inconnue");
Break; }
// OK succès!
p = finger.image2Tz(1); switch (p) { case FINGERPRINT_OK:
Serial.println ("Image convertie");
break;
Case FINGERPRINT_IMAGEMESS: Serial.println ("Image trop désordonnée");
return p;
Case FINGERPRINT_PACKET: Serial.println (" erreur de communication");
return p;
Case FINGERPRINT_FEATUREFAIL:
Serial.println ("impossible de trouver les fonctionnalités d'empreintes digitales");
return p;
Case FINGERPRINT_INVALIDIMAGE:
Serial.println ("impossible de trouver les fonctionnalités d'empreintes digitales");
return p;
Default: Serial.println ("erreur inconnue"); return p; }
Serial.println ("Retirer le doigt");
Delay(2000); p = 0;
White (p != FINGERPRINT_NOFINGER) { p = finger.getImage (); }
Serial.print ("ID ");
Serial.println (id);
p = -1; Serial.println ("Placer à nouveau le même doigt ");
White (p != FINGERPRINT_OK) { p = finger.getImage ();
Switch (p) {
Case FINGERPRINT_OK:
Serial.println ("Image prise"); break;
Case FINGERPRINT_NOFINGER:
Serial.print ("."); break;
Case FINGERPRINT_PACKETRECIEVEERR:
Serial.println ("erreur de communication ");
break;
Case FINGERPRINT_IMAGEFAIL:
Serial.println ("erreur d'imagerie "); break; default:
Serial.println ("erreur inconnue");
break; } }

// OK succès!
p = finger.image2Tz(2);
Switch (p) {
Case FINGERPRINT_OK:
Serial.println ("Image convertie"); break;
Case FINGERPRINT_IMAGEMESS:
Serial.println ("Image trop désordonnée");
return p;
Case FINGERPRINT_PACKETRECIEVEERR:
```

```
 Serial.println ("erreur de communication ");
 return p;
 Case FINGERPRINT_FEATUREFAIL:
 Serial.println ("impossible de trouver les fonctionnalités d'empreintes digitales");
 return p;
 Case FINGERPRINT_INVALIDIMAGE:
 Serial.println ("impossible de trouver les fonctionnalités d'empreintes digitales");
 return p;
 Default:
 Serial.println ("erreur inconnue ");
 return p;}

// OK convertie!
Serial.print ("Création d'un modèle pour #");
Serial.println (id);
p = finger.createModel ();
If (p == FINGERPRINT_OK) {
    Serial.println ("impressions assorties!");
    digitalWrite (11, HIGH);
    Delay(2000);
    digitalWrite (12, LOW);}
 else if (p == FINGERPRINT_PACKETRECIEVEERR) {
          Serial.println ("erreur de communication ");
          return p;   }
 else if (p == FINGERPRINT_ENROLLMISMATCH) {
          Serial.println ("les empreintes digitales ne correspondent pas");
           digitalWrite (12, HIGH);
           Delay(2000);
          digitalWrite (11, LOW);
          return p;     }
 else {    Serial.println ("erreur inconnue");
            return p;}
            Serial.print ("ID "); Serial.println (id);       p = finger.storeModel (id);
 If (p == FINGERPRINT_OK) {
    Serial.println ("Stocké!");     }
 else if (p == FINGERPRINT_PACKETRECIEVEERR) {
       Serial.println ("erreur de communication ");
       return p;   }
 else if (p == FINGERPRINT_BADLOCATION) {
          Serial.println ("impossible de stocker à cet endroit");
          Return p;   }
 else if (p == FINGERPRINT_FLASHERR) {
          Serial.println ("Erreur d'écriture dans le flash");
          return p;   }
  else {Serial.println ("erreur inconnue");
          return p;   }
           return true;
}
```

ANNEXE 2

```
#include <Adafruit_Fingerprint.h>
Adafruit Fingerprint finger = Adafruit_Fingerprint (&mySerial);

Void setup () {
 Serial.begin(9600);
 while (!Serial);  // For Yun/Leo/Micro/Zero/...
 delay(100);
 Serial.println("\n\nAdafruit test de détection d'empreinte");
 pinMode(11,OUTPUT);
 pinMode(12,OUTPUT);

 // Set the data rate for the sensor serial port
 finger.begin(57600);
 delay(5);
 If (finger.verifyPassword()) {
   Serial.println ("capteur d'empreinte digitale trouvé!") ;}
else {
      Serial.println ("capteur NON trouvé :(");
      While (1) {delay(1) ;}  }
 Serial.println (F ("lecture des paramètres du capteur "));
 finger.getParameters ();
 Serial.print (F ("Status: 0x")); Serial.println (finger.status_reg, HEX);
 Serial.print (F ("Sys ID: 0x")); Serial.println (finger.system_id, HEX);
 Serial.print (F ("Capacity: ")); Serial.println (finger.capacity);
 Serial.print (F ("Security level: ")); Serial.println (finger.security_level);
 Serial.print (F("Device address: ")); Serial.println(finger.device_addr, HEX);
 Serial.print (F ("Packet len: ")); Serial.println (finger.packet_len);
 Serial.print (F ("Baud rate: ")); Serial.println (finger.baud_rate);

 finger.getTemplateCount ();
 If (finger.templateCount == 0) {
 Serial.print ("Sensor doesn't contain any fingerprint data. Please run the 'enroll' example.") ;}
 else {
     Serial.println ("En attente d'un doigt valide...");
     Serial.print ("Sensor contains "); Serial.print (finger.templateCount); Serial.println ("
templates");   }}

Void loop ()      {
 GetFingerprintID ();
 digitalWrite (12, LOW);
 digitalWrite (11, LOW);
 delay(50);          }

uint8_t getFingerprintID() {
Uint8_t p = finger.getImage ();
Switch (p)
```

```
{
Case FINGERPRINT_OK:
Serial.println ("Image prise");
break;
    Case FINGERPRINT_NOFINGER:
        Serial.println ("Empreinte NON détectée");
         return p;
    Case FINGERPRINT_PACKETRECIEVEERR:
        Serial.println ("erreur de communication");
         return p;
    Case FINGERPRINT_IMAGEFAIL:
         Serial.println ("erreur d'imagerie");
         return p;
        default:
        Serial.println ("erreur inconnue");
        return p;   }

 // OK succès!

     p = finger.image2Tz ();
 switch (p) {
  Case FINGERPRINT_OK:
    Serial.println ("Image convertie");
    break;
  Case FINGERPRINT_IMAGEMESS:
    Serial.println ("Image trop désordonnée");
    return p;
  Case FINGERPRINT_PACKETRECIEVEERR:
    Serial.println("  erreur de communication ");
    return p;
  Case FINGERPRINT_FEATUREFAIL:
    Serial.println ("Could not find fingerprint features");
    return p;
  Case FINGERPRINT_INVALIDIMAGE:
    Serial.println ("Could not find fingerprint features");
    return p;
  default:
    Serial.println("  erreur inconnue");
    return p;   }

 // OK converted!
 p = finger.fingerSearch ();
 If (p == FINGERPRINT_OK) {
  Serial.println ("Empreinte reconnue avec succès !!!");
  digitalWrite (11, HIGH);
  delay(2000);
  digitalWrite (12, LOW);

 }
```

```
else if (p == FINGERPRINT_PACKETRECIEVEERR) {
   Serial.println("  Erreur de communication");
   return p;   }

else if (p == FINGERPRINT_NOTFOUND) {
   Serial.println ("l'empreinte n'a pas été reconnue");
   digitalWrite(12, HIGH);
   delay(2000);
   digitalWrite (11, LOW);
   return p;   }
   else {
   Serial.println ("erreur inconnue");
   return p;
  }


  Serial.print ("trouver ID #"); Serial.print (finger.fingerID);
  Serial.print (" avec la confiance de "); Serial.println (finger. Confidence);

  return finger.fingerID;}

// returns -1 if failed, otherwise returns ID #
Int getFingerprintIDez () {
  Uint8_t p = finger.getImage ();
  If (p != FINGERPRINT_OK)  return -1;

  p = finger.image2Tz ();
  If (p != FINGERPRINT_OK)  return -1;

  p = finger.fingerFastSearch ();
  If (p != FINGERPRINT_OK)  return -1;

  // found a match!
  Serial.print ("trouver ID #"); Serial.print (finger.fingerID);
  Serial.print (" avec la confiance de "); Serial.println (finger. Confidence);
  return finger.fingerID;
}
```

Références bibliographiques

[1] https://www.researchgate.net/figure/Differentes-modalites-biometriques_fig1_43057044

Consulter le 01/04/2020 a 14 :30

[2] https://blogrecherche.wp.imt.fr/2016/03/07/empreintes-digitales-identite-bout-doigts/

Consulter le 01/04/2020 a 14 :58

[3] https://www.reseaux-telecoms.net/actualites/lire-la-biometrie-mise-sur-l-iris-les-veines- et-la-reconnaissance-faciale-18868.html

Consulter le 02/04/2020 a 10 :15.

[4] https://fr.lovepik.com/image-500770784/iris-recognition.html

Consulter le 19 /05/2020 a 15 :34

[5] Turk, M. A., & Pentland, A. P. (1991, January). Face recognition using eigenfaces. In *Proceedings. 1991 IEEE computer society conference on computer vision and pattern recognition* (pp. 586-587). IEEE Computer Society.

[6] Liu, S., & Silverman, M. (2001). A practical guide to biometric security technology. *IT Professional*, *3*(1), 27-32.

[7] https://business.lesechos.fr/directions-numeriques/02160723842-les-developpeurs-vont-ils-se-mettre-a-la-reconnaissance-vocale-111008.php

Consulter le 29 /05/2020 a 10 :13

[8] Hasnaoui, N. A. *LA RECONNAISSANCE AUTOMATIQUE DES EMPREINTES DIGITALES* (Doctoral dissertation).

[9] Pankanti, S., Jain, A., & Hong, L. (2000). Biometrics: Promising frontiers for emerging identification market. *Comm. ACM*, 91-98.

[10] BOUDJELLAL, S. (2014). *Détection et identification d'individus par méthode biométrique* (Doctoral dissertation, Université de Tizi Ouzou-Mouloud Mammeri).

[11] https://alexscott.io/eigen.html Consulter le 28/05/2020 a 16 :20

[12] BENCHENNANE, I. (2015). *Etude et mise au point d'un procédé biométrique multimodale pour la reconnaissance des individus* (Doctoral dissertation, University of sciences and technology in Oran).

[13] https://slideplayer.fr/slide/465924 consulter le 29/07/2020 a 20:13

[14] https://www.memoireonline.com/03/15/8967/Conception-et-mise-en-place-dune-plateforme-de-securisation-par-synthese-et-reconnaissance-biom.html

Consulter le 13/08/2020 a 01:33

[15] Jain, A. K., Prabhakar, S., & Pankanti, S. (2001, June). Twin test: On discriminability of fingerprints. In *International Conference on Audio-and Video-Based Biometric Person Authentication* (pp. 211-217). Springer, Berlin, Heidelberg.

[16] Yager, N., & Amin, A. (2004). Fingerprint verification based on minutiae features: a review. *Pattern Analysis and Applications*, *7*(1), 94-113.

[17] Xia, X., & O'Gorman, L. (2003). Innovations in fingerprint capture devices. *Pattern Recognition*, *36*(2), 361-369.

[18] Parrain, F. (2002). *Capteur intégré tactile d'empreintes digitales à microstructures piezorésistives* (Doctoral dissertation, Institut National Polytechnique de Grenoble-INPG).

[19] Bian, Z., Zhang, D., & Shu, W. (2002). Knowledge-based fingerprint post-processing. *International journal of pattern recognition and artificial intelligence*, *16*(01), 53-67.

[20] Kook, J. Design and Implementation of a Fingerprint-based Removable Storage Authentication System Using Particle Photon. *Memory*, *1*, 128KB.

[21] Soifer, V. A., Kotlyar, V. V., Khonina, S. N., & Skidanov, R. V. (1996, August). Fingerprint identification using the directions field. In *Proceedings of 13th International Conference on Pattern Recognition* (Vol. 3, pp. 586-590). IEEE.

[22] Leong Chung Ern, D. (2001, August). Ghazali Sulong, “Fingerprint classification approaches”, 6 th International. In *Symposium on Signal Processing and its Applications* (Vol. 1, pp. 347-350).

[23] Pankanti, S., Prabhakar, S., & Jain, A. K. (2002). On the individuality of fingerprints. *IEEE Transactions on pattern analysis and machine intelligence*, *24*(8), 1010-1025.

[24] Belguechi, R. (2006). *Reconnaissances des empreintes digitales par une approche hybride* (Doctoral dissertation, Master’s thesis, Ecole nationale superieure d’informatique, 2006.[cité p. 21, 177]).

[25] Stosz, J. D., & Alyea, L. A. (1994, October). Automated system for fingerprint authentication using pores and ridge structure. In *Automatic systems for the identification and inspection of humans* (Vol. 2277, pp. 210-223). International Society for Optics and Photonics.

[26] Woodward Jr, J. D., Horn, C., Gatune, J., & Thomas, A. (2003). *Biometrics: A look at facial recognition*. RAND CORP SANTA MONICA CA.

[27] http://www.appliedbiometrics.com/fileadmin/download/Article_Empreinte_Digitale.pdf

Consulter le 29/10/2020 a 01 :05

[28] http://www.univ-tebessa.dz/index.php?id_page=61

Consulter le 03/06/2020 a 13 :12

[29] Babler, W. (1991). Embryologic development of epidermal ridges and their configurations. *Birth defects original article series*, *27*(2), 95-112.

[30] http://onin.com/fp/fphistory.html Consulter le 13 /07/2020 a 14:22

[31] LE DIPLOME, D. M. (2016). *Caractéristiques Biométrique pour l'identification* (Doctoral dissertation, Université d'Oran).

[32] Maltoni, D., Maio, D., Jain, A. K., & Prabhakar, S. (2003). 2.6: Fingerprint Scanners and their Features.

[33] https://bentek.fr/2-arduino-uno/ Consulter le 20/10/2020 a 13:29

[34] http://www.mon-club-elec.fr/pmwiki_reference_arduino/pmwiki.php?n=Main.DebuterInstallationWindows

Consulter le 15/10/2020 a 12 :33

[35] https://fr.aliexpress.com/item/32826567205.html Consulter le 15/11/2020 a 10 :50

[36] https://www.arduino.cc/ consulter le 03/10/2020

[37] https://www.hugedomains.com/domain_profile.cfm?d=malvida&e=com

Consulter le 07/10/2020 a 13 :30