

République Algérienne Démocratique et Populaire

Ministère de l'Enseignement Supérieure et de la Recherche Scientifique

Université Abderrahmane Mira

Faculté de Technologie



Département d'Automatique, Télécommunication et d'Electronique

Projet de Fin d'Etudes

Pour l'obtention du diplôme de Master

Filière : Electronique

Spécialité : Instrumentation

Thème

**Réalisation d'un système de pointage des étudiants basé sur la technologie
RFID**

Préparé par :

IBERRAKEN Sabrina

Dirigé par :

Mme Mezzah

Examiné par :

Mr Hadji

Mr Tighzert

Année universitaire: 2021/2022

Remerciement

Tout d'abord, je tiens à remercier Allah le tout puissant et miséricordieux; de m'avoir guidé à travers le chemin et aider à accomplir ce travail, de m'avoir donné la force, le courage mais aussi la santé.

Je remercie mes parents, mes deux frères, ainsi que tous les membres de ma famille; pour leurs amours, confiances, patiences, aides, mais aussi leurs soutiens et encouragements tout au long de mon parcours.

Mes remerciements et mes profondes reconnaissances en particulier, à mon grand frère, pour son amour, son aide précieux, son soutien, et sa présence.

Mes sincères respects et remerciements, à mon encadrante Mme Mezzah; pour sa disponibilité et sa présence, son aide, sa gentillesse, ses compétences et son orientation.

Un grand remerciement à tous les ingénieurs des laboratoires du département de Génie Électrique et d'Électronique

Mes gratitudes aux membres du jury, qui ont accepté d'évaluer ce travail..

Merci

Dédicace

C'est avec une profonde gratitude et sincères mots que je tiens à dédier ce travail à:

Mes parents, aucune dédicace ne saurait exprimer mon respect et mon amour éternel que je leurs porte, ma considération pour toutes ces années de sacrifices qu'ils ont consentis pour moi, depuis ma naissance, durant mon enfance et même à l'âge adulte. Je tiens à les remercier pour leur amour, patience, soutien, encouragement et surtout les valeurs nobles et éducation.

À mes deux frères; des exemples, des anges gardien, source d'amour ; de courage et de persévérance. Je tiens à les remercier pour leurs présences et soutiens. Que le bonheur soit leur compagnon et la réussite leur logis.

Je rends hommage à mes grands-parents, paix à leurs âmes ; le fruit de leurs conseils précieux, eux qui m'ont éclairé le chemin et m'ont montré la voix. Puisse Allah le tout puissant, les avoir en sa sainte miséricorde.

À mes tantes et oncles, à ma deuxième famille et à ma très chère sœur ; je tiens à les remercier d'avoir été toujours là pour moi, leurs soutiens et encouragements, leurs grands cœurs et qualités. Aucune dédicace ne saurait être assez éloquente pour exprimer ce qu'ils méritent. Que la joie soit leur allié et que le succès embellisse leur vie.

À tous les membres de ma famille sans exception chacun(e) par son prénom, eux qui n'ont jamais cessé de m'encourager et de me conseiller, cette humble dédicace ne saurait exprimer mon respect et ma reconnaissance.

Enfin, je dédie ce travail à moi, à mon travail, mes efforts, mes ambitions, ma patience, courage et volonté.

À toutes les personnes, qui par un geste, une phrase ou même un mot, m'ont donné la force et le courage de continuer.

A tous les héros de la santé: médecins, infirmiers, chirurgiens...à tous les fictifs sans exception, et aux victimes de la pandémie, paix à leurs âmes.

Ce projet a pour but de réaliser un système de pointage pour la gestion de présence des étudiants à l'aide d'Arduino UNO, basé sur RFID RC522 qui comporte deux composants, à savoir une étiquette RFID et un lecteur RFID. En intégrant divers composants, afficheur LCD, Horloge RTC DS1307, et Secure Carte numérique (carte SD), un système de présence portable peut être configuré, permettant d'identifier de manière unique la fréquentation des étudiants à l'université en fonction de son type d'étiquette RFID.

Les étudiants n'ont qu'à placer leur carte d'identité sur le lecteur et leur présence sera prise immédiatement. Avec la capacité d'horloge en temps réel du système, les présences prises seront plus précises puisque le temps de présence pris sera enregistré. Le système peut être connecté à l'ordinateur via le port USB et stocker les présences prises dans la base de données.

Mots clés

Arduino, RFID RC522, LCD, système de pointage, carte SD, RTC DS1307, Tag RFID, Lecteur RFID, UID.

TABLE DES MATIERES

LISTE DES FIGURES

LISTE DES TABLEAUX

LISTE DES ABREVIATIONS

INTRODUCTION GENERALE

1. Contexte de l'étude du projet.....	1
2. Enoncé du problème.....	1
3. Objectif.....	2
4. Méthodologie.....	2

CHAPITRE I : GENERALITES SUR LES SYSTEMES DE POINTAGES

1.1. Introduction.....	4
1.2. Fonctionnalités d'un système de pointage.....	4
1.3. Différents systèmes d'identifications.....	6
1.3.1. Système de pointage basé sur un code-barres.....	6
1.3.2. Système de pointage biométrique.....	8
1.3.3. Système de pointage basé sur la technologie RFID.....	16
1.4. Conclusion.....	23

CHAPITRE II : SYSTEME DE POINTAGE BASE SUR LA RFID

2.1.Introduction.....	24
2.2.Description du système de pointage RFID proposé.....	24
2.3.Outils de développement matériel.....	26
2.3.1. Présentation de la carte Arduino.....	26
2.3.2. Carte Arduino Uno.....	30
2.3.3. Module RFID.....	37
2.3.4. Afficheur LCD.....	41
2.3.5. Horloge RTC.....	44
2.3.6. Lecteur carte SD.....	46
2.3.7. Buzzer.....	48
2.4.Outils de développement logiciel.....	49
2.4.1. Logiciel de programmation Arduino IDE.....	49
2.5.Conclusion.....	52

CHAPITRE III : IMPLEMENTATION ET VALIDATION

3.1.Introduction.....	53
3.2.Implémentation matérielle.....	53
3.3.Implémentation logiciel.....	67
3.4.Test et validation.....	76
3.5.Conclusion.....	80

Table des matières

CONCLUSION GENERALE.....	81
REFERENCES	
ANNEXE	

LISTE DES FIGURES

Figure 1.1: Exemple de code à barres.....	6
Figure 1.2: Différents lecteurs de code à barres.....	7
Figure 1.3: Architecture du système à code-barres.....	8
Figure 1.4: Différents systèmes biométriques.....	9
Figure 1.5: Architecture générale d'un système biométrique.....	10
Figure 1.6: (a) Echantillons d'empreintes, (b) Pointeuse biométrique par empreinte digitale.....	12
Figure 1.7: (a) Géométrie de la main, (b) Pointeuse biométrique par géométrie de la main.....	13
Figure 1.8: (a) Image de l'iris, (b) Différentes pointeuses biométriques par reconnaissance de l'iris.....	14
Figure 1.9 : (a) reconnaissance faciale, (b)Différentes pointeuses biométriques par reconnaissance faciale.....	15
Figure 1.10 : Reconnaissance vocale.....	15
Figure 1.11 : Reconnaissance par la signature.....	16
Figure 1.12 : Symbole de la RFID.....	16
Figure 1.13: Composants d'un système RFID.....	17
Figure 1.14: Composants du tag RFID.....	18
Figure 1.15 : Différents supports de tags RFID, (a) schéma synoptique, (b) tag nu, (c) porte clefs, (d) transplantable comparé à un grain de riz, (e) bracelet en face du lecteur.....	18
Figure 1.16: Différents lecteurs RFID.....	19
Figure 1.17: Communication entre lecteur et tag RFID.....	20
Figure 1.18: Architecture d'un système RFID.....	21
Figure 1.19: Exemples d'applications de la RFID.....	23
Figure 2.1: Carte Arduino.....	27
Figure 2.2 : Types de carte Arduino.....	29

Figure 2.3 : Composants de la carte Arduino UNO.....	32
Figure 2.4 : Broches du microcontrolleur ATmega328.....	35
Figure 2.5 : Système RFID.....	37
Figure 2.6 : Etiquettes RFID.....	38
Figure 2.7 : Lecteur RFID RC522.....	38
Figure 2.8 : Broches du module RFID RC522.....	40
Figure 2.9 : Afficheur LCD.....	41
Figure 2.10: Broches de l’Afficheur LCD.....	43
Figure 2.11: Horloge RTC DS1307.....	44
Figure 2.12: Broches du module RTC DS1307.....	45
Figure 2.13 : Module carte SD.....	46
Figure 2.14: Broches du module carte SD.....	47
Figure 2.15: Buzzer.....	48
Figure 2.16: Logiciel Arduino.....	49
Figure 2.17: Constitutions de l’Arduino IDE.....	49
Figure 2.18 : (a) Fonction Setup(), (b) Fonction Loop().....	50
Figure 2.19 : Structure générale du logiciel Arduino.....	51
Figure 3.1 : Schéma synoptique du projet.....	54
Figure 3.2 : Schéma électronique RFID et Arduino.....	56
Figure 3.3 : Cablage RFID et Arduino.....	56
Figure 3.4 : Schéma électronique Arduino et LCD.....	58
Figure 3.5 : Schéma électronique Arduino, LED, Buzzer.....	59
Figure 3.6 : Schéma électronique Arduino et carte sd.....	60
Figure 3.7 : Câblage Arduino et carte SD.....	61
Figure 3.8 : Schéma électronique Arduino et RTC DS1307.....	62
Figure 3.9: Schéma électronique du montage global avec FRITZING.....	63
Figure 3.10 : Circuit global du système de pointage.....	64
Figure 3.11 : Interface de la pointeuse.....	65
Figure 3.12: Vu du circuit à l’intérieure de l’interface.....	66

Figure 3.13: Organigramme de l'intégration RFID (codage).....	68
Figure 3.14: Led verte allumée.....	69
Figure 3.15: Affichage “You are on time” sur le moniteur série.....	69
Figure 3.16: Led rouge allumée.....	70
Figure 3.17: Affichage “You are Late” sur le moniteur série.....	70
Figure 3.18: Fichier DATA1 existe.....	71
Figure 3.19: Fichier DATA1.....	71
Figure 3.20: Caractères stockés dans le fichier DATA1.....	72
Figure 3.21: Affichage de la date et de l’heure sur LCD.....	72
Figure 3.22: Affichage de la date et de l’heure sur le moniteur série.....	73
Figure 3.23: Affichage du message de bienvenu sur LCD.....	77
Figure 3.24: Affichage de la date/heure sur LCD.....	77
Figure 3.25: Led verte allumée et affichage de l’heure d’arrivée de l’étudiant.....	78
Figure 3.26: Affichage du statut « On time » de l’étudiant.....	79
Figure 3.27: Led rouge allumée et affichage de l’heure d’arrivée de l’étudiant.....	79
Figure 3.28: Affichage du statut « Late » de l’étudiant.....	80

LISTE DES TABLEAUX

Tableau 1.1 : Types de systèmes biométriques.....	11
Tableau 1.2 : Différentes fréquences de la RFID.....	22
Tableau 2.1 : Caractéristiques techniques de la carte Arduino UNO.....	31
Tableau 2.2 : Caractéristiques techniques de l'ATMega328.....	35
Tableau 2.3 : Caractéristiques techniques du module RFID.....	39
Tableau 2.4 : Caractéristiques techniques de l'afficheur LCD.....	41
Tableau 2.5 : Caractéristiques techniques de l'horloge RTC DS1307.....	45
Tableau 2.6 : Caractéristiques techniques du module carte SD.....	47
Tableau 2.7 : Caractéristiques techniques du buzzer.....	48
Tableau 3.1 : Liste des composants utilisés.....	53
Tableau 3.2 : Connexion entre les broches Arduino et RFID.....	55
Tableau 3.3 : Connexion entre les broches Arduino et LCD.....	57
Tableau 3.4 : Connexion entre Arduino, led, buzzer.....	58
Tableau 3.5 : Connexion entre Arduino et carte sd.....	59
Tableau 3.6 : Connexion entre l'Arduino et RTC DS1307.....	61

LISTE DES ABREVIATIONS

LCD : Liquid cristal display / Ecran à cristaux liquides.

LED: Light Emitting Diode / diode electroluminescent

IDE : Integrated Development Environment (Environnement de développement intégré).

FIRMWARE : Programme intégré dans un matériel informatique.

RFID: Radio Frequency Identification

HF: Haute Fréquence

RTC: Real-Time Control

SPI: Serial Peripheral Interface

UHF : Ultra haute fréquence

BF: Basse Fréquence

HF: Haute Fréquence

USB: Universal Serial Bus

ICSP: In Circuit Serial Programming

I2C: Inter Integrated Circuit Bus

SCL: Serial Clock

SDA: Serial Data

PWM : Pulse Width Modulation (Modulation de largeur d'impulsion)

INTRODUCTION GENERALE

1. Contexte de l'étude du projet

Avec les progrès technologiques, la gestion de la fréquentation des foules devient plus pratique. Certaines méthodes ont été développées pour faciliter le processus d'enregistrement des présences et le système numérique de présence des élèves en fait partie. Il existe plusieurs types de systèmes de reconnaissance de présence qui sont déjà sur le marché, tels que le système de présence basé sur RFID, le système de présence biométrique, les algorithmes de détection de visage en temps réel et le système de présence basé sur des codes-barres. Comprendre le principe de fonctionnement de chaque système de présence existant est un atout pour improviser le système actuel.

2. Énoncé du problème

La fréquentation irrégulière des étudiants devient l'un des problèmes à l'université car leur comportement peut affecter les performances académiques. En disposant de ce système de présence portable, leurs présences seront strictement (facilement) enregistrées et analysées pour éviter d'être empêchées de passer l'examen final.

Fondamentalement, les conférenciers demanderont aux étudiants à signer manuellement la feuille de présence chaque moment où ils assistent à un cours ou de simplement appeler leur nom en conséquence. Aussi courant que cela puisse paraître, un tel système manque d'automatisation, où un certain nombre de problèmes peuvent surgir. Cela inclut le temps inutilement consommé par les élèves, mais aussi certains élèves peuvent signer par erreur ou intentionnellement un autre nom de l'étudiant. En outre, le papier peut être déchiré, égaré ou perdu pour être enregistré manuellement dans le système de base de données. Avec le système automatisé, toutes les données seront sécurisées et stockées dans la base de donnée.

Au lieu d'utiliser la méthode manuelle, les étudiants peuvent simplement toucher leurs cartes matricielles sur le lecteur de cartes afin d'enregistrer leur présence en classe. De nombreuses recherches ont été menées auparavant pour étudier les principaux objectifs de ce système ainsi que ses avantages.

Un lecteur de carte RFID est nécessaire pour lire et fournir toutes les informations concernant la présence. Une fois l'étiquette et le lecteur scannés, les informations concernant le nom et le numéro d'identification de l'élève seront affichées sur le lecteur pour vérifier leur présence en classe.

3. Objectif

L'objectif de ce projet est de développer un système de pointage pour étudiants qui est un système de présence numérique basé sur l'identification par radiofréquence (RFID) afin d'enregistrer la présence des étudiants en classe. La RFID est choisie car cette application est largement utilisée dans les systèmes de gestion des présences, elle peut raccourcir le temps par rapport à la méthode actuelle qui enregistre manuellement la présence des élèves.

Les principaux objectifs de notre projet sont :

- Comprendre le fonctionnement des méthodes de système de présence existantes.
- Ressembler les données nécessaires du système de présence RFID
- Réaliser un prototype de dispositif de pointage portable proposé et valider ses fonctionnalités.

4. Méthodologie

Dans ce projet, nous allons mettre en œuvre un système de présence basé sur la technologie RFID, à l'aide de la carte électronique Arduino UNO basée sur le microcontrôleur ATmega328P, et du module RFID MFRC522 constitué d'un lecteur et d'étiquettes.

Puisque l'Arduino UNO ne possède pas d'horloge interne, il n'est donc pas capable de retourner ni l'heure ni la date courante, alors on va utiliser un module horloge en temps réel RTC DS1307. Nous allons utiliser un écran LCD qui affichera des messages précis, des informations, ainsi que l'heure et la date, également une carte SD pour enregistrer les informations des étudiants.

On utilise le logiciel Arduino IDE, Chaque fois que nous semons n'importe quelle carte au système matériel, le lecteur obtient les données des cartes RF avec numéros d'identifications unique pour chaque étiquette. Si l'étudiant montre sa carte à l'heure, il est en à temps, donc une led verte s'allume. S'il arrive en retard alors une led rouge s'allume, et l'écran LCD connecté au microcontrôleur va afficher le prénom et le statut de l'étudiant.

Et pour cela nous allons décomposer ce travail en trois chapitres, au chapitre 1 nous présentons une introduction sur les systèmes d'identifications, en particulier le système RFID, dans le chapitre 2 nous nous concentrons sur l'objectif principal de notre projet, et les différents outils matériels et logiciels pour la conception d'un tel système. Le chapitre 3 est pour la réalisation et conception de notre pointeuse.

CHAPITRE I

GENERALITES SUR LES SYSTEMES DE POINTAGE

1.1. Introduction

Le système de pointage est un système utilisé pour suivre les heures d'arrivées et de départs d'une personne en particulier, appliqué dans les industries, les écoles, les universités ou les lieux de travail. Le système de présence basé sur la technologie RFID et le système de présence basé sur la biométrie, a réduit l'implication humaine et les erreurs.

Ainsi, dans ce chapitre, nous allons présenter en profondeur le système de pointage basé sur la technologie RFID, ainsi qu'une étude sur les systèmes biométriques, code à barre est également discutée, notamment en termes d'architectures et de fonctionnalités.

1.2. Fonctionnalités d'un système de pointage

Le logiciel de gestion de pointage/présence pour les universités est une partie importante du système de gestion scolaire qui utilise un système biométrique ou une carte d'accès pour conserver un enregistrement précis et rapide de la présence de chaque élève et offre des enregistrements et des résumés en temps opportun si nécessaire.

Ce logiciel ne sert pas seulement à prendre les présences, mais il possède également de nombreuses autres fonctionnalités, ce qui en fait une solution incontournable dans de nombreuses universités.

Bien qu'il soit nécessaire de noter qui est dans la classe et qui n'y est pas, cela prend beaucoup de temps productif, ce qui n'est bénéfique ni pour l'enseignant ni pour les étudiants, ce scénario est la raison pour laquelle une application de gestion des présences est requise. L'application s'occupera de la présence et fera gagner du temps à la fois à l'enseignant et à l'élève en toute simplicité.

Cela présente de nombreuses fonctionnalités et avantages étonnants, qui sont expliqués en détail ci-dessous : [1]

- ✓ **Gestion des horaires :** Une bonne gestion des horaires est ici plus qu'une nécessité, le logiciel aidera à gérer l'emploi du temps avec précision, et bien qu'il y ait un petit changement dans le calendrier, il est immédiatement notifié aux enseignants concernés sans aucune interférence humaine. Le logiciel vérifiera rapidement la disponibilité des professeurs et organisera d'éventuels cours en conséquence.

- ✓ **Le tableau de bord :** Ce tableau de bord est l'endroit où tous les événements, la fréquentation, le calendrier, etc. tout est présent. L'administration de l'école peut ajouter les détails qu'elle souhaite et personnaliser ce tableau de bord en fonction de ses besoins.
- ✓ **Rationalisation du processus d'admission :** Presque tous les sites Web des établissements d'enseignement ont un processus d'admission en ligne où les étudiants peuvent postuler uniquement via le site Web, il n'est pas facile pour le service d'admission de se tenir au courant du processus d'admission. Il est impossible de répondre manuellement à chaque demande qui arrive sur le site.

C'est dans ce scénario que le logiciel de gestion des présences vous aidera. L'ensemble du processus d'admission est automatisé et les étudiants sont guidés à chaque étape de la marche à suivre par le logiciel lui-même.

- ✓ **Gestion des présences :** Ce logiciel de gestion prend la présence en classe automatiquement, Le système aide également les administrateurs de l'école à gérer et à suivre les heures d'arrivées et de départs du personnel et des étudiants.

Il existe trois types de mécanismes de présence ou de prise de participation qui sont impliqués dans la prise de présence du public :

- RFID
- Biométrie
- Application mobile scolaire

- ✓ **Gestion des absences :** Ce système aide à gérer les absences, il devrait être facile de voir d'un coup d'œil qui est présent, qui n'est pas présent.
- ✓ **Dossier étudiant :** Les fichiers des étudiants peuvent être stockés en toute sécurité. La performance aux examens, les rapports de devoirs...etc. Chaque aspect de l'étudiant peut être téléchargé dans le logiciel.
- ✓ **Dossiers d'examen :** La gestion des examens est une autre tâche importante qui est présente avant la direction de l'école. Cette application de gestion des présences aidera à la bonne organisation des dossiers d'examen.
- ✓ **Une plus grande sécurité :** Ce système garantit que seuls les étudiants et les employés disposant d'informations d'identification valides peuvent accéder à l'établissement.

Toutes les données collectées seront stockées dans la base de données cloud. La manipulation de cette base de données est impossible. A part l'administration, on ne peut pas changer ou faire quoi que ce soit, et il en va de même pour les étudiants.

1.3. Différents systèmes d'identifications

1.3.1. Système de pointage basé sur un code à barres

Un code-barres est une représentation lisible utilisés pour stocker des bits de données, et toutes ces données peuvent être automatiquement collectées et stockées dans une base de données.

Il existe deux types de représentations de données de codes à barres, des codes à barres linéaires et matriciels : [2]

- ✓ **Codes-barres linéaires** : "unidimensionnel" appelé 1D, font référence à l'espacement des lignes parallèles.
- ✓ **Codes-barres matriciels** : Un code-barres bidimensionnelle appelé 2D, est une manière de représenter des informations. Il est similaire à un code-barres linéaire (unidimensionnel), mais peut représenter plus de données par unité de surface.



Figure 1.1: Exemple de code à barres

Le système de pointage basé sur un code à barres est un logiciel qui utilise un scanner de code-barres pour enregistrer et maintenir la présence des individus.

Le matériel principal qui doit être utilisé est le lecteur de code-barres, ce lecteur est utilisé pour lire un code-barres, nous avons différents types d'appareils tels que : [2]

- ✓ **Lecteur de type stylo** : Les lecteurs de type stylo se composent d'une source lumineuse et d'une photodiode placées l'une à côté de l'autre dans la pointe d'un stylo ou d'une baguette.
- ✓ **Lecteur CCD** : Les lecteurs CCD utilisent un réseau de centaines de minuscules capteurs de lumière alignés dans la tête du lecteur. Chaque capteur mesure l'intensité de la lumière immédiatement devant lui.
- ✓ **Appareils photo pour smartphones** : Les appareils photo des smartphones ont la capacité de lire à la fois les codes QR et de scanner un code-barres pour afficher des informations sur les produits.
- ✓ **Numériseur portatif** : Il s'agit d'un scanner avec une poignée et généralement un bouton de déclenchement pour allumer la source lumineuse.
- ✓ **Lecteur automatique** : Un équipement de back-office pour lire les documents à code-barres à grande vitesse.



Figure 1.2: Différents lecteurs de code à barres

Architecture et fonctionnement d'un système codes-barres

Les scanners de codes-barres se composent de trois parties différentes : le système d'éclairage, le capteur et le décodeur.

Un lecteur de code-barres est nécessaire pour identifier les données représentées par chaque code à barres en utilisant un faisceau lumineux et numériser directement au code à barres. La carte d'un individu ou d'un étudiant sera scannée à l'aide du lecteur, en éclairant le code avec une lumière rouge, qui est ensuite convertie en texte correspondant.

En effet, le capteur du lecteur de code-barres détecte la lumière réfléchiée par le système d'éclairage (la lumière rouge) et génère un signal analogique, qui sera convertie en un signal numérique pour ensuite l'envoyer à l'ordinateur.

Le décodeur interprète ce signal, valide le code-barres à l'aide du chiffre de contrôle et le convertit en texte [2]. Les données capturées (qui sont généralement le numéro d'identification personnel de chaque individu) seront stockées dans une base de données de l'ordinateur hôte [3]

La figure ci-dessous montre l'architecture du système codes-barres :

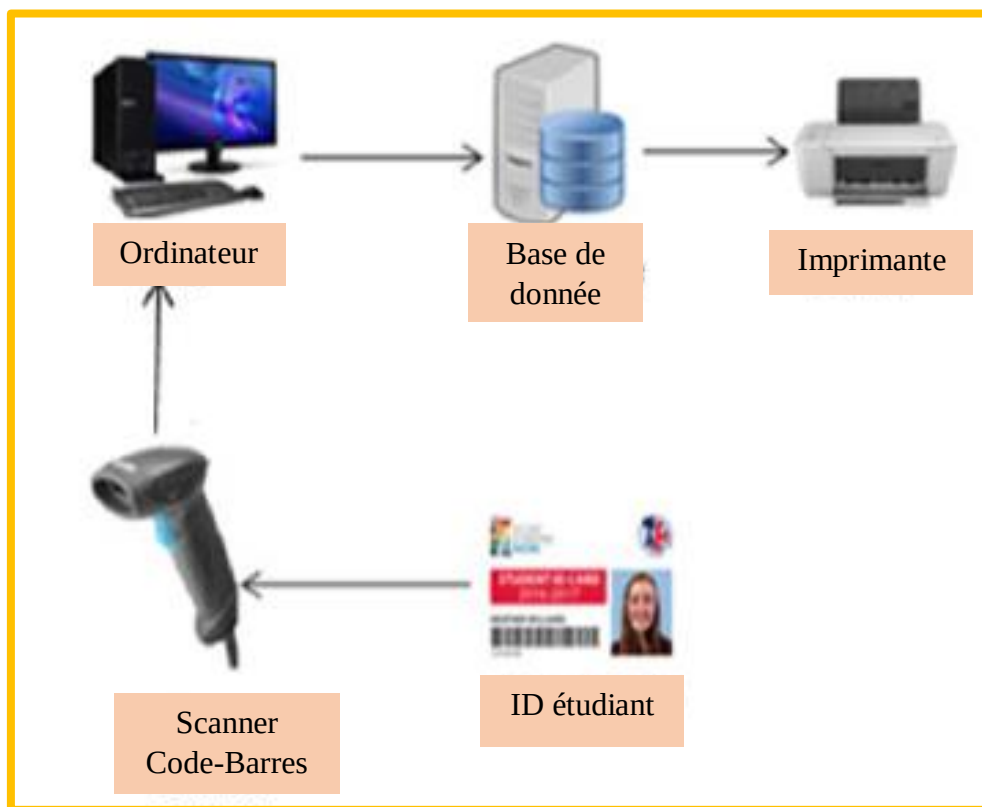


Figure 1.3: Architecture du système à code-barres

1.3.2. Système de pointage biométrique

La biométrie est un ensemble de technologies qui exploitent des caractéristiques humaines physiques ou comportementales telles que l'empreinte digitale, la signature, l'iris, la voix, le visage, ou la démarche pour différencier des personnes.

Cela a été confirmé par Mr Frédéric MASSICOTTE dans sa recherche intitulée 'La biométrie, sa fiabilité et ses impacts sur la pratique de la démocratie libérale' « La biométrie est la mesure des caractéristiques physiques d'un individu, que ce soit ses empreintes digitales, la forme de son visage ou encore son ADN » [4]

Selon le CLUSIF (Club de la Sécurité des systèmes d'Information Français.) « La biométrie est la science qui étudie à l'aide de mathématiques, les variations biologiques à l'intérieur d'un groupe déterminé » [5]

D'après la RAND Public Safety and Justice (Woodward J.D. & al., Biometrics, A Look at Facial Recognition, Documented Briefing prepared for the Virginia State Crime Commission.) « Toute caractéristique physique ou trait personnel automatiquement mesurable, robuste et distinctif qui peut être employé pour identifier un individu ou pour vérifier l'identité qu'un individu affirme » [6]



Figure 1.4: Différents systèmes biométriques

Un système biométrique est essentiellement une reconnaissance de formes, système qui effectue une identification personnelle en déterminant l'authenticité d'un comportement physiologique ou comportemental spécifique caractéristique possédée par l'utilisateur.

Les technologies biométriques doivent idéalement posséder plusieurs caractéristiques : [7]

- ✓ **L'unicité:** Cela signifie que chaque attribut biométrique doit varier énormément d'une personne à l'autre.

- ✓ **Quanti fiabilité** : Cela signifie que les caractéristiques doivent être mesurées de façon quantitative et leur obtention doit être facile (c'est-à-dire que deux personnes ne peuvent posséder exactement la même caractéristique).
- ✓ **L'acceptabilité par la population** : Bien acceptée par les utilisateurs du système.
- ✓ **L'universalité** : Toutes les personnes doivent avoir des caractéristiques biométriques.
- ✓ **Permanence** : Il est requis pour chaque caractéristique ou trait qui est enregistré dans la base de donnée du système doit être constante pendant une certaine période de temps.
- ✓ **Performances** : L'identification doit être précise et rapide.

Architecture d'un système biométrique

Un système biométrique peut fonctionner en mode enrôlement ou en mode vérification ou/et en mode identification: la vérification de l'identité conduit à l'identification et la preuve de l'identité conduit à l'authentification :

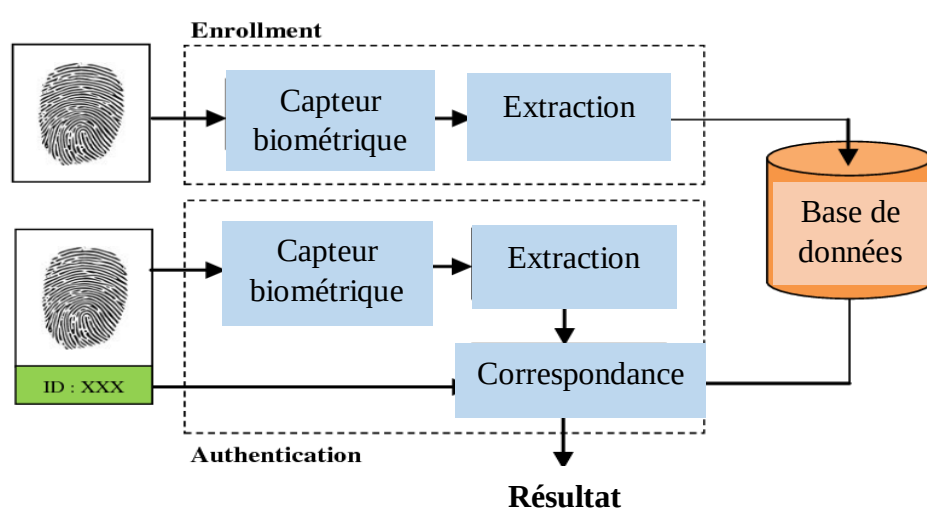


Figure 1.5: Architecture générale d'un système biométrique

- ✚ **Le mode d'enrôlement** : C'est une phase d'apprentissage, qui permet de recueillir des informations biométriques sur les personnes à identifier. Les caractéristiques biométriques des individus sont saisies par un capteur biométrique, puis représentées sous forme numérique (signatures), et enfin stockées dans la base de données.

✚ **Identification (Un à plusieurs):** La biométrie peut être utilisée pour déterminer l'identité d'une personne même à son insu ou consentement, en comparant sa signature avec les signatures de tous les utilisateurs dans la base de données. [8]

✚ **Vérification (Un à Un):** La biométrie peut également être utilisée pour vérifier l'identité d'une personne en comparant ses données biométriques capturées à des données spécifiques déjà enregistrées dans la base de données. [8]

Types du système de pointage biométrique

Il existe deux types de système biométrique:

Analyse biométrique physiologique

Elle est basée sur l'identification de traits physiques particuliers qui, pour toute personne, sont uniques et permanents. Cette catégorie regroupe la reconnaissance des empreintes digitales, l'iris, la forme de la main, les traits du visage, le réseau veineux de la rétine. [8]

Analyse biométrique comportementale

Ce système biométrique est basé sur l'identification et l'analyse de certains comportements d'une personne, il comprend la reconnaissance vocale, la dynamique de frappe au clavier, la dynamique de signature, la manière de marcher. [8]

Tableau 1.1 : Types de systèmes biométriques

- Reconnaissance d'empreinte digitale

La reconnaissance d'empreinte digitale est un type de biométrie physique. Pour cette méthode d'authentification, un lecteur d'empreinte digitale est utilisé pour authentifier les données. Avec un scanner, l'utilisateur devra juste appliquer son doigt ou le guider à travers un capteur.



Figure 1.6: (a) Echantillons d'empreintes, (b) Pointeuse biométrique par empreinte digitale

- Identification par géométrie de la main

La géométrie de la main est utilisée pour le contrôle d'accès physique, ainsi que pour le pointage horaire, 90 points caractéristiques de la main sont nécessaires pour identifier une personne telles que la longueur et la largeur des doigts.

Il suffit à l'utilisateur de poser sa main sur une platine, et une caméra infrarouge prend l'image de la main sous deux angles différents, afin d'obtenir un rendu en trois dimensions.

L'image scannée est convertie en un modèle numérique et associé à un code. Quand une personne passe sa main devant le lecteur, celle-ci cherche le code dans la base de données et procède à une comparaison.

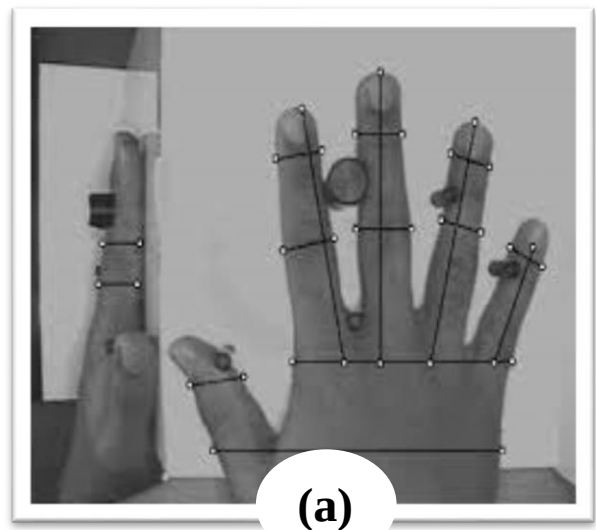


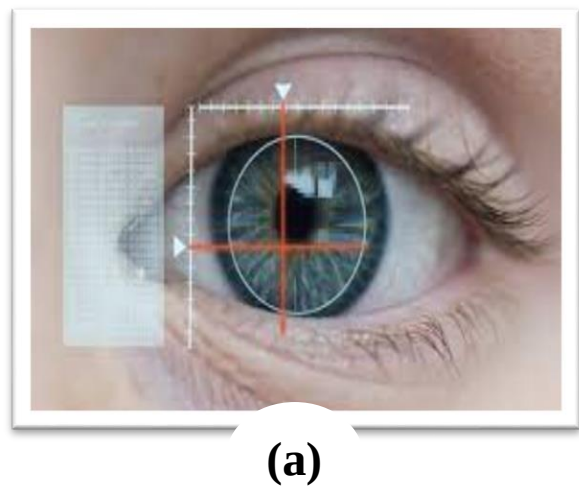


Figure 1.7: (a) Géométrie de la main, (b) Pointeuse biométrique par géométrie de la main

- Identification biométrique par IRIS

Ce système d'authentification biométrique est l'un des plus fiables, extrêmement précis, l'image de l'iris est réalisée avec des scanners dédiés, capturée à l'aide d'une caméra proche des infrarouges.

En effet, lorsque la pupille est localisée, suivi de la détection de l'iris et des paupières, les paupières et les cils sont exclues pour laisser partie qu'à l'iris, ce dernier est divisé en blocs et convertie en valeurs numériques représentant l'image.



Afin de vérifier l'identité, la mise en correspondance avec les données collectées est effectuée avec les mêmes méthodes.

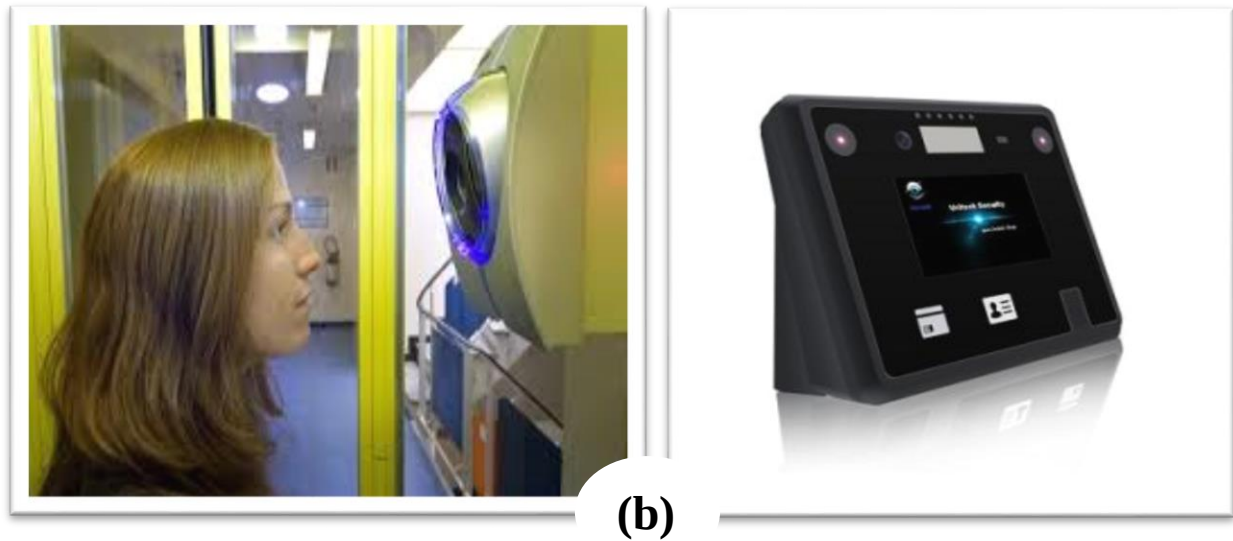


Figure 1.8: (a) Image de l'iris, (b) Différentes pointeuses biométriques par reconnaissance de l'iris

- Reconnaissance faciale

Les approches pour faire face à la reconnaissance faciale sont basées sur l'emplacement et la forme des attributs du visage comme les yeux, les sourcils, le nez, les lèvres et le menton.

On utilise un module d'acquisition d'images avec une caméra afin de confirmer l'identité d'une personne sur la base des données disponibles.

Il suffit à la personne de se positionner devant l'appareil, puis une localisation automatique du visage sera effectuée et l'envoi à un logiciel pour la numériser.

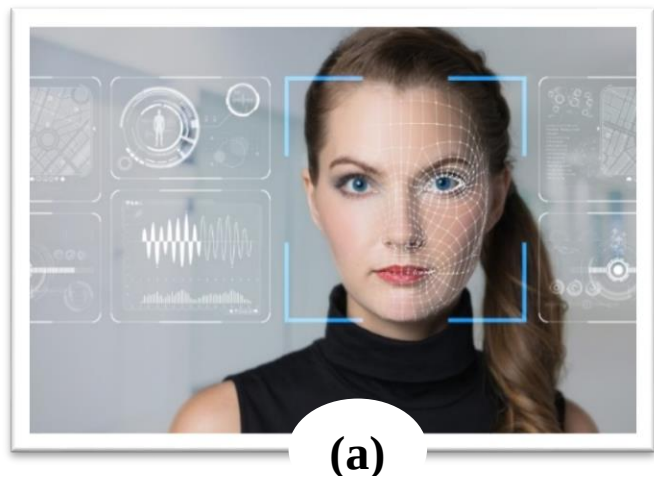




Figure 1.9: (a) reconnaissance faciale, (b) Différentes pointeuses biométriques par reconnaissance faciale.

- Reconnaissance vocale

Il s'agit d'une méthode d'authentification biométrique qui utilise des caractéristiques vocales pour identifier des personnes, en utilisant des systèmes de reconnaissance vocale automatisés

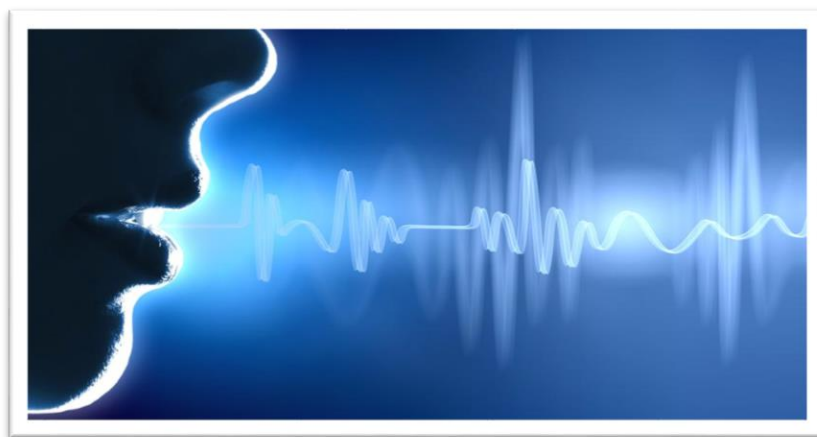


Figure 1.10: Reconnaissance vocale

- Reconnaissance par la signature

La collecte d'échantillons pour cette biométrie comprend sujet coopération et nécessite l'instrument d'écriture, il faut disposer d'une table à digitaliser et d'un stylo adapté, les mouvements de ce dernier sur la tablette sont enregistrés.



Figure 1.11: Reconnaissance par la signature

1.3.3. Système de pointage basé sur la technologie RFID

RFID, qui signifie Radio Frequency Identification, est une technologie d'identification et de communication à distance.

Elle utilise des ondes radio pour transférer les données d'une étiquette électronique, appelée tag ou carte RFID, reliée à un objet, via un lecteur ayant pour but d'identifier et de surveiller l'objet.

Monsieur Calais l'a confirmé dans son livre sur les RFID, disant que « La technologie RFID permet d'identifier, à plus ou moins longue distance, un objet muni d'une étiquette, capable d'émettre des données en utilisant les ondes radios » [9]



Figure 1.12: Symbole de la RFID

Par contre, Dat Son Nguyen a décrit cette dernière dans sa recherche sur cette technologie d'identification, comme étant une technologie qui permet la lecture des tags même

sans ligne de vue directe et peut traverser de fines couches de matériaux (peinture, neige, etc.) [10]

En effet, Monsieur Pierre Georget, définit la technologie RFID dans son ouvrage intitulé « L'identification par Radio Fréquence, principes et application » que c'est une technologie déjà largement utilisée pour reconnaître ou identifier à plus ou moins grande distance (du contact à plusieurs mètres) et dans un minimum de temps, un objet, un animal ou une personne porteuse d'une étiquette capable d'émettre des données en utilisant les ondes radio. [11]

Composants fondamentaux de la technologie RFID

Une combinaison de la technologie RFID et de la technologie informatique est appelée système RFID. Un système RFID se compose principalement :

- ✓ D'un ou plusieurs tags
- ✓ D'une antenne
- ✓ D'un lecteur/interrogateur
- ✓ Infrastructure de communication
- ✓ Logiciel d'application (base de données utilisateur/application/interface)

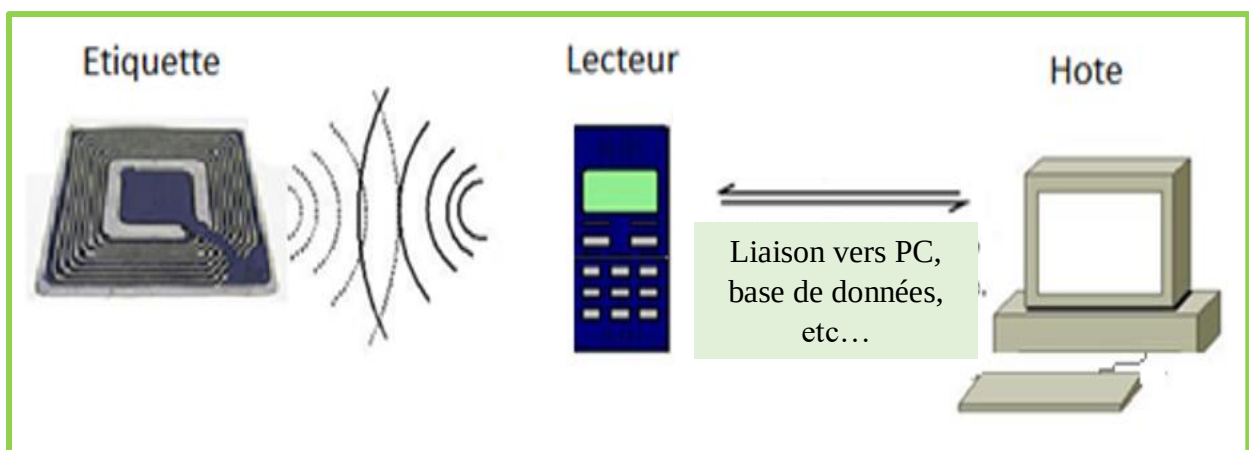


Figure 1.13: Composants d'un système RFID

1) Tag (étiquette électronique)

Une étiquette RFID est un petit appareil électronique, constituée d'une antenne et d'une puce électronique, Il peut être attaché à un objet.

Les informations sont collectées par puce et peuvent être transmises sans fil.

Le tag possède un code d'identification qui peut être transmis vers le lecteur. [12]

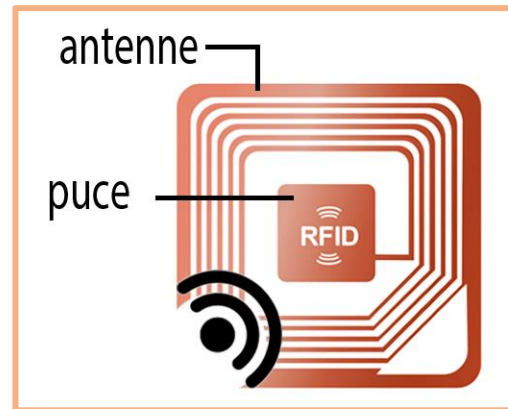


Figure 1.14 : Composant du Tag RFID

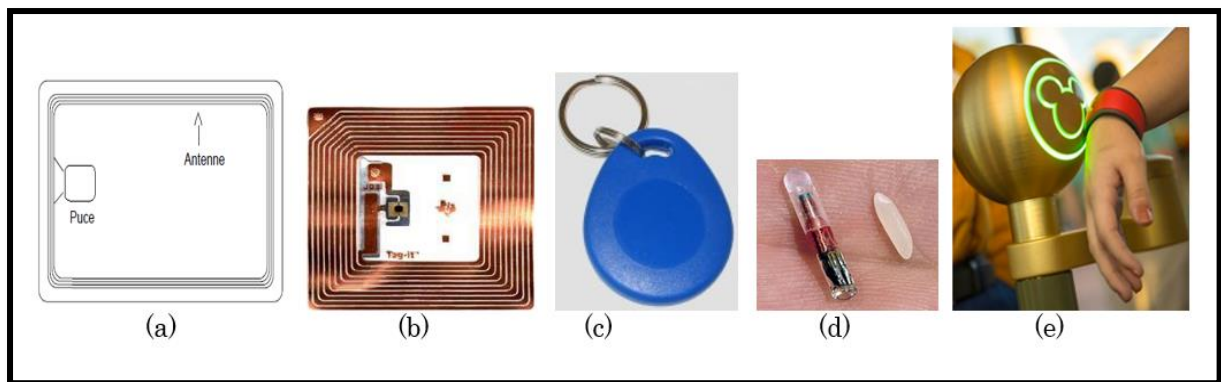


Figure 1.15: Différents supports de tags RFID, (a) schéma synoptique, (b) : tag nu, (c) porte-clés, (d) transplantable comparé à un grain de riz, (e) bracelet en face du lecteur

Il existe deux grandes catégories de Tags RFID ; Tag passif et actif comme le montre la figure

- **Tag RFID actif :** La radio identification active utilise des étiquettes actives, qui sont équipés d'une batterie qui peut être utilisée comme source d'alimentation. Une source d'énergie qui a la capacité de diffuser un signal vers le lecteur RFID. [3]
- **Tag RFID passif :** Contrairement aux tags actifs, les tags passifs ne disposent pas d'une source d'énergie. Ils tirent leur puissance à travers le signal électromagnétique du lecteur qui permet d'activer le tag et lui permet ainsi d'émettre les informations. [3]

2) Lecteur/Interrogateur

Ce sont des équipements actifs, portables ou fixes, constitués d'un circuit qui émet une énergie sous forme de champ magnétique ou d'onde radio, qui permet de communiquer avec les tags RFID.

Le lecteur envoie et reçoit des données RF vers et depuis l'étiquette via des antennes. Celle-ci capte le signal et renvoie ensuite une réponse au lecteur, qui contient un numéro de série ainsi qu'éventuellement d'autres informations. [12]

En effet, Il est le lien entre les tags RFID et le Serveur/ PC. Le coupleur peut envoyer des informations dans deux directions : il peut lire les informations d'une étiquette et les envoyer au Serveur/PC (mode lecture), ou il peut lire les informations du serveur et les envoyer à l'étiquette RFID (mode écriture).



Figure 1.16: Différents lecteurs RFID

Architecture et fonctionnement de la technologie RFID

La RFID est une méthode d'identification automatique, dans laquelle les données d'identification sont stockées dans des dispositifs électroniques, appelés étiquettes RFID (transpondeurs), et les lecteurs RFID (interrogeurs) récupèrent ces données.

Lorsque le lecteur est alimenté en tension, il génère un champ magnétique et lorsqu'un tag s'en approche par un effet électromagnétique cela génère un courant électrique et donc une différence de potentiel. [13]

Le tag RFID est équipée d'une puce électronique reliée à une antenne émettrice, la puce contient des informations sur le produit, c'est cette différence de potentiel qui permet à cette puce d'être alimentée en tension.

Lorsque des étudiants ou employés placent leurs étiquettes RFID qui contiennent un numéro d'identification unique sur le lecteur, un échange de donnée de courte distance se fait entre eux, en effet, l'antenne permet à la puce de stocker ces données et à les transmettre au lecteur RFID via des fréquence radio.

Les informations transmises sous forme d'ondes radio sont reçues par le lecteur RFID, celui-ci va convertir les ondes-radios en données qui pourront être lues par un système informatique comportant un logiciel.

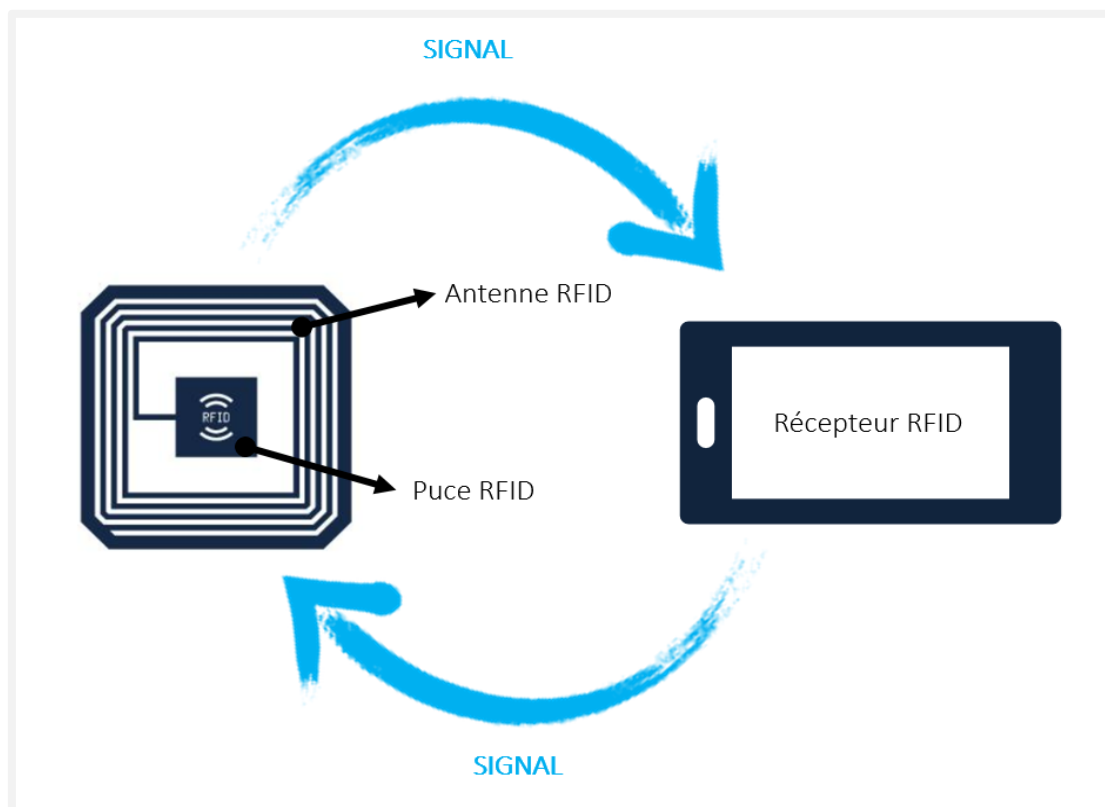


Figure 1.17: Communication entre Lecteur et Tag RFID

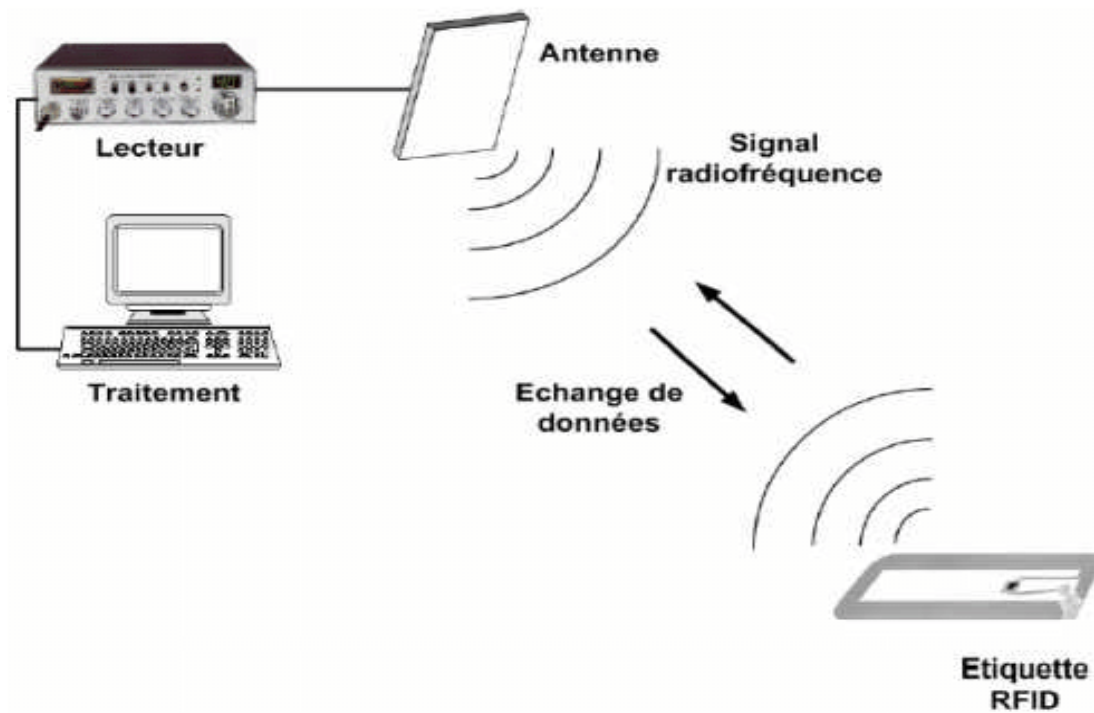


Figure 1.18: Architecture d'un système RFID

Fréquences de communication de la RFID

La fréquence est la caractéristique qui permet d'établir la communication entre la puce et l'antenne, tag et lecteur doivent être sur la même fréquence, la fréquence influe sur la distance émetteur-récepteur : plus la fréquence est élevée plus la distance entre le tag et le lecteur est importante.

Quatre types de fréquences sont principalement utilisés dans la technologie RFID : [12]

Fréquences	Descriptions
Basse fréquence (120-140 KHz)	Cette fréquence est utilisée pour déposer, retirer et contrôler le suivi des actifs, et surtout utilisée pour des applications à très faible portée.

Haute fréquence (13,56 MHz)	Cette fréquence est plus utilisée dans les applications de suivi des actifs, les cartes à puce sans contact, les badges d'identification.
Ultra-haute fréquence (869 MHz-928 MHz)	Cette fréquence est utilisée pour les applications spécifiques, longue portée. Elle offre une plage de lecture plus longue et est moins chère à fabriquer en vrac.
Super haute fréquence (les micros ondes) (2,4 GHz-2,5 GHz)	Le système à micro-ondes offre un taux de lecture plus élevé. Les étiquettes sont utilisées dans les applications de télépéage. Ces systèmes sont altérables par l'humidité et donc le climat (neige, pluie, humidité, brouillard,).

Tableau 1.2: Différentes fréquences de la RFID

Domaines d'applications de la technologie RFID

Les systèmes d'identifications par radio fréquence ont un large domaine d'application, ainsi, de nombreuses utilisations dans la vie quotidienne.

On peut citer ses différentes applications:

- Dans les outils de paiement
- Dans la gestion du transport (cartes de transports)
- Dans l'identification (les documents d'identité comme le passeport)
- Ils interviennent également dans les hôpitaux
- Suivi des animaux domestiques et les logistiques
- Emprunt et suivi des livres dans les bibliothèques et centres documentaires

- Dans la sécurité.
- La traçabilité
- Dans le contrôle d'accès
- Accélérer les inventaires en magasin, préparer les commandes plus rapidement
- Les clés sans contact dans les voitures...etc

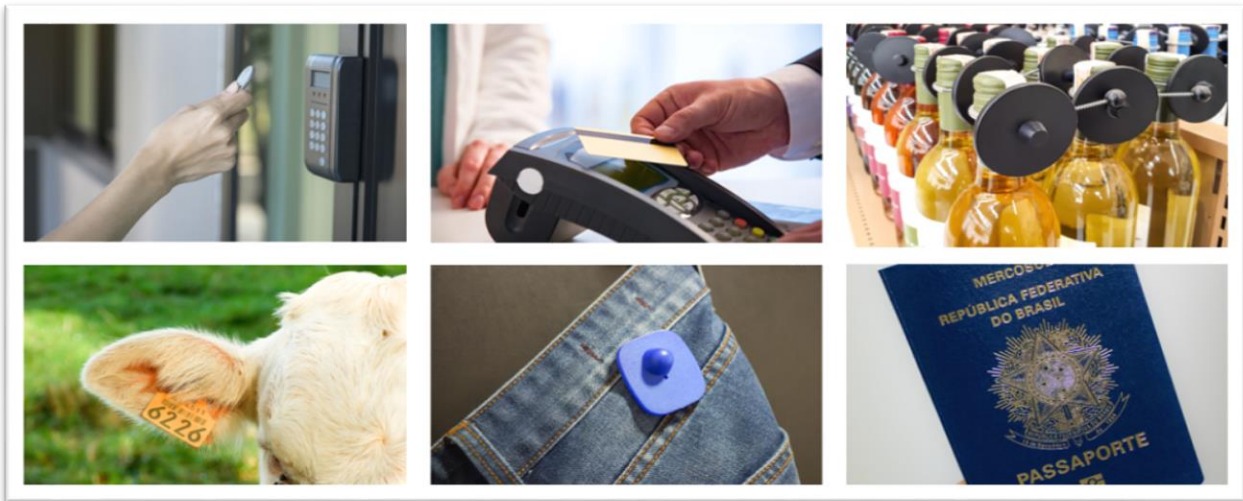


Figure 1.19: Exemples d'applications de la RFID

1.4. Conclusion

Dans ce chapitre, nous avons présenté différents systèmes de pointages comme les systèmes biométriques ou code à barres, ainsi que la technologie principale « RFID », les architectures et les fonctionnalités de ces technologies ont été discutées en profondeur.

À l'avenir, d'autres travaux d'ajout de fonctionnalités en termes de sécurité sur le système de pointage seront effectués. En outre, les fusions entre les identifiants biométriques tels que les traits du visage et les fonctionnalités des RFID seront également mises en œuvre.

Dans l'ensemble, cette nouvelle technologie, le RFID, devrait apporter des avantages aux étudiants de la génération actuelle dans les universités. La principale contribution d'un tel mouvement est d'utiliser complètement les capacités du smartphone au maximum et de tirer parti du phénomène actuel des smartphones chez les jeunes utilisateurs.

CHAPITRE II
SYSTÈME DE POINTAGE
BASE SUR LA RFID

2.1. Introduction

L'enregistrement de la présence des étudiants à l'aide de cartes RFID nécessite un système de pointage, appareil à concevoir et à construire. Plusieurs composants doivent être intégrés afin de concevoir cet appareil, capable de lire les cartes RFID.

Dans ce chapitre, nous allons expliquer l'objectif principale de notre projet, et étudier l'ensemble des outils de développement matériels et logiciels que nous allons avoir besoin pour la réalisation d'un tel système.

2.2. Description du système de pointage RFID proposé

Le système de gestion des présences des étudiants élimine le processus de marquage des présences au stylo et au papier et favorise l'enregistrement des présences en ligne en un seul clic. Avec ce module, les enseignants peuvent gérer et suivre les informations de présence des étudiants à partir d'appareils mobiles et de bureau.

Avec l'intégration du dispositif RFID, l'ensemble du processus devient automatisé car les étudiants peuvent scanner leur carte numérique devant ce dispositif, et la présence sera enregistrée dans le module de gestion des présences, qui aide au suivi des retardataires et des absences.

Le système rend la fréquentation quotidienne sans tracas, mais surtout pour libérer le temps afin que le seul objectif de l'enseignant soit d'enseigner et de transmettre des connaissances aux étudiants.

Dans ce projet, nous allons concevoir une pointeuse basée sur la technologie RFID, en utilisant une carte Arduino UNO pour le traitement des données.

Les étudiants sont émis d'étiquettes RFID contenant un numéro d'identification unique, et leur présence est marquée lorsqu'ils glissent leurs cartes devant le lecteur RFID.

Chaque fois que l'élève entre dans la classe, il doit scanner son badge devant le lecteur RFID, ce dernier lira le code d'identification et va le transférer au PC, qui se connectera par USB. Un programme sur PC récupérera l'identité de l'étudiant à partir de la base de données en utilisant le code d'identification qu'il a reçu et prendra les présences.

Tout d'abord, nous allons stocker les données UID des étiquettes RFID, on utilisera 04 numéros d'étiquettes pour 04 étudiants.

- ✓ Lorsqu'un étudiant arrive à l'heure et glissera sa carte devant le lecteur, cela enregistrera l'UID de l'utilisateur et l'heure d'arrivée sur une carte SD, affichant son statut « On

time » sur un écran LCD.

- ✓ Lorsqu'un autre étudiant arrive en retard et glissera sa carte devant le lecteur, le système enregistrera son heure d'arrivée en affichant son statut « Late » sur l'écran LCD.

Pour cela, le module RFID sera interfacé avec l'Arduino UNO pour lire les données sur la carte RFID. Egalement pour le module carte SD, en tant qu'enregistreur de données où les données seront enregistrées au format TXT.

On utilisera le module d'horloge RTC DS1307 en temps réel, pour afficher l'heure et la date sur l'écran LCD.

Une carte Secure Digital (SD) qui a une partie de stockage sera utilisé pour stocker les données, on stockera l'heure d'arrivée de chaque étudiant et l'identifiant de son étiquette.

Les principales caractéristiques du projet

- Contient un lecteur RFID RC522 qui lit les étiquettes.
- Un module d'horloge en temps réel RTC DS1307
- Lorsque le lecteur RFID lit une étiquette RFID, l'heure sera enregistrée dans la carte SD, ainsi que l'identifiant de l'étiquette
- Si un étudiant est à l'heure alors une LED verte va s'allumer, s'il arrive en retard alors une LED rouge s'allumera.
- Un buzzer qui émet un signal sonore lorsqu'une étiquette sera lue.
- Un écran LCD 16x2 qui affichera l'heure et la date, un message de bienvenu « Welcome to Control System », des messages comme « Status : On time » et « Status : Late » lorsqu'un étudiant arrive à l'heure ou en retard.
- Ensuite on affichera sur l'écran LCD un menu principal, en utilisant quatre boutons poussoirs pour naviguer dans ce menu : Un bouton poussoir de sélection, un bouton poussoir UP et un autre bouton DOWN pour aller de haut en bas, un bouton poussoir Go Back pour revenir dans le menu principal.

L'utilisation des boutons poussoirs est comme suite :

- Affichage du message « Press 1 for menu » sur l'écran LCD.
- Lorsqu'on clique sur le 1^{er} bouton poussoir de sélection, le menu s'affiche comme suite :

1.View All
2.Go Back

- Lorsqu'on clique sur le 1^{er} bouton poussoir (View All), les données de chaque étudiant seront affichées, comme suite :

[Name of student] [Status: Present/ Absent]
Arriving Time
1. Next 2.Go Back

- Lorsqu'on clique sur le 1^{er} bouton Next, alors les informations d'un autre étudiant seront affichées et ainsi de suite.
- Lorsqu'on clique sur le 2^{ème} bouton Go Back, on revient au menu principal.

2.3. Outils de développement matériels

2.3.1. Présentation de la carte Arduino

La carte Arduino est une plateforme d'objets interactifs à usage créatif utilisée pour réaliser des projets électroniques plus développée. Elle est composée d'un circuit physique programmable dit microcontrôleur, et d'un logiciel qui est utilisé pour créer et télécharger un code de l'ordinateur à la carte. [14]

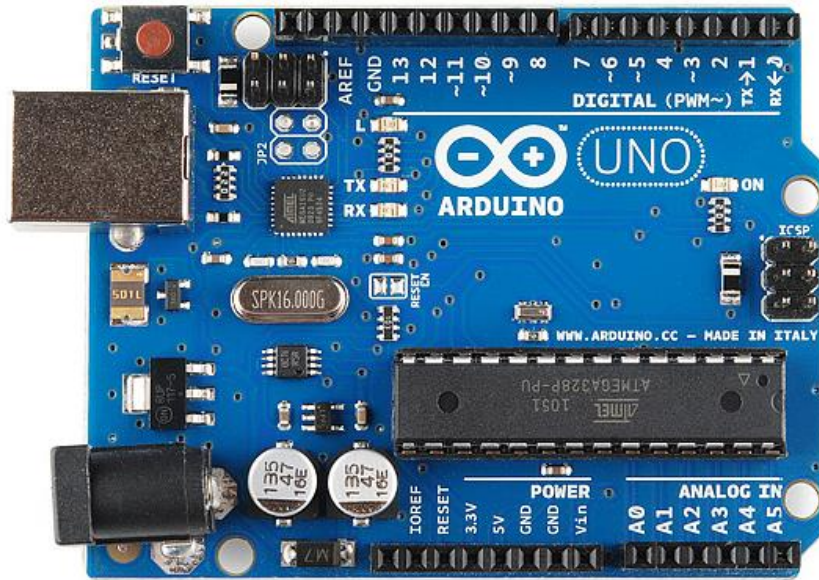


Figure 2.1 : Carte Arduino

Différents types de la carte Arduino

- **Carte Arduino Uno :** Cette carte Arduino dépend d'un microcontrôleur basé sur ATmega328P. Il comprend tout ce qui est nécessaire pour maintenir le microcontrôleur, il suffit de le connecter à un PC avec l'aide d'un câble USB et donner l'alimentation pour démarrer avec un adaptateur.
- **Carte Arduino Nano :** Il s'agit d'une petite carte basée sur les microcontrôleurs comme ATmega328P sinon ATmega628, la connexion de cette carte est la même que celle de la carte Arduino UNO. Comprend 8 broches analogiques, 14 broches numériques, broches E / S, broches d'alimentation et des broches RST (réinitialisation). [15]
- **Carte Arduino Mega :** Equipée d'un ATmega1280. Il comprend de nombreuses broches d'E / S numériques (à partir de là, 14 broches peuvent être utilisées comme PWM o / ps), 6 entrées analogiques, un bouton de réinitialisation, une prise d'alimentation, une connexion USB. [15]
- **Carte Arduino Mega2560 :** Equipée d'un ATmega2560, équipée aussi de la nouvelle puce USB ATmega8U2.

- [Carte Arduino Micro](#) : La carte Arduino Micro dépend principalement du microcontrôleur ATmega32U4, comprend 7 broches PWM, 12 broches d'entrée analogiques, un bouton RST, une petite connexion USB, un oscillateur à cristal de 16 MHz. [15]
- [Carte Arduino Due](#) : C'est une évolution de l'Arduino Mega2560, comprend 12 broches PWM o / p, 12 broches analogiques, 4 UARTs, un CLK avec 84 MHz, un USB OTG, 2 DAC, une prise d'alimentation, 2 TWI, un en-tête JTAG, un en-tête SPI, deux boutons pour réinitialiser et effacer. [15]
- [Carte LilyPad Arduino](#) : Une conception de type minimaliste pour permettre une application portable utilisant un ATmega168, cette carte se remarque par son vernis violet.
- [Carte Arduino dix mille](#) : Cette carte microcontrôleur dépend principalement de l'ATmega168 dans un boîtier format DIL28. Elle comprend 14 broches numériques d'E/S où 6 broches peuvent être utilisées comme des sorties PWM et 6 comme entrées analogiques, une connexion USB, un oscillateur à quartz-16 MHz, un en-tête ICSP, un bouton de réinitialisation et une prise d'alimentation. [15]
- [Carte Arduino Leonardo](#) : Basée sur l'ATmega32U4. La première carte de développement d'un Arduino est la carte Leonardo.
- [Carte Arduino Explora](#) : Elle a la forme d'une manette de jeu, c'est une carte « tout en un » qui comporte de nombreux capteurs, ainsi que quatre boutons poussoirs, un potentiomètre et un joystick analogique.
- [Micro Arduino Pro](#) : La carte Arduino Pro Micro est la même que la carte Arduino Mini à l'exception du microcontrôleur ATmega32U4. Elle comprend 12 broches numériques d'E / S, 5 broches PWM, des connexions séries de TX & RX et ADC 10 bits. [15]
- [Ethernet Arduino](#) : La carte dépend du microcontrôleur ATmega328. Elle comprend 5 broches analogiques, 14 broches numériques E / S, un bouton RST, une connexion RJ45, un oscillateur à cristal, une prise d'alimentation, un en-tête ICSP, etc. [15]

- Carte Arduino Extreme : Programmable via une connexion USB et utilisant un ATmega8
- Carte Arduino Yun : utilise le microcontrôleur Atmel ATmega32U4. Possède des fonctionnalités avancées permettant de réaliser des prototypes d'objets connectés. Elle dispose du WiFi intégré, d'un contrôleur Ethernet et un port RJ45 ainsi qu'un port USB host. [15]
- Carte Bluetooth Arduino : Ce Bluetooth dépend du microcontrôleur comme ATmega168. Ce type de carte comprend différents composants tels que 16 broches numériques, 6 broches analogiques, un oscillateur à cristal 16 MHz, un bouton de réinitialisation, des bornes à vis, en-tête ICSP. [15]

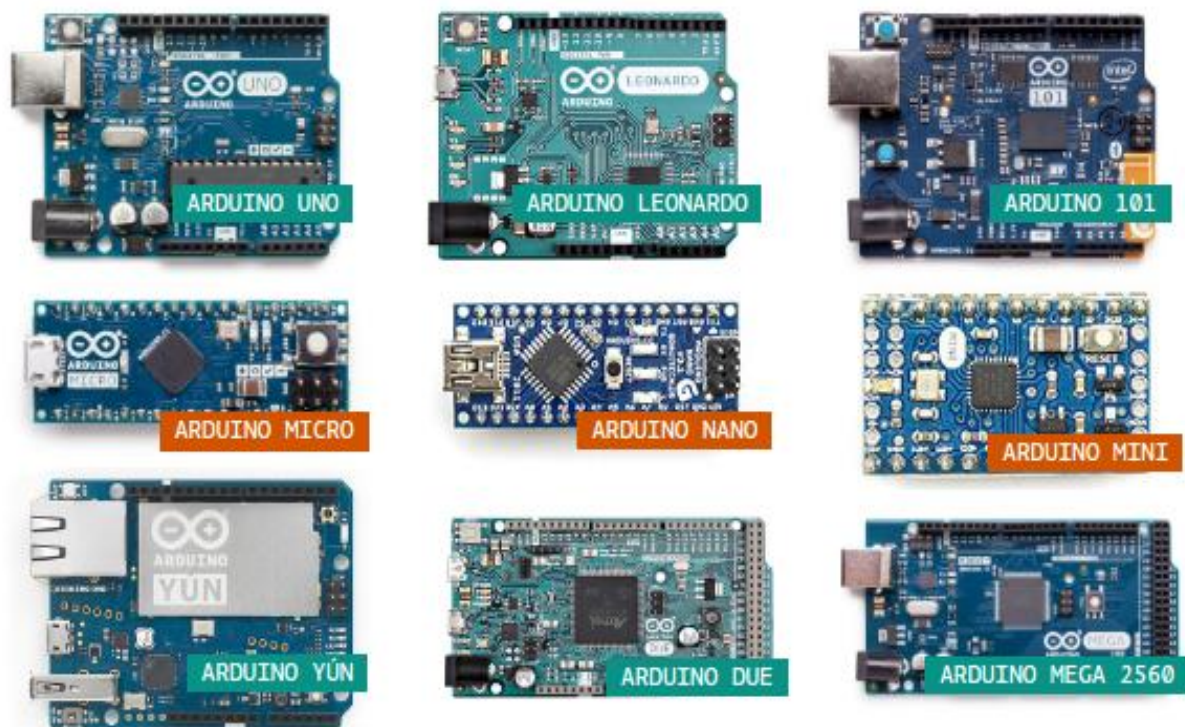


Figure 2.2 : Types de carte Arduino

2.3.2. Carte Arduino UNO

C'est une petite carte électronique équipée d'un microcontrôleur, ce dernier permet, à partir d'événements détectés par des capteurs, de programmer et de commander des actionneurs ; la carte Arduino est donc une interface programmable.

Le modèle UNO de la société ARDUINO est une carte électronique dont le cœur est un microcontrôleur ATMEL de référence ATmega328 cadencé à 16 MHz. [14]

Caractéristiques techniques de la carte Arduino UNO

La carte Arduino UNO est basée sur l'ATmega328, ses principales caractéristiques sont présentées sur le tableau ci-dessous :

Microcontrôleur	ATmega328
Fréquence horloge	16 Mhz
Tension de fonctionnement	5V
Tension d'alimentation (recommandée)	7-12V
Tension d'alimentation (limite)	6-20V
Broches E/S numériques :	14 (dont 6 peuvent être utilisées en sorties PWM)
Broches d'entrées analogiques	6 (qui peuvent également être utilisées en broches d'entrées/sorties numériques)
Intensité maxi disponible par broche E/S (5V)	40 mA (200 mA cumulés pour toutes les broches)

Intensité maxi disponible pour la sortie (3.3V)	50 mA
Intensité maxi disponible pour la sortie 5V	500 mA max sur port USB
Mémoire Programme Flash	32Ko (ATmega328) dont 0.5 Ko sont utilisés par le bootloader (Chargeur d'amorçage)
Mémoire SPRAM (mémoire variable)	2Ko (ATmega328)
Mémoire EEPROM (mémoire non variable)	1Ko (ATmega328)
Interface	USB (permet d'alimenté la carte et de transférer les programmes vers la carte)
Alimentation externe	Jack
Dimensions	6.86 cm × 5.3cm

Tableau 2.1 : Caractéristiques techniques de la carte Arduino UNO [16]

Les principaux composants d'une carte Arduino

Dans la figure ci-dessous, une carte Arduino étiquetée. Voyons ce que veut dire chaque partie : [17]

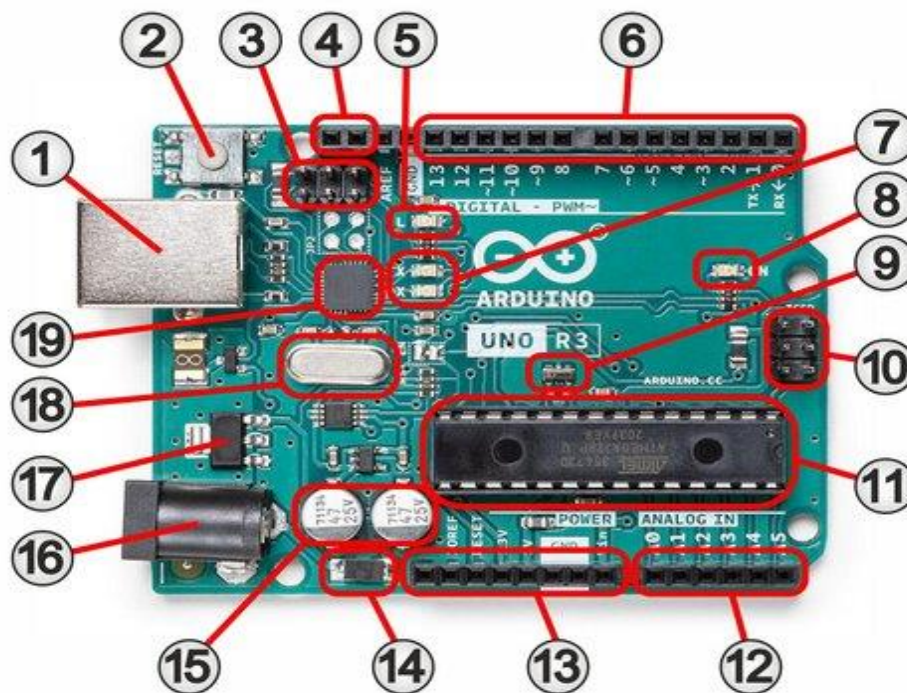


Figure 2.3: Composants de la carte Arduino UNO

1. Port USB

Branchez un câble USB sur cette prise pour transférer le programme de commande personnalisé à l'Arduino.

2. Bouton de réinitialisation

Ce bouton permet une réinitialisation manuelle de l'Arduino en cas d'arrêt automatique de celui-ci suite à un dysfonctionnement.

3. Interface ICSP (interface USB)

L'interface ICSP (In Circuit Serial Programming) permet de programmer un circuit logique directement dans le système de codage.

4. Bus I2C

Le bus I2C permet l'envoi des données de l'appareil maître aux esclaves raccordés via les lignes SCL (Serial Clock) et SDA (Serial Data).

5. LED intégrées

La LED « L » est connectée de manière interne à la broche 13 et est utilisée pour effectuer des tests.

6. Entrées/sorties numériques

Ces broches d'E/S peuvent être configurées comme des entrées ou des sorties numériques. Six d'entre elles fonctionnent comme des sorties PWM à modulation de largeur d'impulsion si nécessaire.

7. LED témoins

Les LED « RX » et « TX » indiquent visuellement le transfert des données du PC vers la carte Arduino UNO.

8. LED de tension de fonctionnement

La LED « ON » indique l'alimentation en tension de la carte microcontrôleur. Elle s'allume lorsque la carte est alimentée en courant.

9. Oscillateur à quartz (contrôleur)

Le quartz garantit que l'oscillateur dans le contrôleur oscille de manière stable et avec une fréquence constante.

10. Interface ICSP (contrôleur)

Il est possible de programmer le contrôleur via cette interface. Mais cela a déjà été fait à la sortie d'usine et n'est donc généralement pas nécessaire.

11. Microcontrôleur

Un microcontrôleur est une puce à semi-conducteurs qui combine le processeur, les périphériques et la mémoire. C'est lui qui va recevoir le programme que vous aurez créé et qui va le stocker dans sa mémoire puis l'exécuter.

12. Entrées analogiques

Ces six entrées doivent être utilisées si des tensions analogiques, provenant de capteurs par exemple, sont disponibles comme valeurs d'entrée.

13. Broches d'alimentation

Ces broches peuvent être utilisées pour alimenter en tension la carte du microcontrôleur ou pour prélever des tensions (3,3 V ou 5 V).

14. Diode redresseuse

L'utilisation d'une diode redresseuse permet d'alimenter le système avec un courant alternatif ou continu.

15. Condensateurs de charge

Les condensateurs de charge égalisent la tension d'alimentation. L'un des condensateurs est placé avant le stabilisateur de tension de 5 V et l'autre, après.

16. Port d'alimentation

Pour que la carte microcontrôleur puisse également fonctionner sans connexion USB après la programmation, elle doit être alimentée par une source externe.

17. Stabilisateur de tension

Le stabilisateur de tension génère un courant stable de 5 V à partir de la tension d'alimentation, qui doit être comprise entre 7 et 12 V (AC ou DC).

18. Oscillateur à quartz (contrôleur USB)

Le quartz garantit que l'oscillateur dans le contrôleur USB oscille de manière stable et avec une fréquence constante.

19. Interface USB

L'interface USB convertit les signaux reçus dans un format utilisable pour le contrôleur.

Le microcontrôleur ATMEL ATmega328

Le microcontrôleur ATmega328 est un microcontrôleur 8bits de la famille AVR dont la programmation peut être réalisée en langage C. Les différentes broches de ce microcontrôleur sont représentées dans la figure ci-dessous.

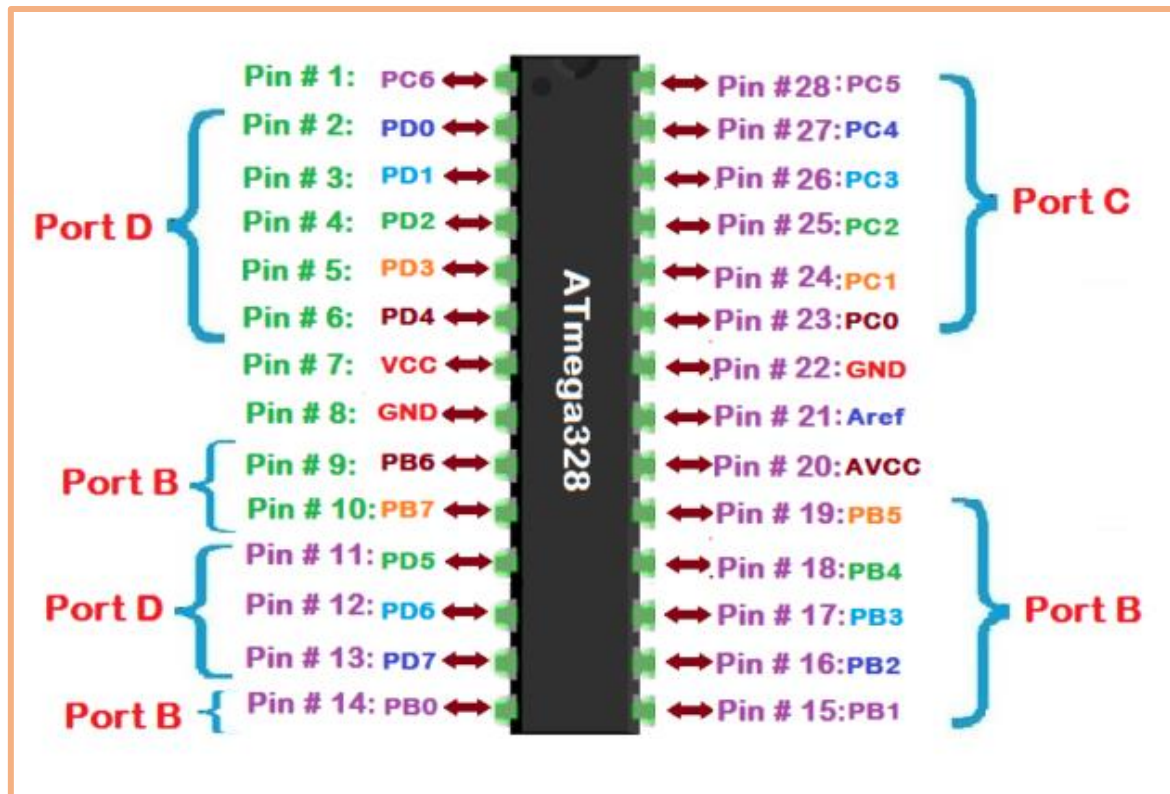


Figure 2.4 : Broches du microcontrôleur ATmega328

Ses principales caractéristiques sont :

FLASH

Mémoire 32Ko

SRAM	Données 2Ko
EEPROM	Données 1Ko
Broches numériques d'E/S	3 ports Port B, Port C, Port D
Timer/Compteurs	Timer0 et Timer2 (comptage 8 bits), Timer1 (comptage 16bits). Chaque Timer peut être utilisé pour générer deux signaux PWM

Plusieurs broches multifonctions	certaines broches peuvent avoir plusieurs fonctions différentes choisies par programmation.
PWM	6 broches.
Convertisseur analogique vers numérique	6 entrées multiplexées ADC0(PC0) à ADC5(PC5).
Gestion bus I2C	le bus est exploité via les broches SDA(PC5)/SCL(PC4).
Port série	émission/réception série via les broches TXD(PD1)/RXD(PD0)
Comparateur Analogique	broches AIN0(PD6) et AIN1 (PD7) peuvent déclencher l'interruption Watch dog Timer programmable.
Gestion d'interruptions	- Interruptions liées aux entrées INT0 (PD2) et INT1 (PD3) - Interruptions sur changement d'état des broches PCINT0 à PCINT23 - Interruptions liées aux Timers 0, 1 et 2 (plusieurs causes configurables) - Interruption liée au comparateur analogique - Interruption de fin de conversion ADC - Interruptions du port série USART Interruption du bus TWI (I2C).

Tableau 2.2 : Caractéristiques techniques de l'ATMega328 [18]

Alimentation de la carte Arduino UNO

Pour fonctionner, la carte a besoin d'une alimentation. Le microcontrôleur fonctionnant sous 5V, la carte peut être alimentée en 5V par le port USB (en 2) ou bien par une alimentation externe (en 3) qui est comprise entre 7V et 12V.

Cette tension doit être continue et peut par exemple être fournie par une pile 9V. Un régulateur se charge ensuite de réduire la tension à 5V pour le bon fonctionnement de la carte.

2.3.3. Module RFID

RFID veut dire radio-frequency identification (identification par radiofréquence), c'est un module qui utilise des champs électromagnétiques pour transférer des données sur de courtes distances. La RFID est utile pour identifier les personnes, transactions, ou même pour ouvrir des portes. Et seule la personne ayant les bonnes informations sur sa carte est autorisée d'accès. Un système RFID a besoin d'étiquettes et d'un lecteur :

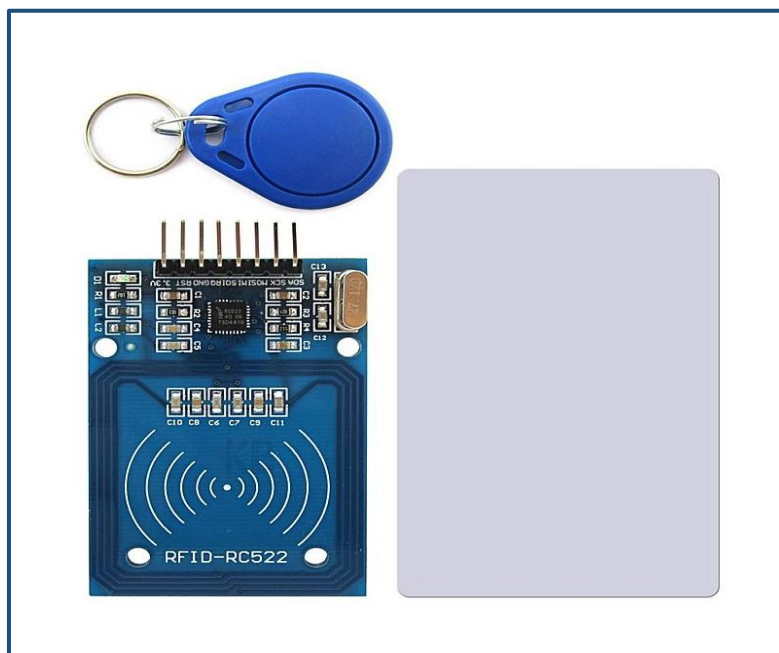


Figure 2.5: Système RFID

Etiquettes RFID

Ces étiquettes sont attachées à l'objet à identifier et permettent un contrôle d'accès fiable et sécuritaire. Chaque étiquette a sa propre identification unique (UID).

Dans l'exemple ci-dessous nous avons un porte-clés et une carte électromagnétique (badge).

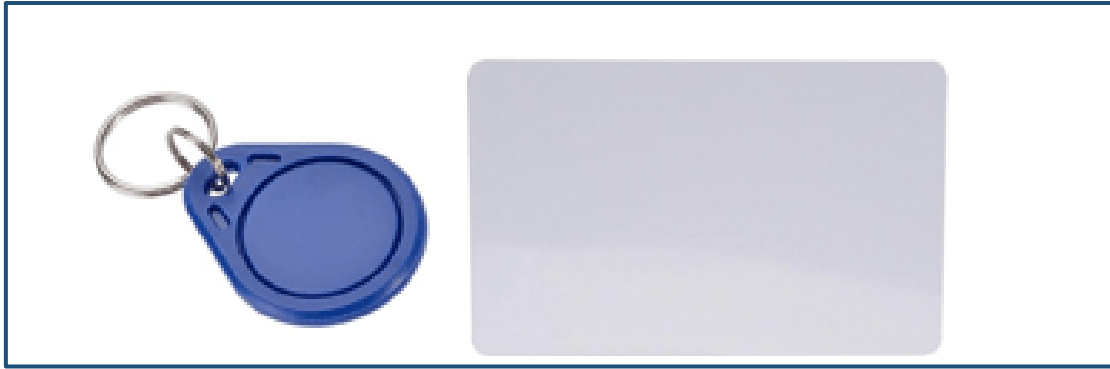


Figure 2.6: Etiquettes RFID

- **Porte clé :** Le porte-clés RFID est un produit simple et pratique qui s'adapte à toutes les situations.
- **Badge RFID :** Le badge RFID sont munis d'une antenne et d'une puce aussi, pour permettre la transmission de données avec un lecteur. Il est réalisé à partir d'un PVC ultra blanc offrant une résistance parfaite lors de manipulations répétées.

Le module lecteur RFID RC522

C'est est un émetteur-récepteur radio bidirectionnel qui envoie un signal à l'étiquette RFID et lit sa réponse.

Conçu pour créer un champ électromagnétique de 13,56 MHz qu'il utilise pour communiquer avec les Tags.

Le lecteur est basé sur le circuit intégré MFRC522, et communique avec un microcontrôleur via une interface périphérique série (SPI) avec un débit de données maximal de 10 Mbps. Il prend également en charge la communication via les protocoles I2C et UART. [18]



Figure 2.7: Lecteur RFID RC522

Caractéristiques techniques du module RFID

Fréquence de fonctionnement	13,56 MHz
Tension d'alimentation	3.3V
Courant de travail.	13-26mA/DC 3.3V
Courant de repos	10-13mA/DC 3.3V
Courant de veille	<80uA
Courant de crête	<30mA
Portée de lecture	Environ 3 cm avec la carte et le porte-clés fournis
Interface de communication	SPI
Taux de transfert de données maximum	10 Mbit / s
Température de fonctionnement	- 20 - 80 degrés Celsius
Température de stockage	-40 - 85 degrés Celsius
Humidité relative	5% à 95%
Distance lecture	≥ 50mm
Taille du module	40mm × 60mm

Tableau 2.3 : Caractéristiques techniques du module RFID [19]

Le Brochage

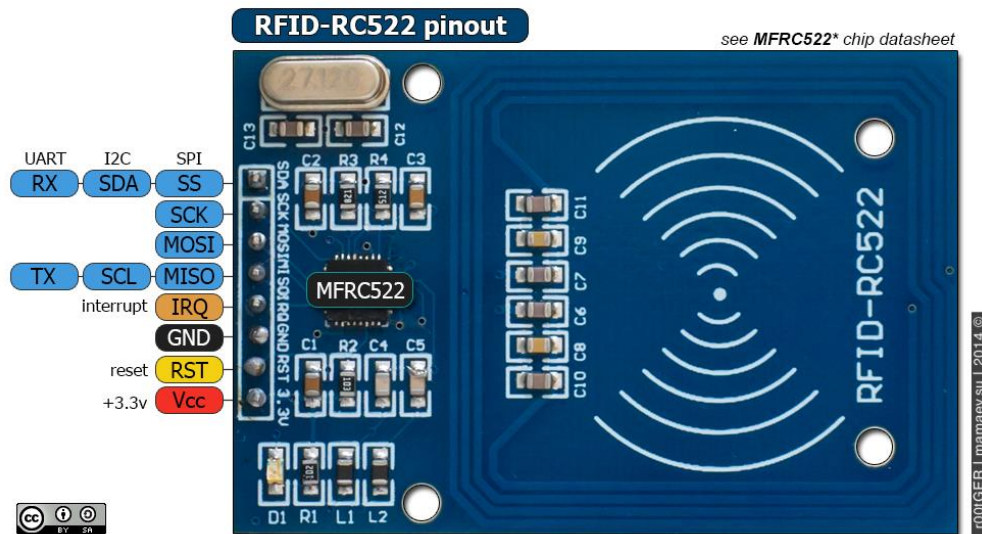


Figure 2.8: Broches du module RFID RC522

- **VCC** : utilisé pour alimenter le module, généralement on utilise 3,3 V
- **RST**: broche de réinitialisation - utilisée pour réinitialiser ou éteindre le module
- **Ground**: connectée à la masse du système.
- **IRQ**: broche d'interruption - utilisée pour interrompre le module lorsqu'un périphérique entre dans la portée.
- **MISO/SCL/Tx**: la broche MISO lorsqu'elle est utilisée pour la communication SPI, agit comme SCL pour I2C et Tx pour UART.
- **MOSI**: sortie maître entrée esclave pour la communication SPI
- **SCK**: broche d'horloge série - utilisée pour fournir une source d'horloge
- **SS/SDA/Rx**: Agit comme entrée série (SS) pour la communication SPI, SDA pour I2C et Rx pour l'UART

Principe de fonctionnement du module RFID

La RFID fonctionne sur le principe des ondes radiofréquences. Ce système se compose d'un microcontrôleur et d'un lecteur RFID.

Ce lecteur RFID est constitué d'une bobine, lorsqu'une tension est donnée à cette bobine elle produit un champ électromagnétique. Lorsque ce champ électromagnétique est lié à

l'étiquette RFID, il produit un courant. Ce courant transfère les données de l'étiquette RFID au lecteur RFID.

2.3.4. Afficheur LCD

Les afficheurs à cristaux liquides, autrement appelés afficheurs LCD (Liquid Crystal Display), sont des modules compacts intelligents et nécessitent peu de composants externes pour un bon fonctionnement. Ils consomment relativement peu (de 1 à 5 mA).

L'afficheur LCD a 16 colonnes et 2 rangées, alors il y aura 32 caractères au total et chaque caractère sera composé de 5×8 Pixel Dots.



Figure 2.9 : Afficheur LCD

Caractéristiques techniques d'un afficheur LCD:

Format d'affichage LCD	16x2 points
Dimension de contour	85 (L) x 30 (H) x 12 (T) mm
Zone visuelle	64,5 × 16,0 mm
Zone active	55,45 (L) x 10,75 (H) mm
Taille des caractères	2,95 x 5,15 mm

Taille diagonale	2,6"
Taille de point (pixel)	0,55 x 0,60 mm
IC Package	COB LCD
IC ou équivalent	SPLC780C/HS44780/KS0066 ou équivalent
Interface	6800 parallèle 8 bits/4 bits
Interface	Parallèle
Type d'affichage	STN/FSTN
Direction de visualisation	6 : 00
Couleur du rétroéclairage	LED bleu/jaune vert/gris
Apparence	Blanc sur bleu/noir sur jaune vert/blanc sur gris
Alimentation (typique)	3,3 V/5 V
Courant de rétroéclairage (typique)	15 mA
Température de fonctionnement	-20°C~70°C
Température de stockage	-30°C~80°C

Tableau 2.4 : Caractéristiques techniques de l’afficheur LCD [19]

Le brochage

Un écran LCD est "normalisé" avec 14 broche (ou 16 si l'écran est rétro éclairé) pour les "petits formats »

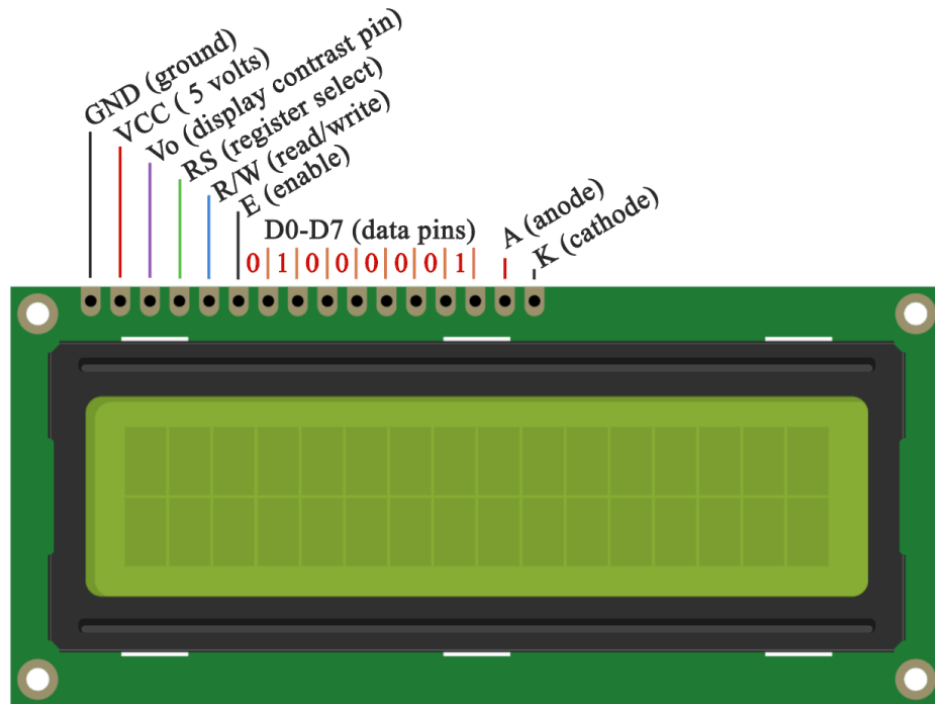


Figure 2.10 : Broches de l'afficheur LCD

- **GND** : Masse
- **VCC** : Alimentation positive (+5V).
- **Vo** : Cette tension permet, en la faisant varier entre 0 et +5V, le réglage du contraste de l'afficheur.
- **RS** : Sélection du registre (Registre Select). Grâce à cette broche, l'afficheur est capable de faire la différence entre une commande et une donnée. Un niveau bas indique une commande et un niveau haut indique une donnée.
- **RW** : Lecture ou écriture (Read/Write)
- **E** : Entrée de validation (Enable) active sur front descendant. Le niveau haut doit être maintenue pendant au moins 450 ns à l'état haut.
- **D0-D7** : Bus de données bidirectionnel 3 états (haute impédance lorsque E=0)
- **A** : Anode rétro éclairage (+5V)
- **K** : Cathode rétro éclairage (masse)

2.3.5. Horloge RTC

RTC (Real Time Clock signifie horloge en temps réel). Les modules RTC sont simplement des systèmes de mémorisation de l'heure et de la date.

Ils sont dotés d'une configuration de batterie qui, en l'absence d'alimentation externe, maintient le module en marche, cela maintient l'heure et la date à jour. Ainsi, nous pouvons avoir une heure et une date précises à partir du module RTC quand nous le voulons.

L'horloge temps réel équipée d'une pile pour rester à l'heure même lorsqu'un projet est hors tension ou pendant la reprogrammation d'un microcontrôleur.

Le circuit intégré d'horloge en temps réel DS1307 est un dispositif d'horloge/calendrier basse consommation à 8 broches. Et fournit des données qualifiées de secondes, minutes, heures, jour, date, mois et année.

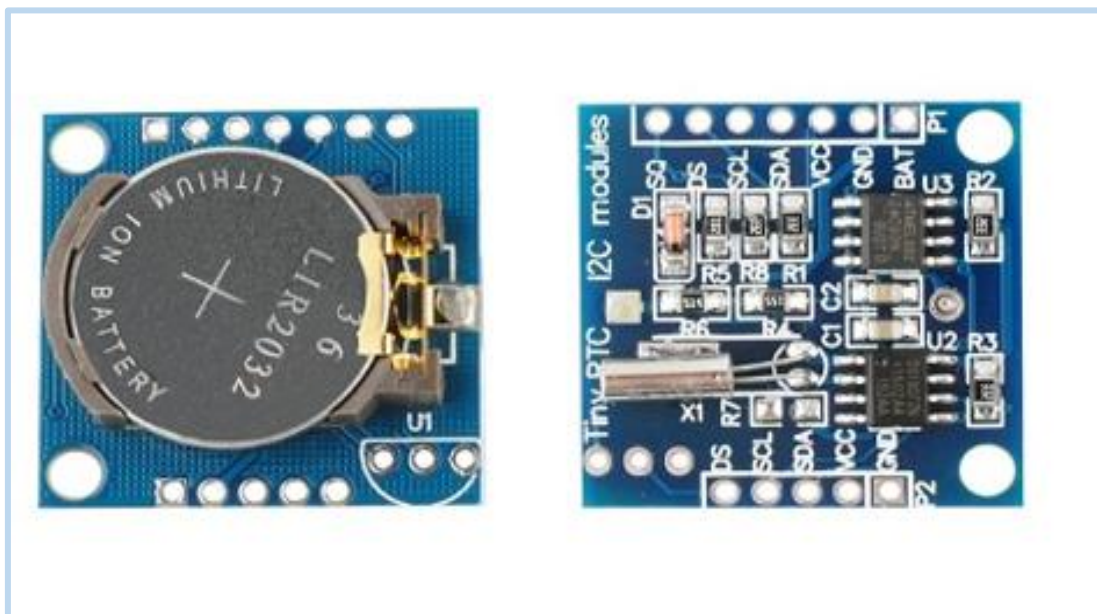


Figure 2.11: Horloge RTC DS1307

Caractéristiques techniques de l'horloge RTC

Alimentation	4.5V à 5V
Interfacer de niveau pour la carte de circuit de conversion	5 V ou 3,3 V
Consommation	1,5 mA (0,5 mA en mode batterie de secours)
Interface de communication	Une interface SPI standard, ou I2C
Dimension	32 x 23 x 11 mm
Taille	4,1 x 2,4 cm
Poids	5g

Tableau 2.5 : Caractéristiques techniques de l'horloge RTC DS1307 [19]

Le Brochage

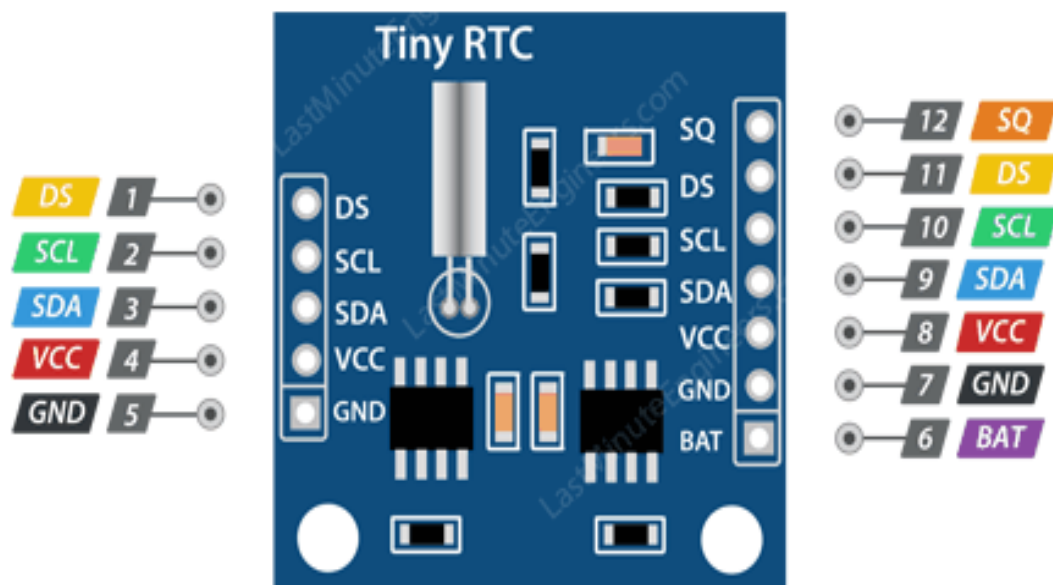


Figure 2.12: Broches du module RTC DS1307

- **SCL** : L'entrée d'horloge pour l'interface I2C et est utilisée pour synchroniser le mouvement des données sur l'interface série.
- **SDA** : L'entrée/sortie de données pour l'interface série I2C.
- **VCC** : Alimente le module, et peut être entre 3,3 V et 5,5 V.
- **GND** : Broche de terre.
- **SQ** : Produit l'une des quatre fréquences d'onde carrée 1Hz, 4kHz, 8kHz ou 32kHz et peut être activée par programmation.
- **DS** : Censée produire des lectures de température
- **BAT** : Entrée d'alimentation de secours pour toute pile au lithium 3V standard ou autre source d'énergie afin de maintenir un chronométrage précis lorsque l'alimentation principale de l'appareil est interrompue.

2.3.6. Lecteur Carte SD

Le module de carte micro-SD est une solution simple pour transférer des données vers et depuis une carte SD standard. Le brochage est directement compatible avec la carte Arduino, mais peut également être utilisé avec d'autres microcontrôleurs.

Ce module a une interface SPI qui est compatible avec n'importe quelle carte SD et il utilise une alimentation 5V ou 3.3V qui est compatible avec la carte Arduino UNO et Mega..



Figure 2.13: Module Carte SD

Caractéristiques techniques du module carte SD

Tension de fonctionnement	3.3 V - 5 V
Exigence de courant	0,2-200 mA
Régulateur de tension intégré	3,3 V
Micro SD	compatible
Prend en charge	le système de fichiers FAT
Interface	SPI

Tableau 2.6: Caractéristiques techniques du module carte SD [19]

Le Brochage

Le module de carte micro SD est assez simple à connecter. Il a six broches :

- **VCC** : Alimente le module et doit être connectée à la broche 5V.
- **GND** : Connectée à la masse.
- **MISO (Master In Slave Out)** est la sortie SPI du module de carte Micro SD.
- **MOSI (Master Out Slave In)** est l'entrée SPI du module de carte Micro SD.
- **SCK (Serial Clock)** accepte les impulsions d'horloge qui synchronisent la transmission des données générées par Arduino.

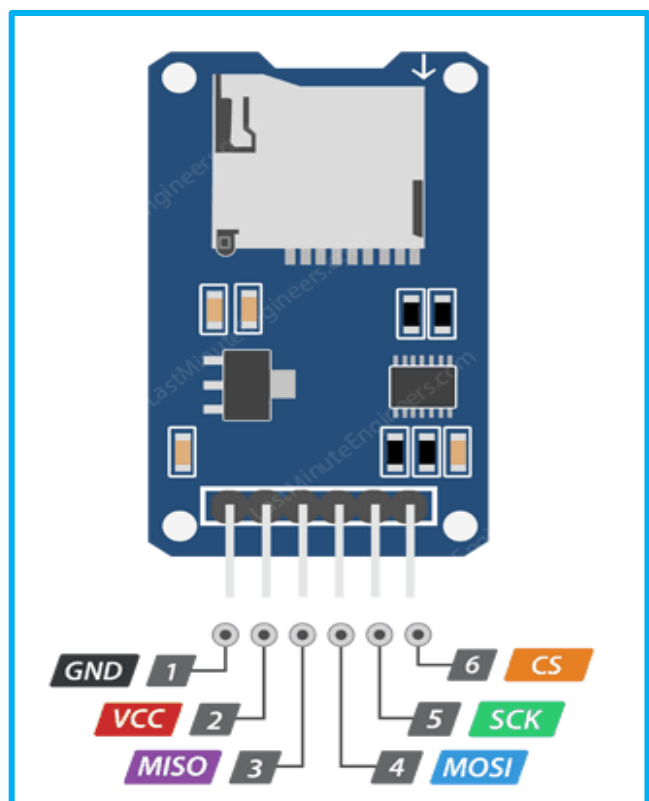


Figure 2.14: Broches du module carte SD

- **SS (Slave Select)** est utilisée pour activer et désactiver des périphériques spécifiques sur le bus SPI.

2.3.7. Buzzer

Le principe de fonctionnement principal est basé sur la théorie selon laquelle, chaque fois qu'un potentiel électrique est appliqué à travers un matériau piézoélectrique, une variation de pression est générée.

Dans le monde de l'Arduino, un buzzer est un appareil produisant du son.



Figure 2.15: Buzzer

Caractéristique technique d'un buzzer

Tension nominale 6 V CC

Tension de fonctionnement	4-8V CC
Courant nominal	<30mA
Type de son	Bip continu

Fréquence de résonance

~2300 Hz.

Tableau 2.7: Caractéristiques techniques du buzzer [19]

2.4. Outils de développement logiciel

2.4.1. Logiciel de programmation Arduino IDE

Le logiciel de programmation des modules Arduino est une application Java, libre et multiplateformes, servant d'éditeur de code et de compilateur, et qui peut transférer le FILMWARE et le programme au travers de la liaison série (Bluetooth ou USB selon le module). [14]

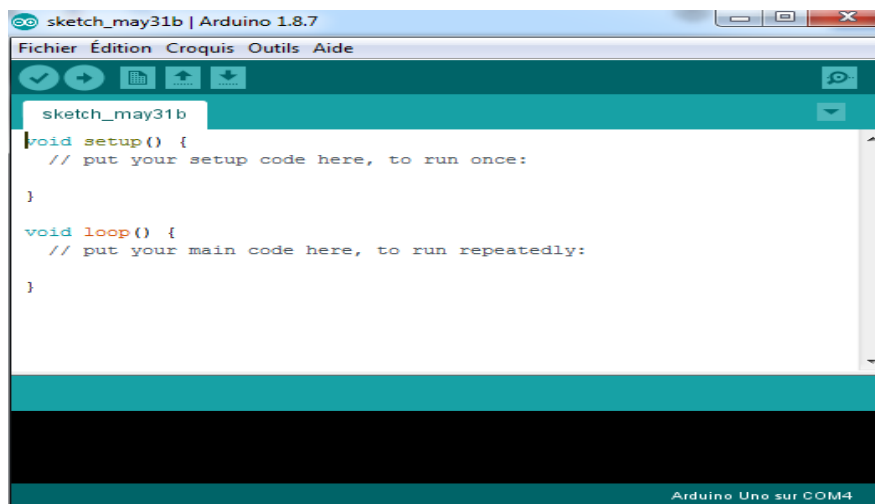
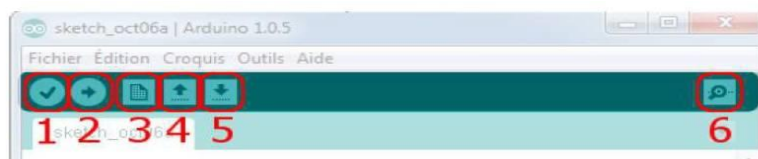


Figure 2.16 : Logiciel Arduino



1		Verify : Ce bouton permet de vérifier le programme, il actionne un module qui cherche les erreurs dans votre programme
2		Upload : Charge (téléverse) le programme dans la carte Arduino
3		New : pour créer un nouveau programme (sketch)
4		Open : pour ouvrir un programme existant
5		Save : pour sauvegarder le programme
6		Verify : pour visualiser les données transmises par le port série

Figure 2.17: Constitutions de l'Arduino IDE

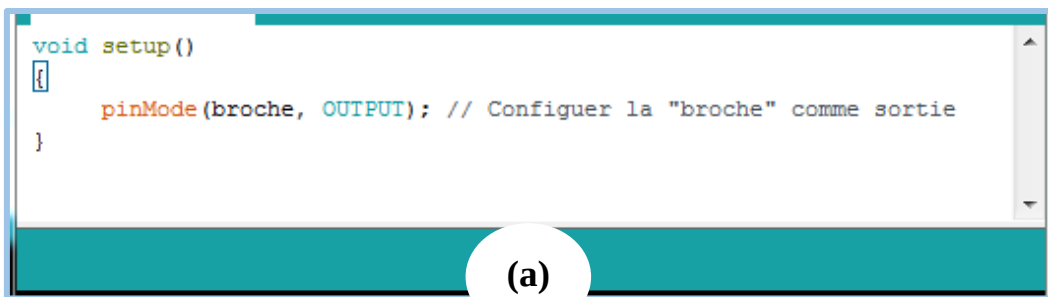
Le langage de programmation utilisé est le C++, et lié à la bibliothèque de développement Arduino, permettant l'utilisation de la carte et de ses E/S.

Syntaxe du langage (Arduino)

La structure de base du langage de programmation Arduino comprend au moins deux parties, qui sont impérativement requises pour que le programme fonctionne, ces deux parties ou fonctions contiennent des blocs d'instructions et sont : Setup() et Loop().

Setup () est l'initialisation du programme et Loop () est l'exécution du programme.

- **La fonction setup ()** doit suivre la déclaration des variables au tout début du programme. Il s'agit de la première fonction à exécuter dans le programme. Elle est exécutée une seule fois et sert à établir le mode d'une broche (pinMode) ou à initialiser la communication série.

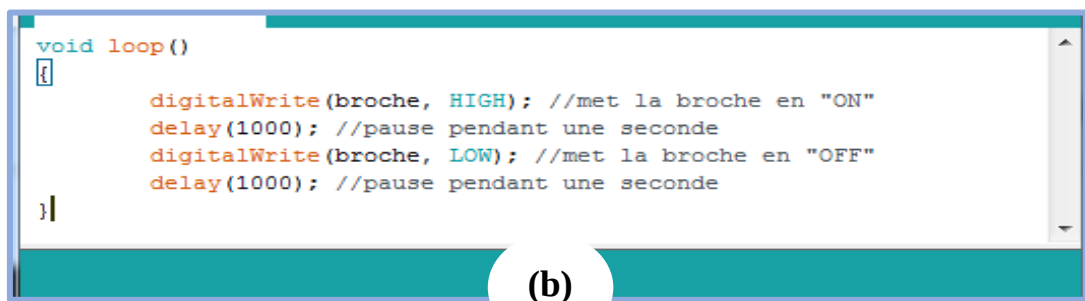


```
void setup()
{
    pinMode(broche, OUTPUT); // Configurer la "broche" comme sortie
}
```

(a)

- **La fonction loop ()** suit immédiatement et comprend le code à exécuter en continue – mettant les capteurs en entrée et les actionneurs en sortie.

Cette fonction est le noyau de tout programme Arduino et réalise l'essentiel du travail.



```
void loop()
{
    digitalWrite(broche, HIGH); //met la broche en "ON"
    delay(1000); //pause pendant une seconde
    digitalWrite(broche, LOW); //met la broche en "OFF"
    delay(1000); //pause pendant une seconde
}
```

(b)

Figure 2.18: (a) Fonction Setup(), (b) Fonction Loop()

Le logiciel comprend aussi un moniteur série qui permet d'afficher des messages textes émis par la carte Arduino, et d'envoyer des caractères vers notre carte.

La carte lit puis effectue les instructions les unes après les autres, dans l'ordre défini par les lignes de code. La structure d'écriture d'un programme sous Arduino est de la forme suivante:

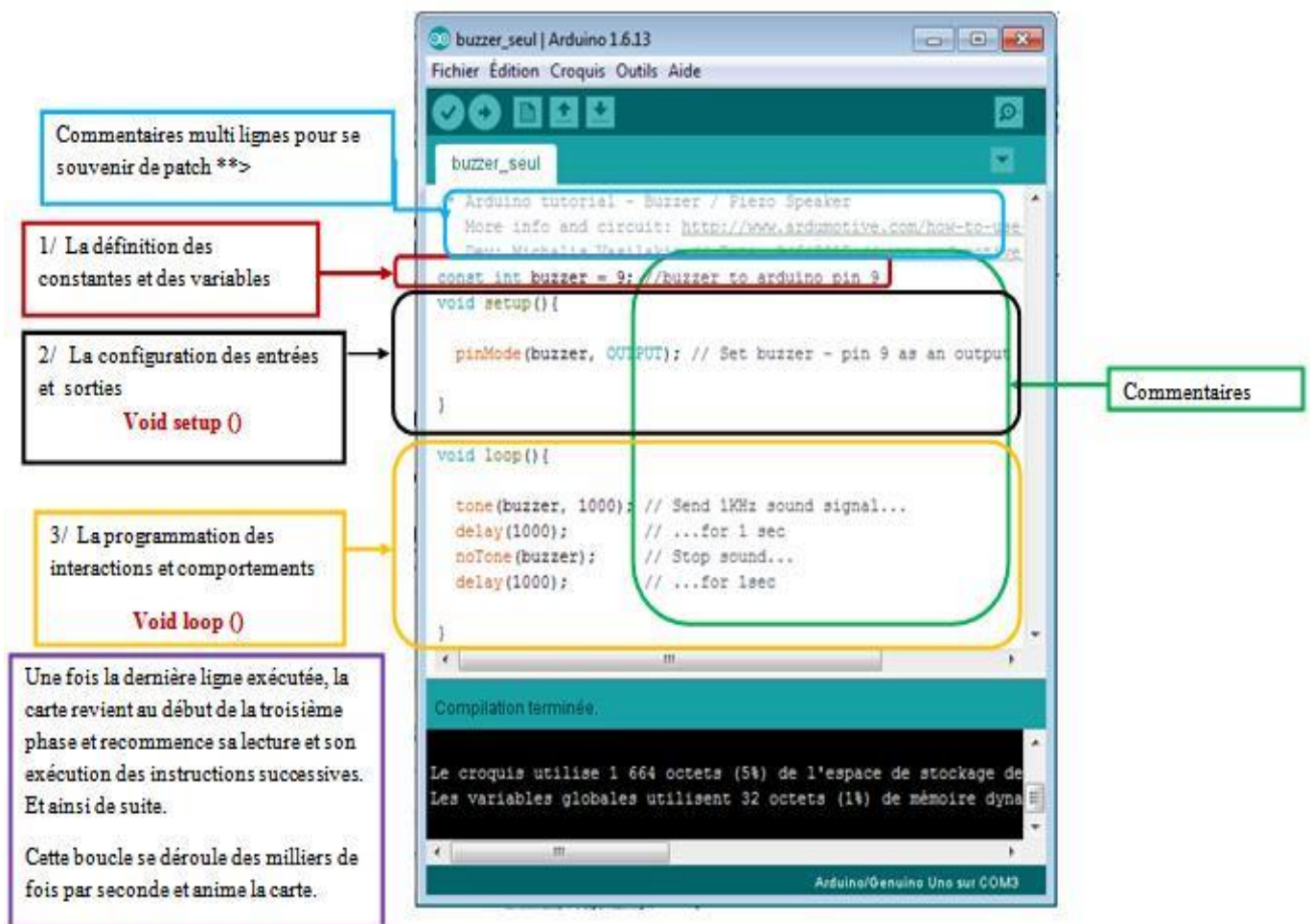


Figure 2.19: Structure générale du logiciel Arduino

D'après la figure ci-dessus, lorsqu'on écrit un code certains mots apparaissent dans différentes couleurs :

- **En orange :** Mots clés reconnus par le langage Arduino comme des fonctions existantes.
- **En bleu :** Mots clés reconnus par le langage Arduino comme des constantes.
- **En gris :** Commentaires qui ne seront pas exécutés par le programme.

Principe de fonctionnement du logiciel IDE Arduino

- ❖ On ouvre le logiciel ARDUINO.
- ❖ On compile et vérifie notre programme avec le logiciel ARDUINO.
- ❖ Si des erreurs sont signalées, on corrige notre programme.
- ❖ On réalise le montage électronique de façon correct.
- ❖ On copie notre programme sur la carte.
- ❖ On vérifie que notre montage fonctionne.

2.5. Conclusion

A travers ce chapitre nous avons présenté l'objectif principal de notre projet, qui est de mettre en œuvre un système de gestion de présence basé sur RFID.

Nous avons ensuite étudié le fonctionnement des différents outils matériels et logiciels que nous allons avoir besoin dans le chapitre suivant, afin de concevoir notre système.

CHAPITRE III

IMPLEMENTATION ET VALIDATION

3.1. Introduction

Après avoir présenté le principe de fonctionnement de notre système, ainsi que les différents outils matériels et logiciels dans les chapitres précédents, la question que nous comptons aborder dans ce chapitre est sa réalisation et conception.

Pour cela, ce chapitre est scindé en trois parties : la première partie est consacrée à la conception de la pointeuse basée sur le RFID, à partir des composants décrits au premier chapitre, tandis que la deuxième partie concerne la programmation, la troisième partie est pour les différents tests effectués.

En ce qui concerne le développement et la mise en œuvre du système, il devrait être en mesure d'aider les enseignants à gérer leur fréquentation scolaire systématique. Le système doit avoir une base de données contenant des informations sur les étudiants, et aussi une belle interface pour le rendre plus facile à utiliser.

3.2. Implémentation matérielle

Liste des pièces requises

Pour concevoir un système de pointage basé sur RFID à l'aide d'Arduino, nous avons besoin des composants suivants :

Nom du composant	Description
Carte Arduino	Un Arduino UNO R3
Module RFID	Un module RC522
Horloge RTC	Un module DS1307
Carte SD	
Buzzer	Un buzzer 5V
LED	Une LED rouge et une LED verte
Résistance	2 résistances 220ohm, 4 résistances 1Kohm

Potentiomètre	Un potentiomètre 10k
Afficheur LCD	Ecran LCD 16x2
Plaque d'essai	
Fils de connexion	Fils de cavalier
Bouton poussoir	Quatre interrupteur tactile à poussoir

Tableau 3.1 : Liste des composants utilisés

Schéma synoptique

Le système est basé sur une carte Arduino Uno. L'ensemble du matériel peut être divisé en quatre parties comme l'interface d'affichage Liquid Cristal Display (LCD), section d'interface de module RFID, section d'interface d'horloge en temps réel et SD Section d'interface de carte. Les blocs de construction de base sont présentés dans le schéma ci-dessous.

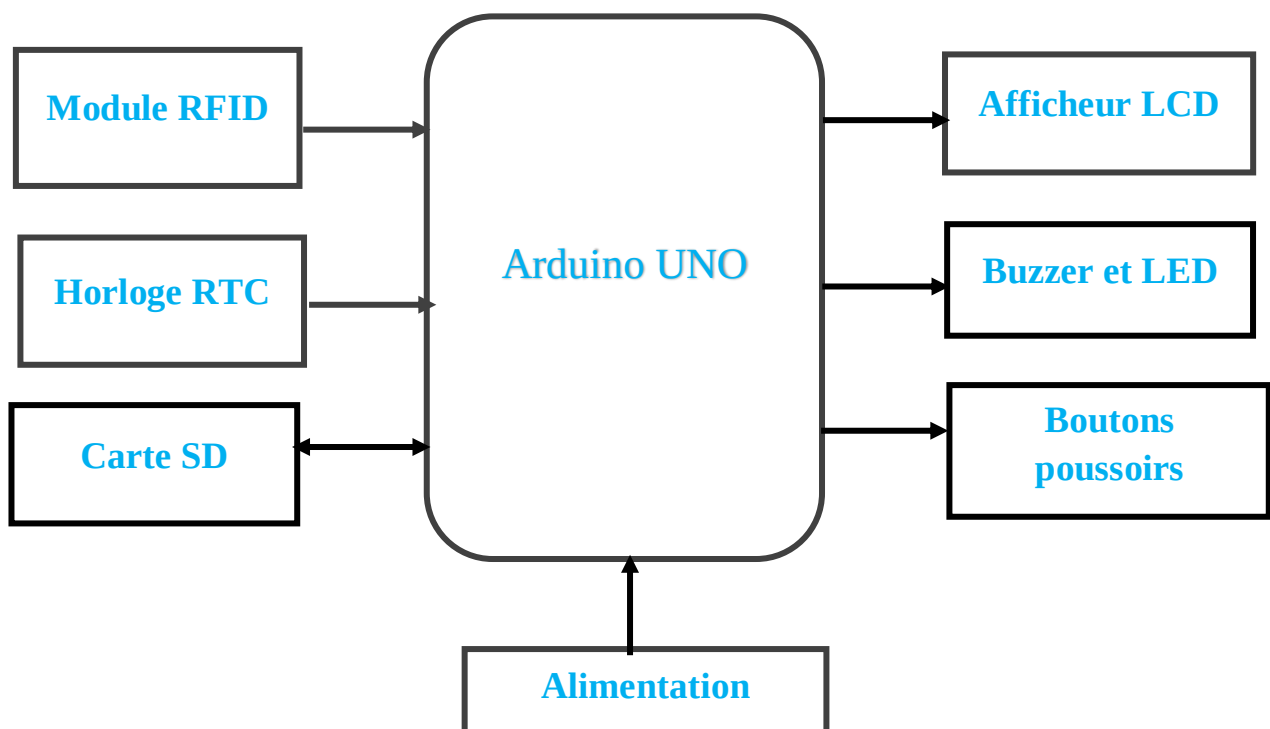


Figure 3.1 : Schéma synoptique du projet

Module RFID

La fonction principale du module lecteur RFID est de lire les données installées à l'intérieur de la carte.

La présence d'une clé d'autorisation peut éviter tout accès non autorisé. Il y a deux cas possibles lorsqu'un étudiant est en retard ou est à temps, le lecteur enverra les données au microcontrôleur pour les cartes. Pour chacun des cas « You are at time » ou « You are Late » s'afficheront sur le moniteur série.

i. Connexion entre l'Arduino et le module RFID RC522

Pin RFID RC522	Pin Arduino
SDA	Digital 10
SCK	Digital 13
MOSI	Digital 11
GND	GND
MISO	Digital 12
IRQ	Unconnected
RST	Digital 9
3.3V	3.3V

Tableau 3.2: Connexion entre les broches Arduino et RFID

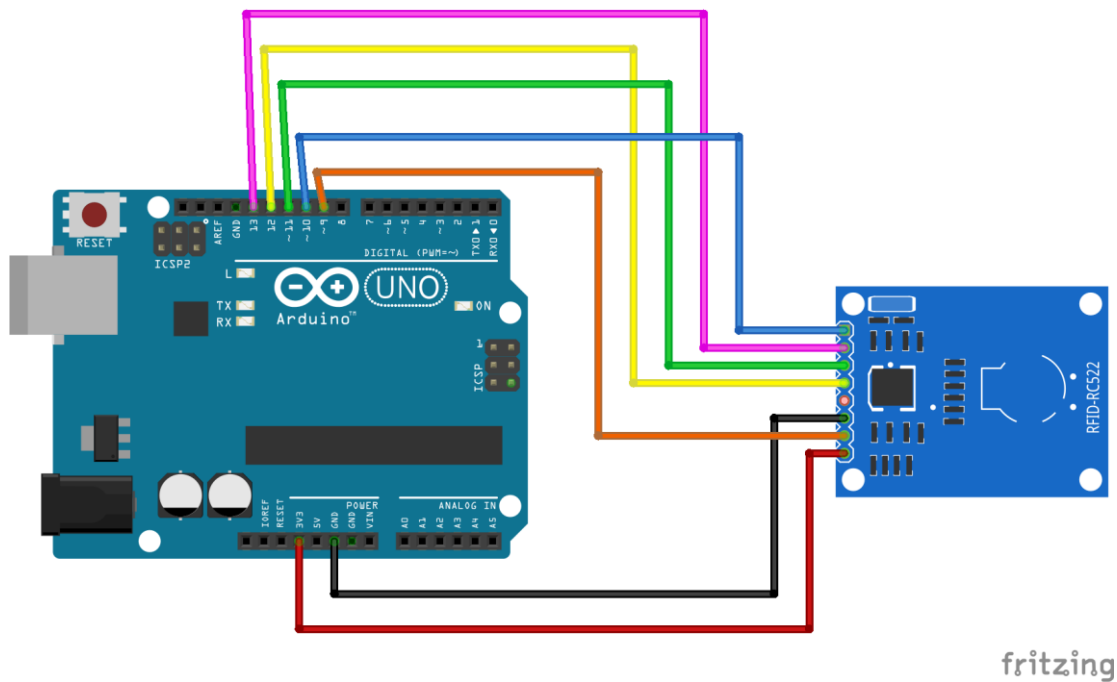


Figure 3.2 : Schéma électronique RFID et Arduino

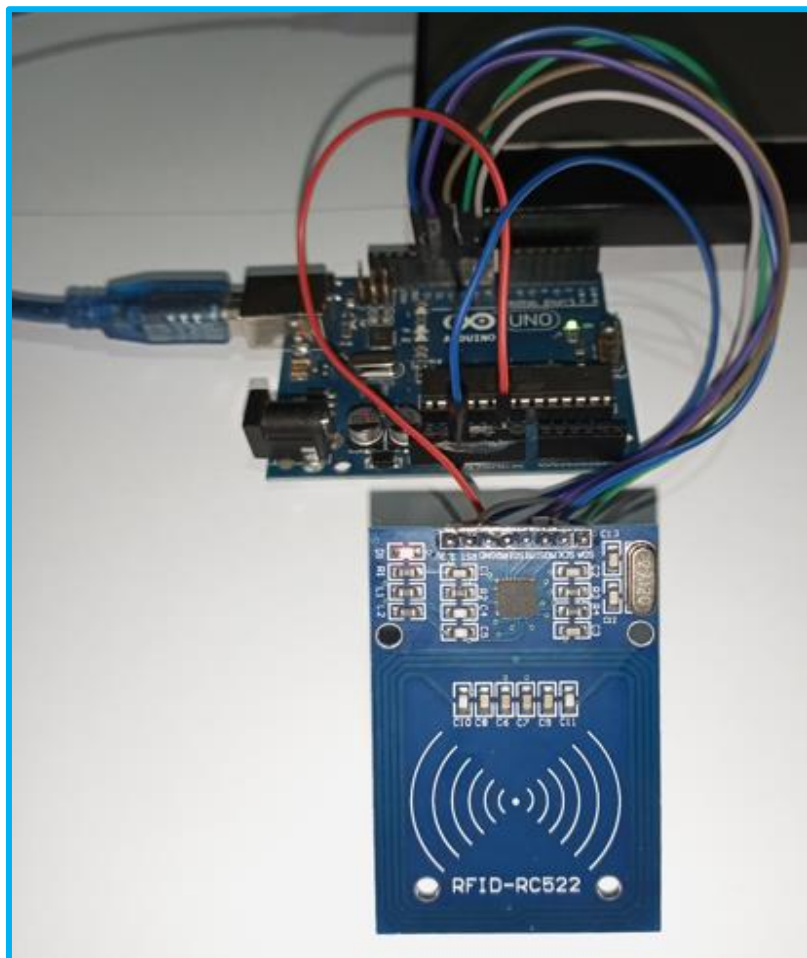


Figure 3.3 : câblage RFID et Arduino

Afficheur LCD

L'affichage à cristaux liquides (LCD) est un module d'affichage électronique qui fournit à l'utilisateur 4 bits interface avec matrice 5x7 pixels. Cet écran LCD a deux registres qui sont la commande et données où le protocole SPI est utilisé pour le connecter au microcontrôleur.

Dans ce projet, l'écran LCD affichera la date et l'heure, un menu qui permet de voir les données et informations de chaque étudiant. Lorsque la carte RFID a été glissée, l'identification, le nom de l'étudiant et son statue (absent, présent, en retard) seront ainsi affichées sur l'écran.

ii. Connexion entre l'Arduino et l'afficheur LCD

Pin Afficheur LCD	Pin Arduino
Vo	OUT du Potentiometer
VDD	5V
RS	Digital pin 3
RW	GND
E	Digital pin 2
D4	Analog pin A0
D5	Analog pin A1
D6	Analog pin A2
D7	Analog pin A3
A	5V
K	GND
VSS	GND

Tableau 3.3: Connexion entre les broches Arduino et LCD

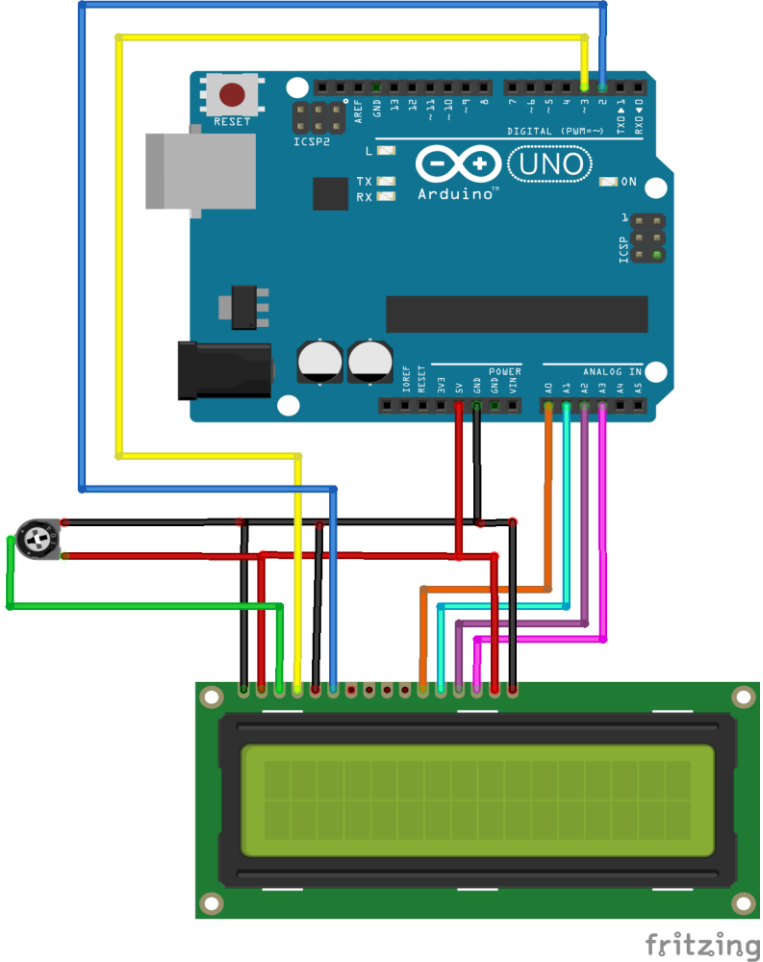


Figure 3.4: Schéma électronique Arduino et LCD

Buzzer et Leds

iii. Connexion entre l'ARDUINO, LED, buzzer

Composant	Pin Arduino
LED Rouge	Digital Pin 6
LED-Verte	Digital Pin 7
Buzzer	Digital Pin 4

Tableau 3.4: Connexion entre Arduino, led, buzzer

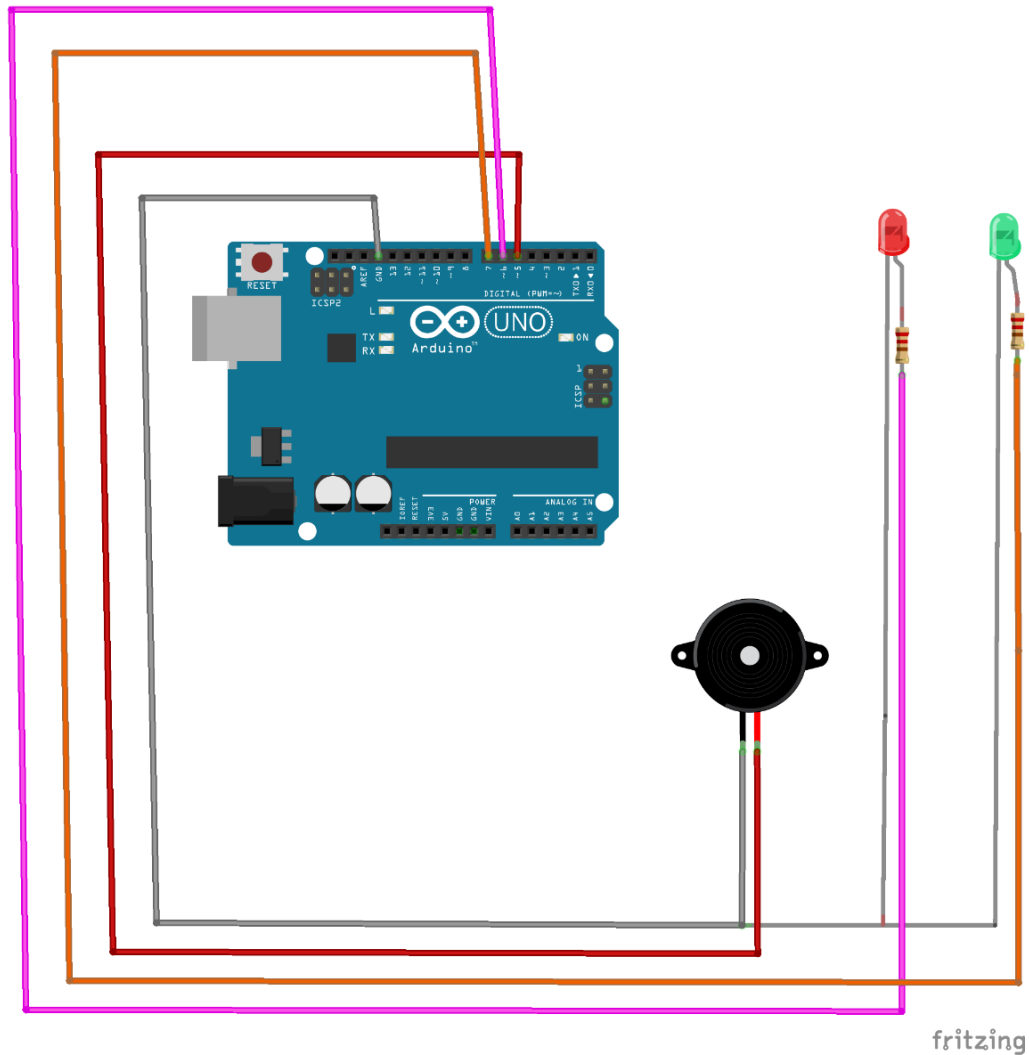


Figure 3.5: Schéma électronique Arduino, LED, Buzzer

Carte SD

Une carte Micro SD est utilisée comme pièce de stockage qui est connectée avec un microcontrôleur via le protocole SPI alimenté par une alimentation 3.3v.

Son format est un système de fichiers FAT32, et la base de données stockée à l'intérieur de cette carte est dans un fichier texte. Lorsqu'une étiquette est lue, son numéro d'identification UID et son heure seront enregistrés sur la carte afin qu'on puisse suivre les enregistrements.

iv. Connexion entre l'ARDUINO et la carte SD

Pin Carte SD	Pin Arduino
--------------	-------------

VCC	+3.3V
CS	Digital 4
MOSI	Digital 11
CLK	Digital 13
GND	GND
MISO	Digital 12

Tableau 3.5: Connexion entre Arduino et carte sd

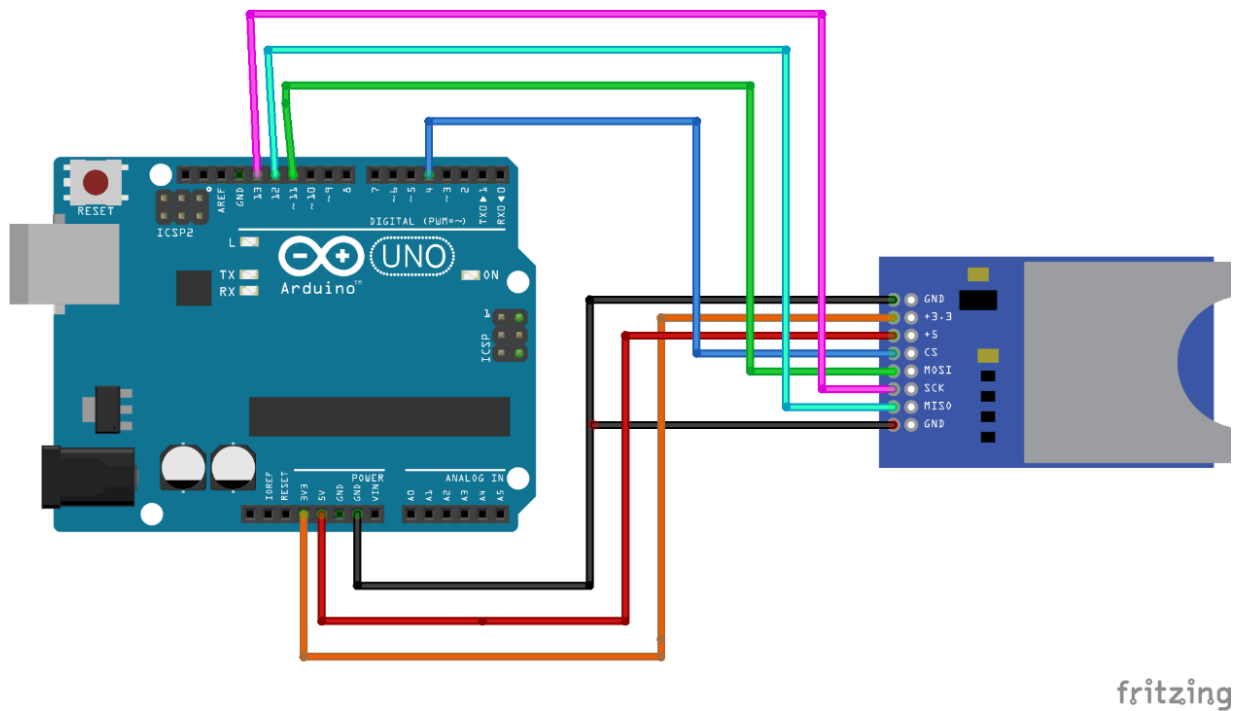


Figure 3.6: Schéma électronique Arduino et carte sd

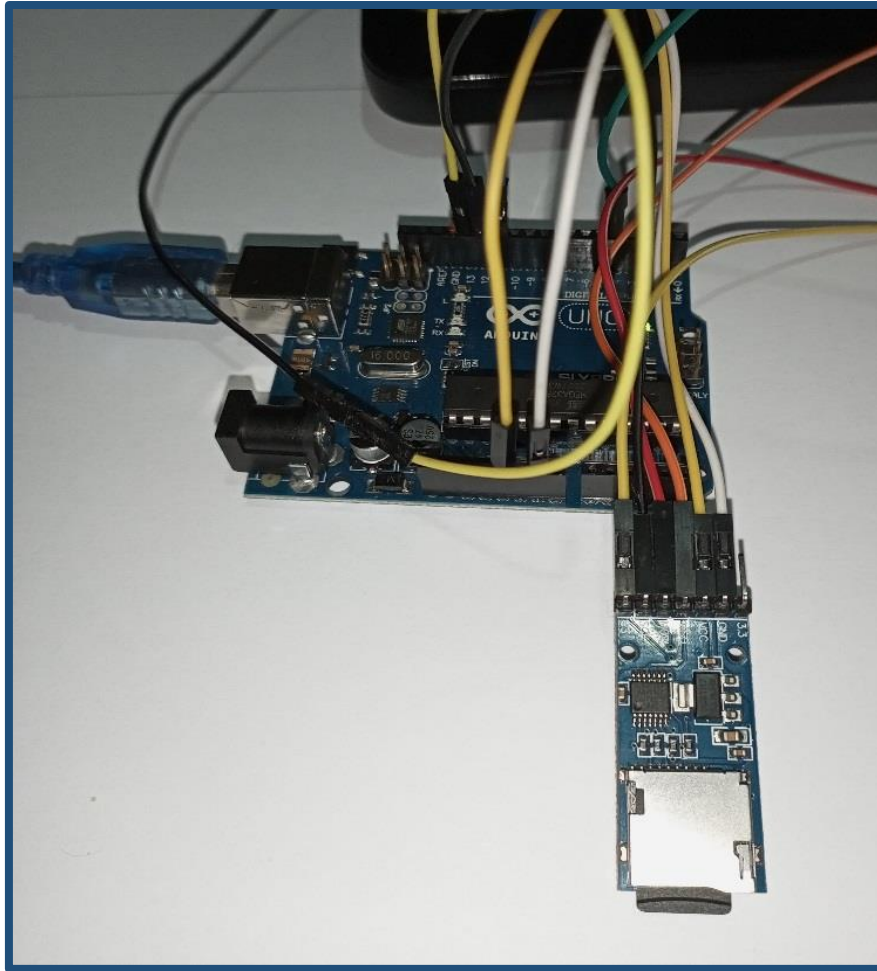


Figure 3.7: Câblage Arduino et carte SD

Horloge DS1307 (RTC)

L'horloge en temps réel série DS1307 (RTC) est alimentée par une pile bouton intégrée qui peut le faire fonctionner.

De plus, cette horloge stocke également les années, les mois, les semaines et les jours.

De cela, il corrige automatiquement les mois, y compris les années.

v. Connexion entre l'ARDUINO et l'horloge RTC DS1307

Pin composant	Pin Arduino Nano
VCC	+5V
SCL	Analog pin 5

SDA	Analog pin 4
GND	GND

Tableau 3.6: Connexion entre l'Arduino et RTC DS1307

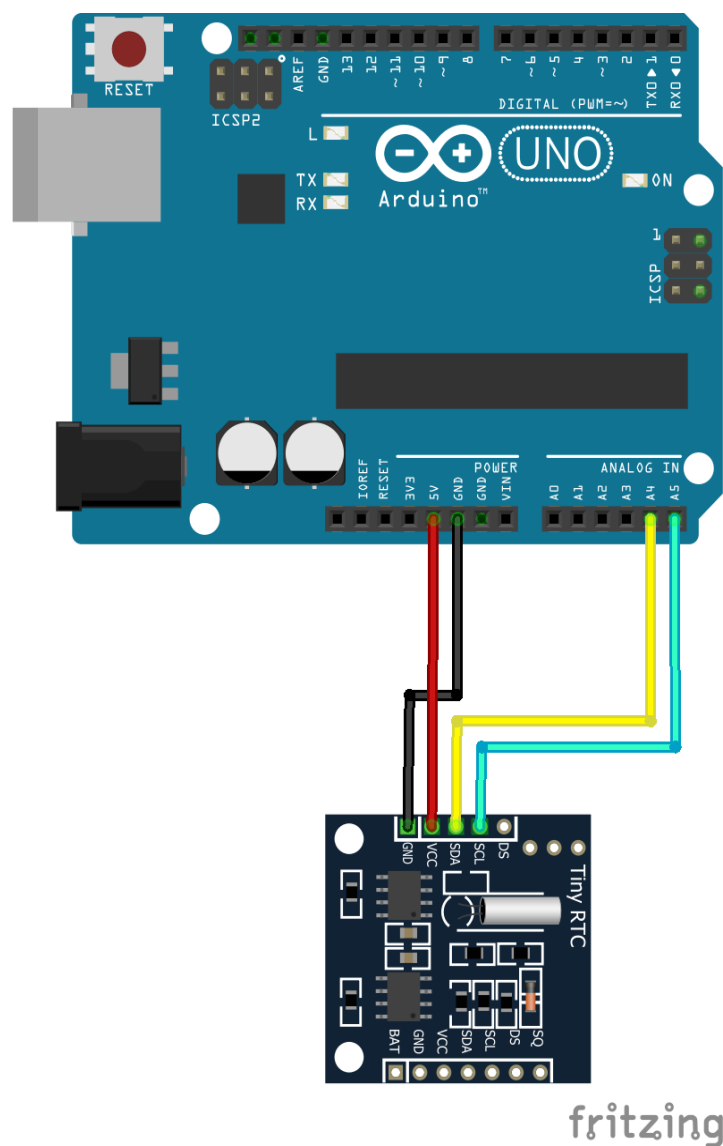


Figure 3.8: Schéma électronique Arduino et RTC DS1307

Schéma électronique du montage global sans les boutons poussoirs

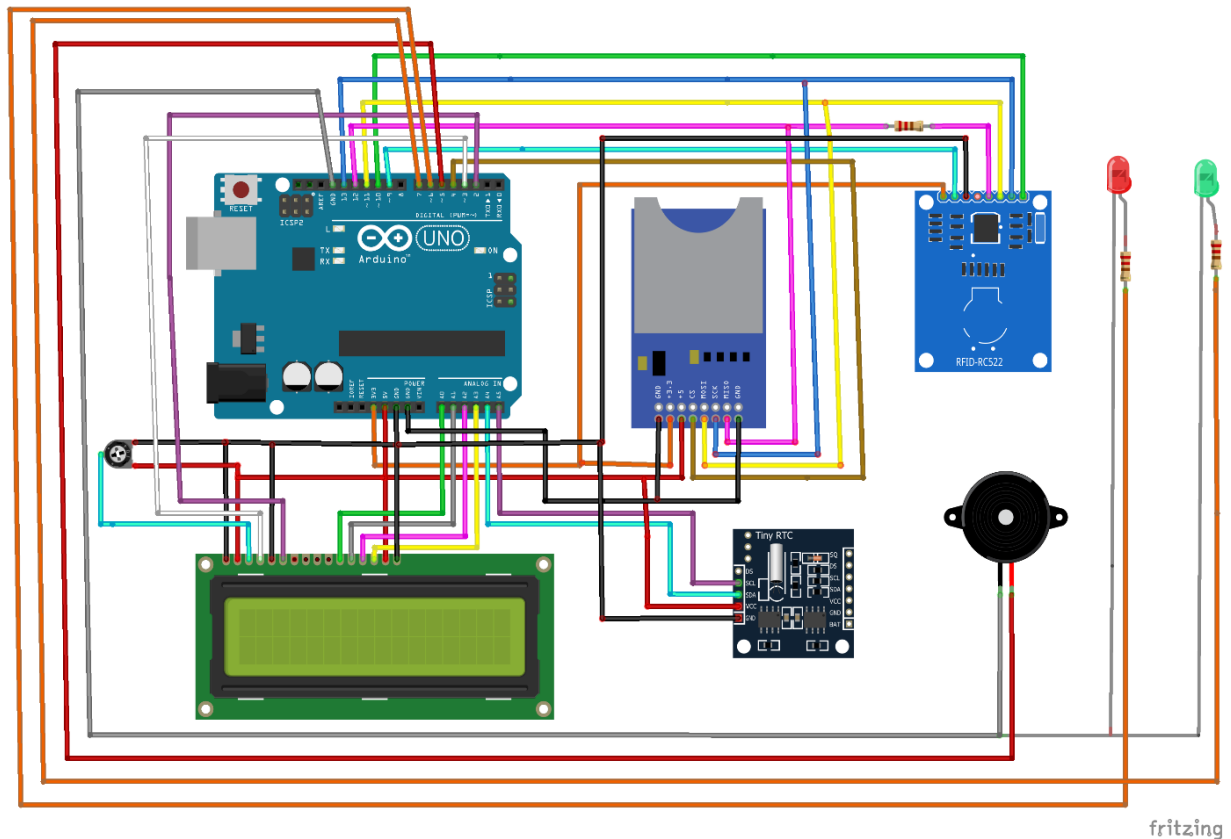


Figure 3.9: Schéma électronique du montage global avec FRITZING

Remarque :

On a rencontré un problème lors du branchement du circuit précédant.

- Le module RFID et la carte SD sont constitués de trois broches similaire MISO, MOSI, SCK, et doivent être brancher aux mêmes pins de l'Arduino, à savoir PIN 12 PIN 11 PIN 13, les deux modules fonctionnement séparément, avec toutes les recherches et tests effectués, ils ne fonctionnent toujours pas ensemble.
- Sans la synchronisation des deux modules ensemble et le stockage des données qu'on aurait besoins dans la carte SD, on ne peut ni afficher le menu ni utiliser les boutons poussoirs.

Schéma du montage global sur la plaque d'essai

Le circuit global de notre projet de réalisation du système de pointage basé sur RFID, est présenté par la figure suivante:

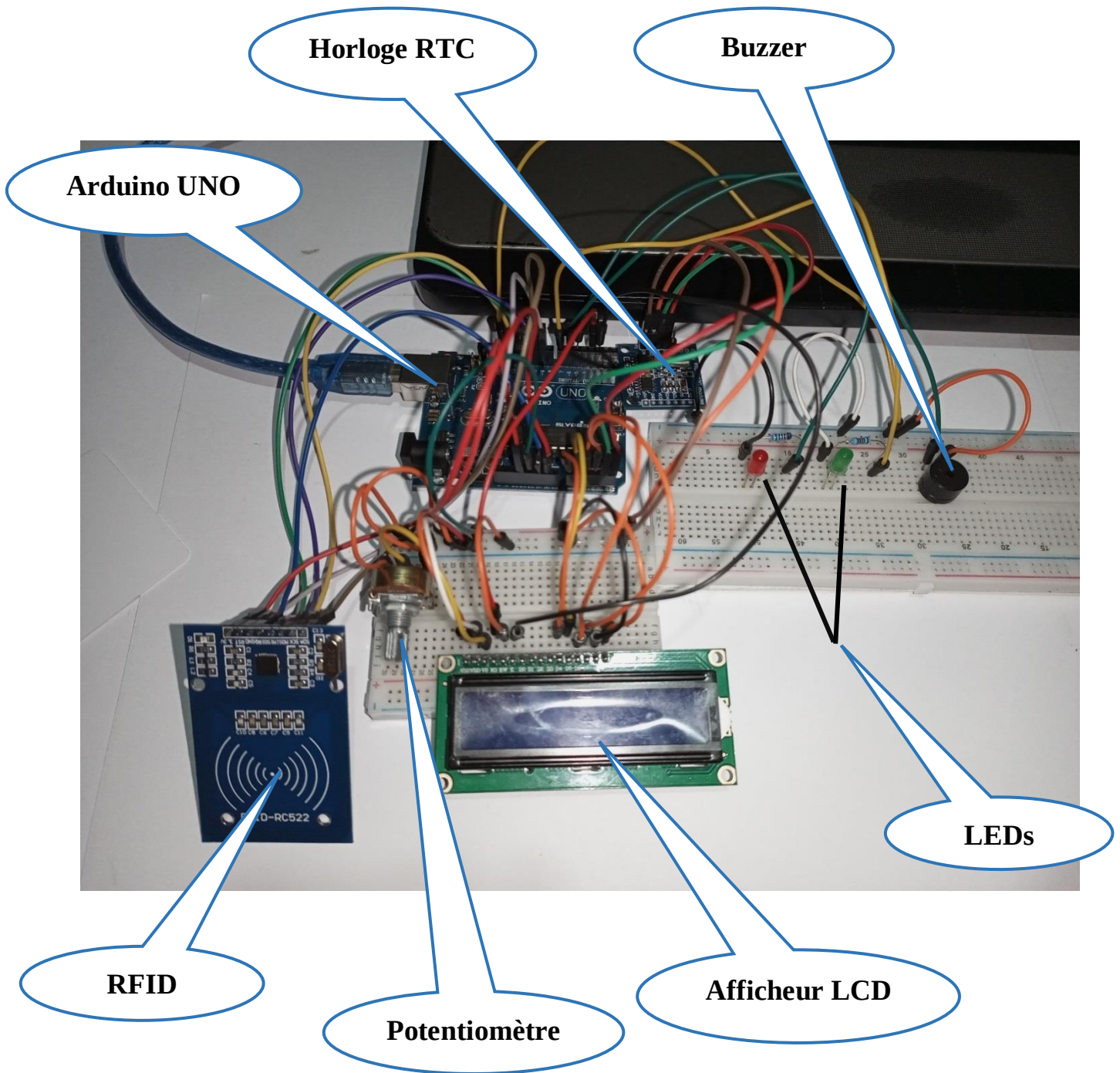


Figure 3.10: Circuit global du système de pointage

Prototype final de l'interface du système

On a acheté une boîte de dérivation électrique 7cm de hauteur, 18cm de largeur et 15cm de longueur.

Ensuite, on la découpé à l'aide d'un cutteur et d'une meuleuse, afin de mettre en place les composants et de les fixer avec une colle de fixation.

L'interface obtenu est représenté dans la figure ci-dessous.



Figure 3.11 : Interface de la poutreuse

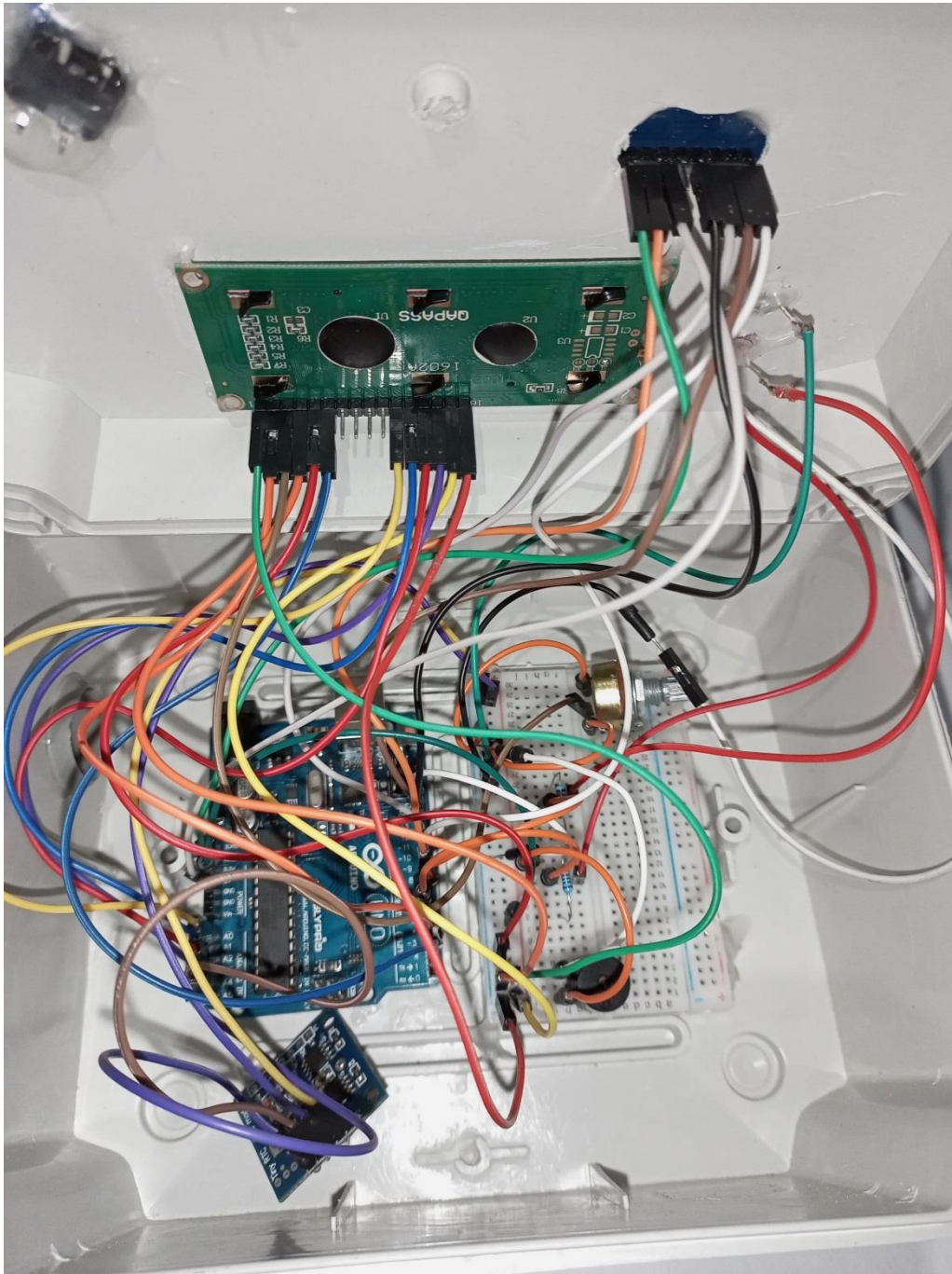


Figure 3.12 : Vu du circuit à l'intérieure de l'interface

3.3. Implémentation logiciel (Arduino IDE)

La fabrication du matériel n'est pas suffisant pour une bonne utilisation du système, les logiciels jouent un rôle majeur pour le bon fonctionnement du matériel.

Dans ce projet, le langage utilisé est la programmation Arduino DE.

Après avoir terminé la conception et le développement du système, un logiciel d'application sera développé.

Il y a un groupe principal de logiciel d'application qui doit être pris en compte, le processus de la carte qui sera glissée sur l'appareil portable.

Une fois la carte RFID glissée sur le lecteur RFID, l'état passe en mode lecture. Ensuite, le lecteur identifiera la carte et mémorisera les informations dans la carte SD qui peuvent être appelées en mode écriture.

Test des modules

Le test des modules est l'une des phases importantes de ce projet. Nous allons tout d'abord tester et s'assurer que les modules fonctionnent comme prévu.

Les différents composants sont le module RFID, le module de carte Micro SD, module LCD et Horloge RTC DS1307.

Arduino UNO avec le RFID , LEDs et buzzer

Nous allons tester le système de pointage à base de RFID, en utilisant une led rouge et une led verte, ainsi qu'un buzzer, deux résistances de 220ohm.

A chaque fois qu'on glisse une étiquette RFID à côté du lecteur RFID, le buzzer émet une impulsion sonore, si l'utilisateur est en retard ou si la carte n'est pas valide, la LED rouge s'allume; si l'utilisateur est à l'heure, la LED verte s'allume. On verra le numéro d'identification de l'étiquette et le statut sur le moniteur série.

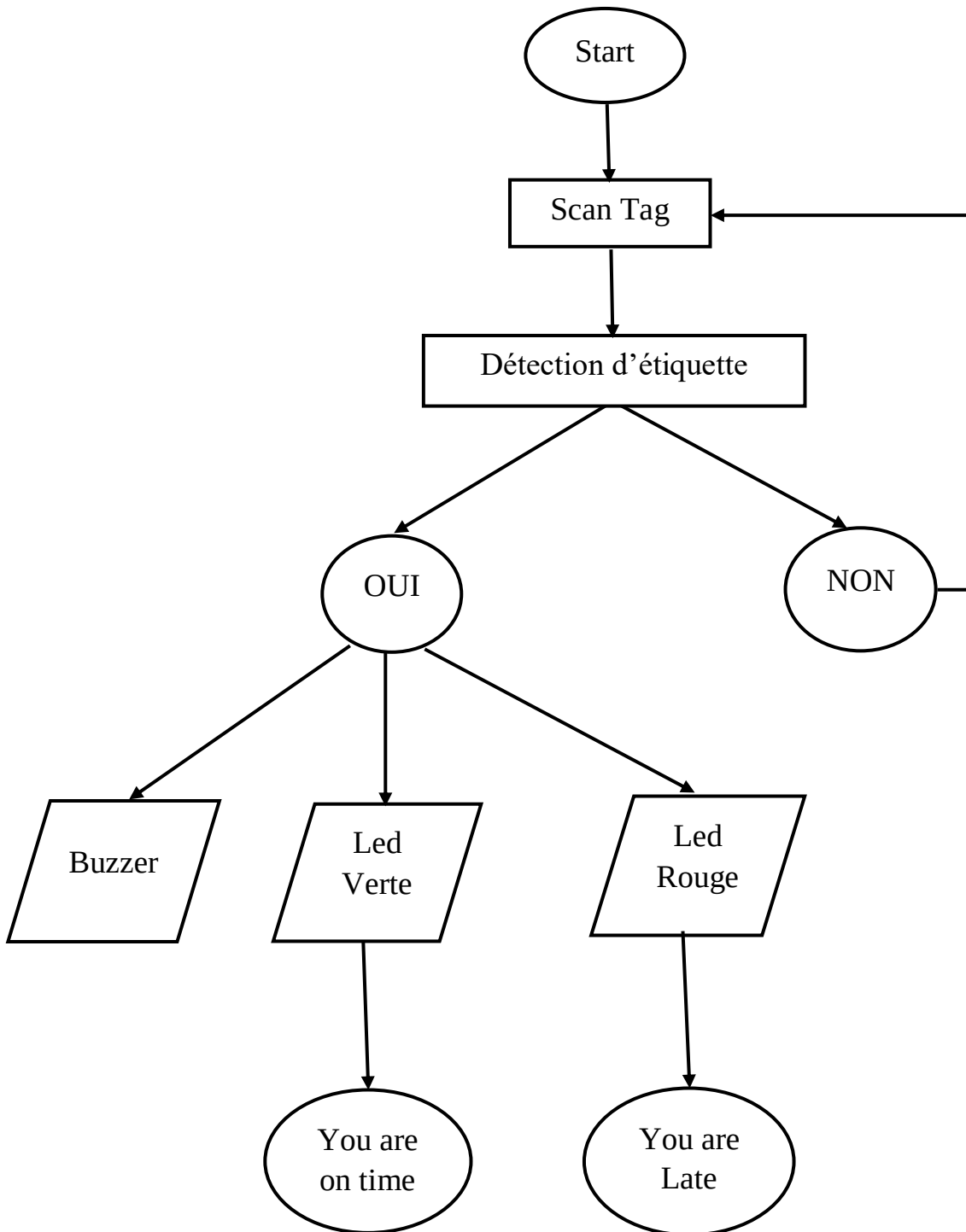


Figure 3.13: Organigramme de l'intégration RFID (codage)

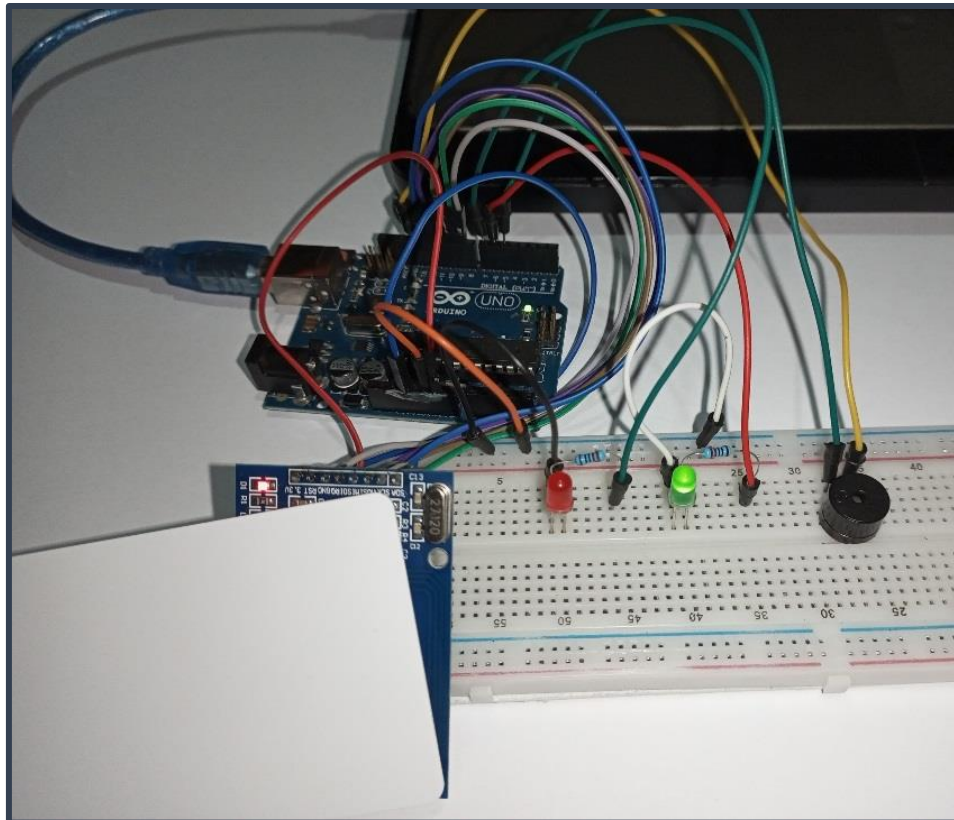


Figure 3.14: Led verte allumée

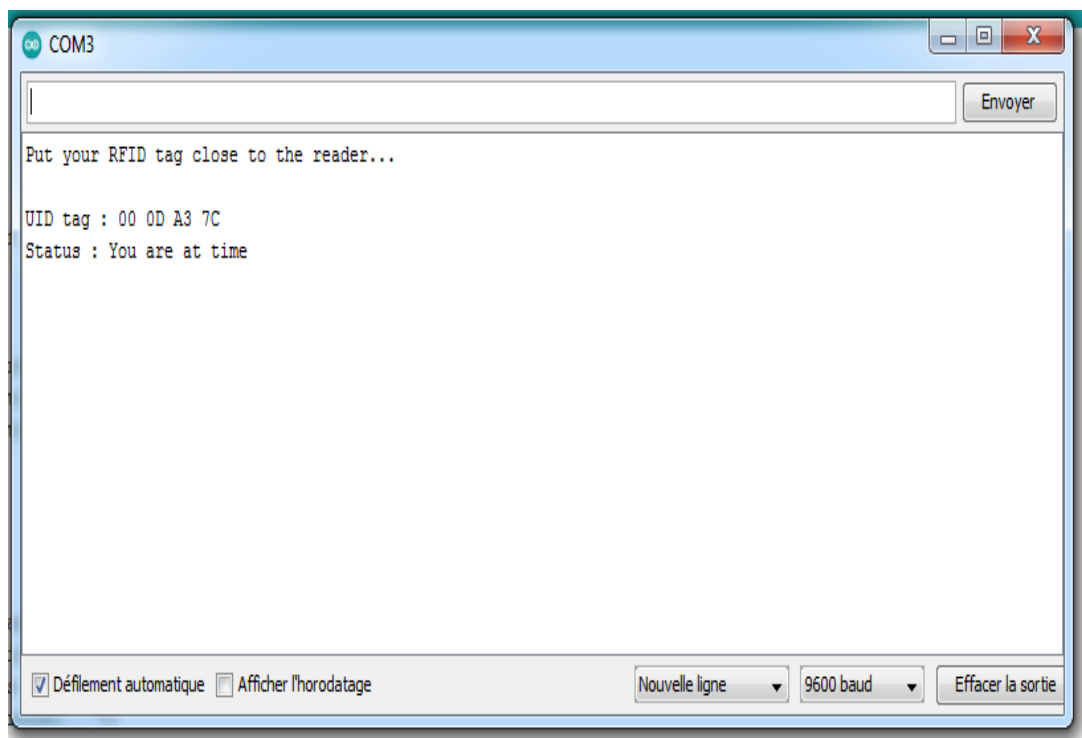


Figure 3.15: Affichage “You are on time” sur le moniteur série

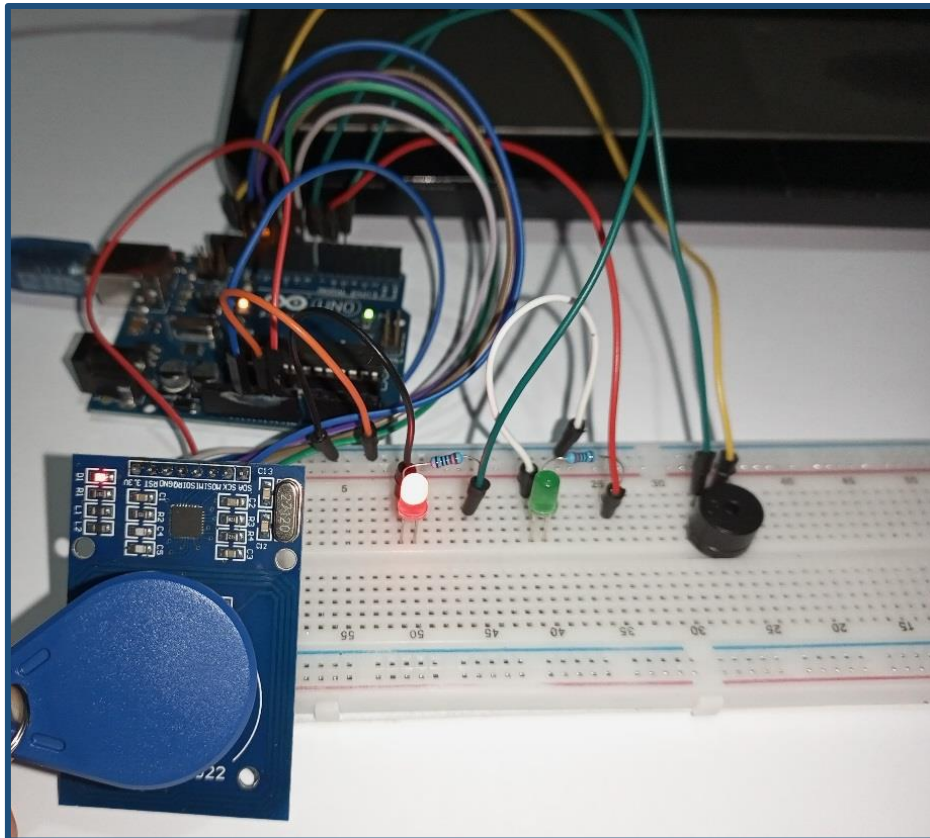


Figure 3.16: Led rouge allumée

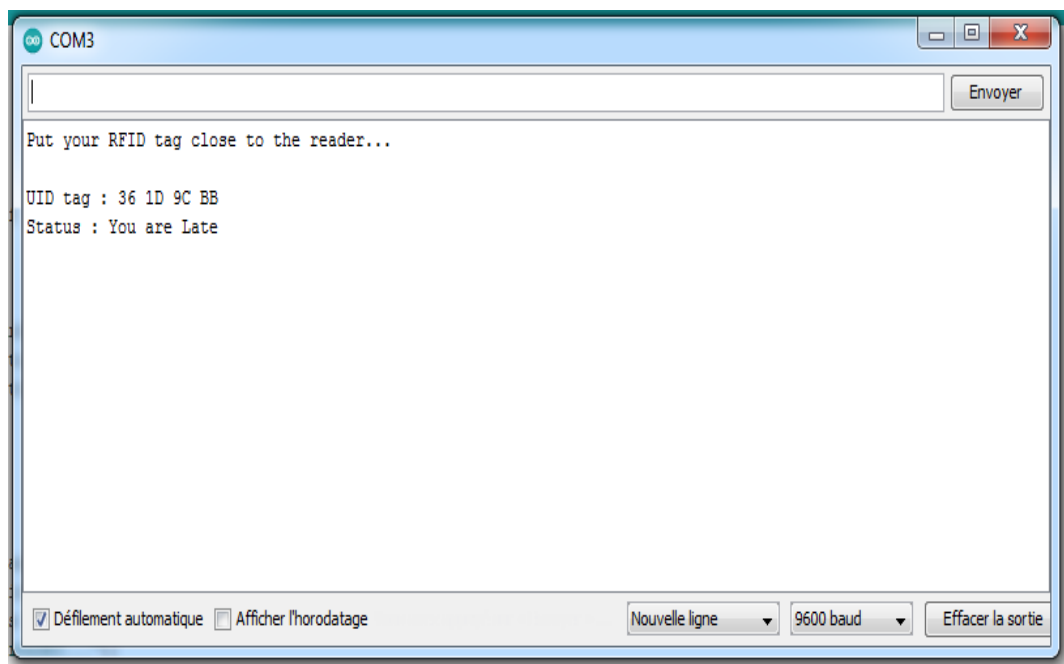


Figure 3.17: Affichage “You are Late” sur le moniteur série

Arduino UNO avec Carte SD

Dans cette étape nous allons tester la carte SD, en créant un fichier TXT appelé « DATA1 » sur la carte, puis nous allons stocker les caractères suivant : « Arduino » « SD card » dans le fichier créer.

La première étape lors de l'utilisation du module de carte SD avec Arduino consiste à formater la carte SD en FAT16 ou FAT32.

- 1) On l'insère dans un ordinateur. Allez dans Poste de travail et faire un clic droit sur la carte SD. On sélectionne Format.
- 2) Une nouvelle fenêtre apparaît. Sélectionner FAT32, appuyer sur Démarrer pour initialiser le processus de formatage
- 3) Insérer la carte SD formatée dans le module de carte SD, et connecter le module de carte SD à l'Arduino.

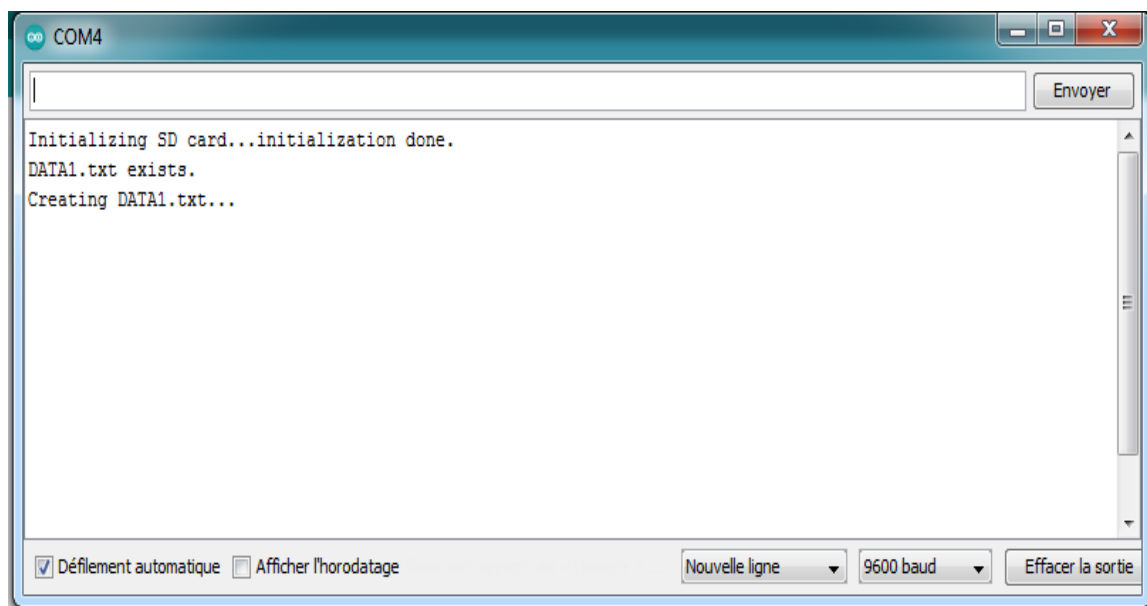


Figure 3.18: Fichier DATA1 existe

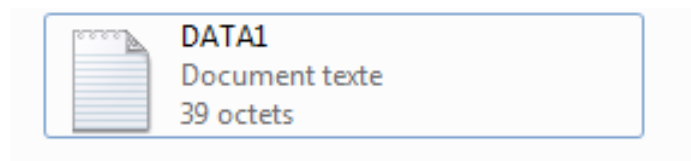


Figure 3.19: Fichier DATA1

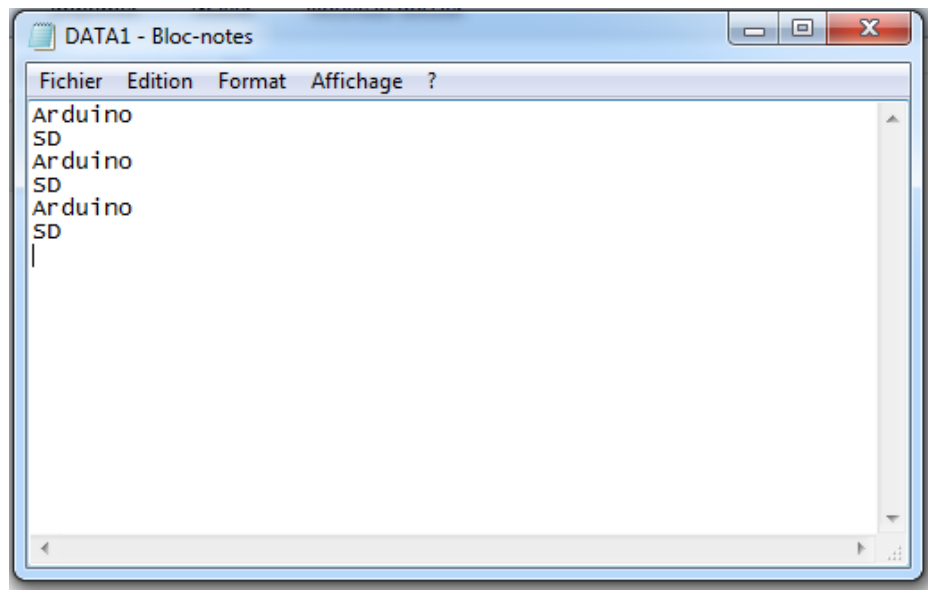


Figure 3.20: Caractères stockés dans le fichier DATA1

Arduino UNO avec écran LCD et horloge RTC DS1307

Nous allons tester l'écran LCD et l'horloge RTC DS1307, en affichant l'heure et la date.

La figure ci-dessous montre l'heure et la date sur l'écran LCD.

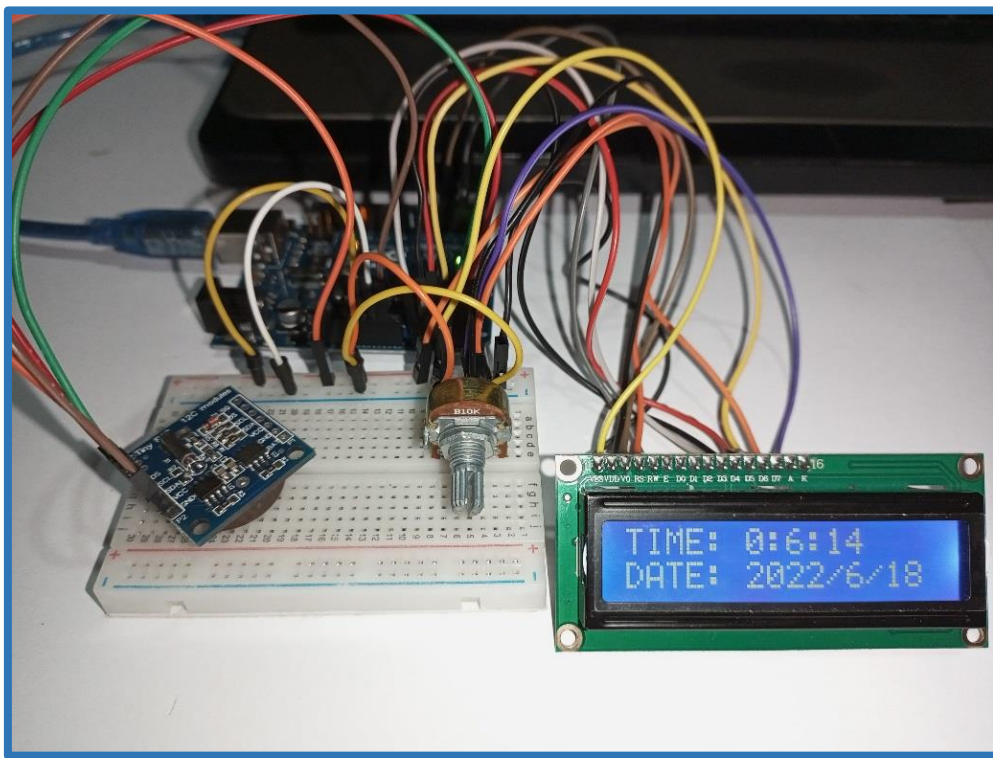


Figure 3.21: Affichage de la date et de l'heure sur LCD

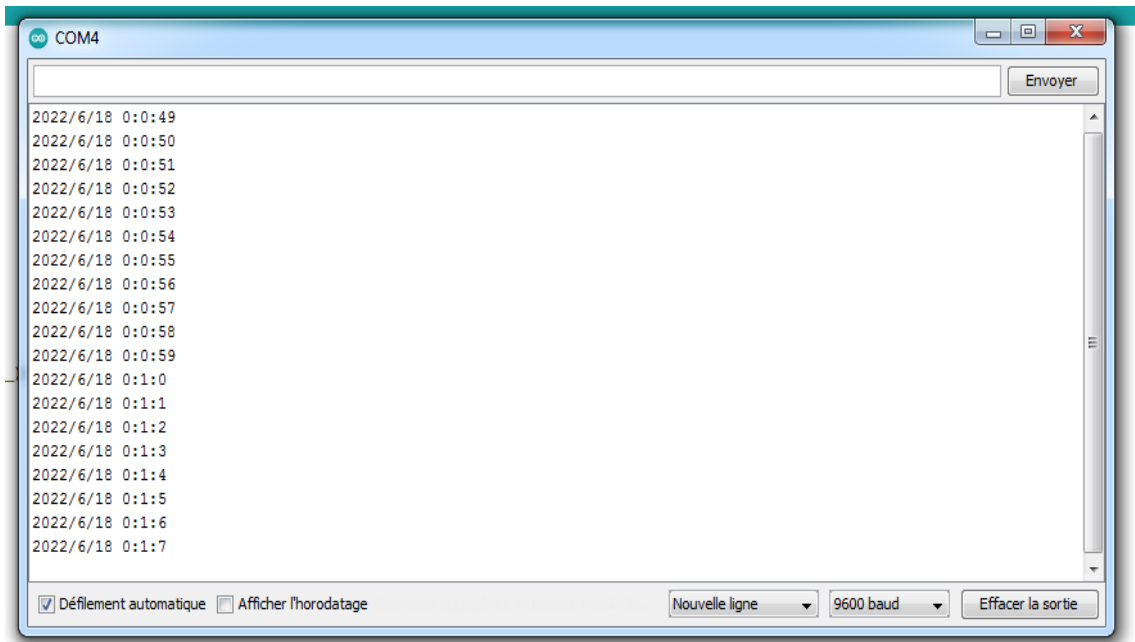


Figure 3.22: Affichage de la date et de l'heure sur le moniteur série

Etapes de Programmation du montage global sans les boutons poussoirs

Le code commence par importer les bibliothèques nécessaires. Le MFRC522 pour le lecteur RFID, le SD pour le module carte SD et le RTCLib pour le RTC. On inclue également la bibliothèque SPI pour la communication SPI avec le module RFID et carte SD, et LiquidCrystal pour l'écran LCD.

```
#include <SPI.h> //include library code for SPI comms
#include <MFRC522.h> //include library code for RFID
#include <LiquidCrystal.h> //include library code for LCD1602 I2C module
#include <RTCLib.h> //include library code for the RTC
#include <Wire.h> //include library code for the RTC
```

Ensuite, on définit les broches pour le lecteur RFID, lecteur LCD et le module de carte SD. Pour le RFID, la broche SCK (CS_RFID) est connectée à la broche 10 et la broche RST (RST_RFID) est connectée à la broche 9. Pour le module de carte SD, la broche Chip Select (CS_SD) est connectée à la broche 4.

```
// .....  
const int SS_PIN = 10; //RFID SS (SDA) pin attached to Arduino Uno digital pin 10  
const int RST_PIN = 9; //RFID RST pin attached to Arduino Uno digital pin 9  
const int redLEDPin = 7; //red LED attached to Arduino Uno digital pin 7  
const int greenLEDPin = 6; //green LED attached to Arduino Uno digital pin 6  
const int buzzerPin = 5; //active piezo buzzer attached to Arduino Uno digital pin 5  
LiquidCrystal lcd(3, 2, A0, A1, A2, A3);
```

Ensuite, on crée une instance pour le RFID et pour le RTC :

```
RTC_DS1307 rtc; //create instance RTC  
MFRC522 mfrc522(SS_PIN, RST_PIN); //create MFRC522 instance.
```

Les lignes suivantes créent des variables pour définir l'heure et la minute d'enregistrement. Dans ce cas, nous définissons l'heure d'enregistrement par exemple à 20h06. On peut modifier l'heure d'enregistrement en modifiant ces valeurs :

// Définir l'heure d'arrivée

```
// Define check in time  
const int checkInHour = 20;  
const int checkInMinute = 06;
```

On doit également créer des variables pour maintenir l'heure de vérification de l'utilisateur. Ces variables enregistreront l'heure à laquelle une certaine étiquette UID a été lue. Les variables suivantes contiennent l'heure d'enregistrement et la minute d'enregistrement.

//Variable pour maintenir l'enregistrement de l'utilisateur

```
int userCheckInHour;  
int userCheckInMinute;
```

Enfin, on attribue les numéros de broches aux LED et au buzzer.

// Pins pour LEDs et le buzzer

```
const int redLEDPin = 8; //red LED attached to Arduino Uno digital pin 8  
const int greenLEDPin = 7; //green LED attached to Arduino Uno digital pin 7  
const int buzzerPin = 6; //active piezo buzzer attached to Arduino Uno digital pin 6
```

La fonction `verifyCheckIn()` compare simplement l'heure d'enregistrement de l'utilisateur avec l'heure d'enregistrement prédéfinie et donne un retour en conséquence.


```
void verifyCheckIn() {
  if ((userCheckInHour < checkInHour) || ((userCheckInHour == checkInHour) && (userCheckInMinute <= checkInMinute))) {
    digitalWrite(greenLEDpin, HIGH);
    delay(500);
    digitalWrite(greenLEDpin, LOW);
    lcd.setCursor(8, 0);
    lcd.print("on time");
    delay(5000);
    lcd.clear();
  }
  else {
    digitalWrite(redLEDpin, HIGH);
    delay(500);
    digitalWrite(redLEDpin, LOW);
    lcd.setCursor(8, 0);
    lcd.print("late");
    delay(5000);
    lcd.clear();
  }
}
```

Lorsqu'on place une étiquette RFID à côté du lecteur RFID, on verra le nombre d'étiquettes/cartes RFID et leurs statuts. Ce sera affiché sur l'écran LCD. On verra également l'état avec la LED verte/rouge. Si l'utilisateur est en retard, la LED rouge s'allume ; si l'utilisateur est à l'heure, la LED verte s'allume.

Ensuite, dans **setup ()**, on définit les LED et le buzzer comme sorties.

// Définir les LED et le buzzer comme sorties

```
pinMode(redLEDpin, OUTPUT); //sets redLEDpin as OUTPUT
pinMode(greenLEDpin, OUTPUT); //sets greenLEDpin as OUTPUT
pinMode(buzzerPin, OUTPUT); //sets buzzerPin as OUTPUT
... ..
```

Après cela, chaque module est initialisé.

```
Serial.begin(9600); //initiate a serial communication
lcd.begin(16,2);
while(!Serial);
SPI.begin(); //initiate SPI bus
mfr522.PCD_Init(); //initiate MFRC522p
```

Configuration pour le RTC et définition de la date et l'heure à laquelle cette esquisse a été compilée


```
// Setup for the RTC
if (!rtc.begin()) {
  lcd.println("Couldn't find RTC");
  Serial.flush();
  while (1); delay(10);
}
if (!rtc.isrunning()) {
  // following line sets the RTC to the date & time this sketch was compiled
  Serial.println("RTC is NOT running!");
  rtc.adjust(DateTime(F(__DATE__), F(__TIME__)));
}
```

Afficher sur l'écran LCD un message de bienvenu

```
lcd.setCursor(2, 0);           //setting display position
lcd.print("Welcome to");
lcd.setCursor(0, 1);           //setting display position
lcd.print("Control System");
delay(2000);
lcd.clear();
}
```

Après avoir étudié les fonctions créées, la boucle **Loop ()** est assez simple à comprendre.

Tout d'abord, le code vérifie si une étiquette RFID a été glissée. Si oui, il lira l'UID RFID, enregistrera l'UID et l'heure d'arrivée sur la carte SD, puis il donnera un retour à l'utilisateur en émettant une impulsion sonore grâce au buzzer, et en allumant l'une des LED.

3.4. Test et validation

Dans ce chapitre, les résultats seront affichés en détail à partir de la lecture de l'étiquette RFID jusqu'à ce qu'ils soient stockés et affichés dans l'ordinateur. Le nom de la personne identifiée au sein de l'Arduino et correspondant à chaque carte s'affichera sur l'écran LCD lors de la lecture du Tag RFID et en même temps il sera stocké dans un Fichier TXT sur la SD Et ensuite les informations seront glissées vers le fichier Excel pour être affichées sur l'écran de l'ordinateur.

Premièrement, lorsque la carte Arduino est mis sous tension, alors l'écran LCD affiche un message de bienvenu « Welcome to Control System », puis il affiche l'heure et la date en nous demandons de scanner une carte RFID sous message tel que « Scan your card... ».

Les figures ci-dessous : figure 3.23, figure 3.24 montrent le déroulement de ces étapes.



Figure 3.23: Affichage du message de bienvenu sur LCD



Figure 3.24: Affichage de la date/heure sur LCD

Avant de commencer à mentionner les détails de la lecture du Tag RFID. Lorsque vous placez un Tag sur le lecteur RFID, un buzzer émet un son, un flash est émis par la LED.

Le Tag RFID a été lu. Dans les figures suivantes, nous allons examiner les informations de lecture RFID où quatre étiquettes ont été programmées et lues par le système.

A commencer par les figure 3.25 et figure 3.26 qui montrent qu'effectivement, un étudiant qui arrive à l'heure et glisse sa carte à côté du lecteur, une led verte est allumée, et l'affichage de l'heure d'arrivée et du statut de l'étudiant « On time » sur l'écran LCD.

Les figure 3.27 et figure 3.28 montrent un étudiant arrivant en retard et glisse sa carte à côté du lecteur, une led rouge est allumée, et l'affichage de l'heure d'arrivée et du statut de l'étudiant « Late » sur l'écran LCD.

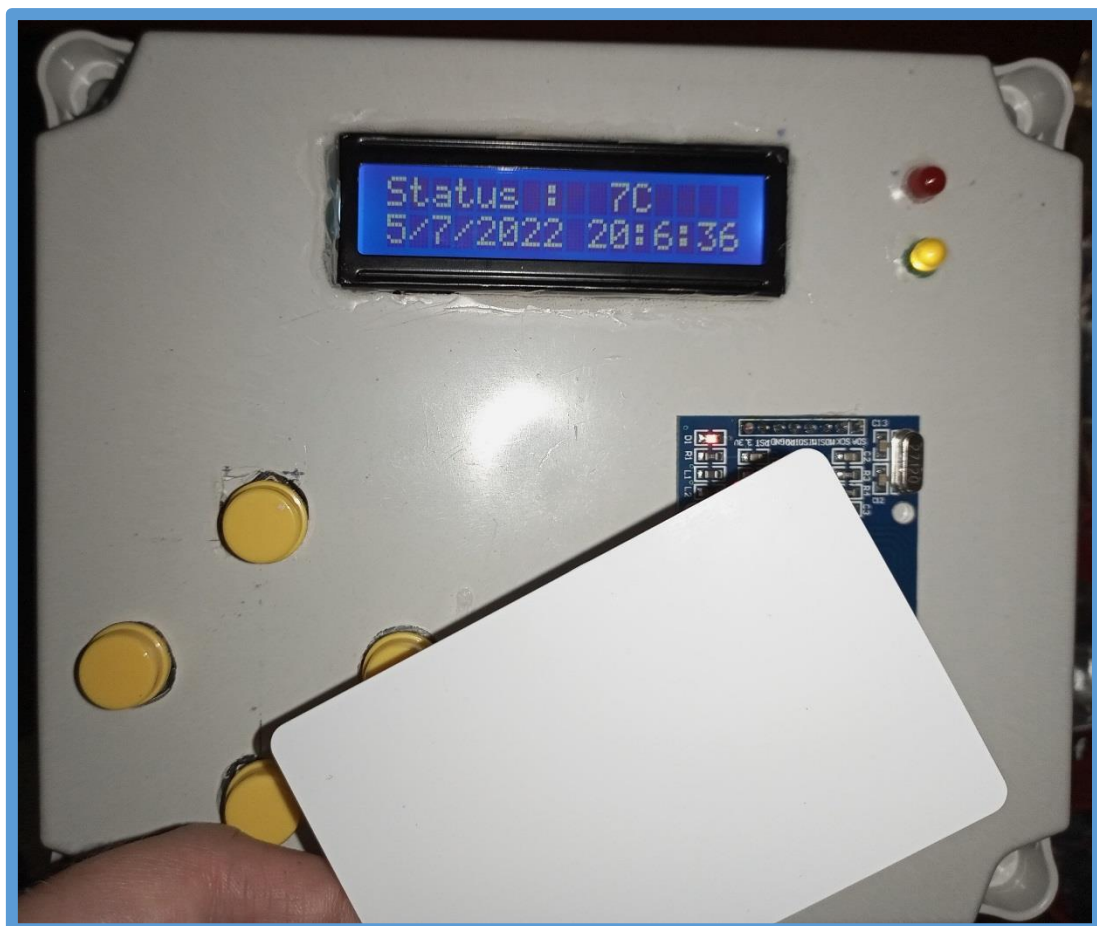


Figure 3.25: Led verte allumée et affichage de l'heure d'arrivée de l'étudiant



Figure 3.26: Affichage du statut « On time » de l'étudiant



Figure 3.27: Led rouge allumée et affichage de l'heure d'arrivée de l'étudiant



Figure 3.28: Affichage du statut « Late » de l'étudiant

3.5. Conclusion

Dans ce chapitre, on a pu avoir fait la conception et réalisation d'un système de pointage basé sur la technologie RFID, à l'aide d'une carte Arduino UNO.

On a présenté les étapes qui ont consisté à l'interconnexion entre les différents composants, ensuite la programmation de la carte à microcontrôleur ARDUINO. Nous avons parlé des différents tests effectués et les résultats obtenus.

CONCLUSION GENERALE

La recherche proposée est menée pour mettre en œuvre la technologie RFID dans le domaine éducatif pour résoudre les problèmes de la méthode de présence manuelle. Les instructeurs doivent toujours noter la présence de leurs étudiants à chaque cours. La majorité des institutions du monde entier utilisent encore la méthode traditionnelle pour enregistrer les présences.

Il existe plusieurs problèmes associés aux méthodes de présence électroniques et manuelles, telles que la perte de temps, la tendance aux erreurs, l'implication d'efforts humains, la fatigue et l'inefficacité.

Le pointage est un aspect important dans chaque établissement d'enseignement et lieu de travail. Cette recherche vise à répondre aux principales limites des méthodes existantes de prise de présence en proposant un système de pointage entièrement automatisé utilisant la technologie RFID.

Le système de présence basé sur RFID fournit une méthode pratique de prise de présence en fournissant un lecteur RFID pour identifier de manière unique l'étudiant portant une étiquette RFID et enregistrer ses données.

Parmi les problèmes rencontrés pendant la réalisation :

- ❖ Le délai de réalisation du projet a été très court.
- ❖ Problème pour synchroniser le module RFID et Carte SD ensemble, ce qui a engendré un problème pour afficher le menu principal.

REFERENCES

- [1] Frontier ehr “must have features of your time attendance system”, article disponible à l’adresse URL : <https://www.frontier-ehr.com/2017/04/must-have-features-of-attendance-system/>
- [2] K Sudha, Shirish Shinde, Aris Abdugani. Barcode based Student Attendance System. **International Journal of Computer Applications** (0975 – 8887) Volume 119 – No.2, June 2015.
- [3] NOR AINI BT ZAKARIA. THE DEVELOPMENT OF STUDENT ATTENDANCE SYSTEM. UNIVERSITI TEKNOLOGI PETRONAS.
- [4] Frédéric Massicotte : LA BIOMÉTRIE, SA FIABILITÉ ET SES IMPACTS SUR LA PRATIQUE DE LA DÉMOCRATIE LIBÉRALE, UNIVERSITÉ DU QUÉBEC À MONTRÉAL, Novembre 2007.
- [5] CLUSIF (Club de la Sécurité des systèmes d’Information Français.). « Techniques de contrôle d’accès par biométrie ». Juin 2003
- [6] John D. Woodward, Jr., Christopher Horn, Julius Gatune, and Aryn Thomas. “Biometrics, A Look at Facial Recognition”. **RAND Public Safety and Justice**. Documented Briefing prepared for the Virginia State Crime Commission.
- [7] Anis Chaari : Nouvelle approche d’identification dans les bases de données biométriques basée sur une classification non Supervisée, THESE de doctorat. Université d’Evry Val d’Essonne-France et Université de la Manouba, soutenue le 6 Octobre 2009, **HAL Open Science**.
- [8] Biometric: Types and its Applications. **International Journal of Science and Research** National Conference on Knowledge, Innovation in Technology and Engineering (NCKITE), 10-11 April 2015
- [9] Calais, La RFID, Identification par radio fréquence, 8 Novembre 2007.
- [10] Dat Son Nguyen : Développement des capteurs sans fils basés sur les tags RFID uhf passifs pour la détection de la qualité des aliments. THESE de doctorat dirigée par « Pr. Smaïl TEDJINI » et « Dr. HDR. Mau- DANG, UNIVERSITÉ DE GRENOBLE et UNIVERSITÉ NATIONALE DU VIETNAM, soutenue le 27 Septembre 2013. **HAL Open Science**.
- [11] Pierre GEORGET, Directeur Général de Gencod EAN France « L’identification par Radio Fréquence, Principes et applications » version 4.01, mars 2004 **EPCglobal**

- [12] Davinder P, Twinkle K, Preet K. “The RFID Technology and its applications”. International Journal of Electronics, Communication & Instrumentation Engineering Research and Development. September 2012
- [13] Mémoire de fin d’étude « Intégration d’une solution RFID dans la gestion de pointage Odoo » par Saber Zenagui K et Ouadah A, Université Abou Bakr Belkaid– Tlemcen 2016-2017
- [14] Iberraken S, Contrôle automatique de niveau d’eau, mémoire de fin de cycle, Université Abderrahmane Mira-Béjaia, 2019-2020.
- [15] Elprocus « Different Types of Arduino Boards », article disponible à l’adresse URL: <https://www.elprocus.com/different-types-of-arduino-boards/>
- [16] Tachnolab “Description de la carte Arduino UNO”, article disponible à l’adresse URL : <http://technolab.fr/wp-content/uploads/2017/10/Description-de-la-Carte-Arduino-UNO.pdf>
- [17] Rui Santos « 18 + Arduino Projects »
- [18] B. Cottenceau, « Carte Arduino UNO, Microcontrôleur ATmega328 ». ISTIA (École Polytechnique de L’Université d’Angers-France), 2016-2017
- [19] Omkar G, Prajwal Sp, « RFID Attendance using RC522 », Mai 2020.

Annexe

Test du module RFID, Led, Buzzer

```

#include <SPI.h>
#include <MFRC522.h>
#define SS_PIN 10
#define RST_PIN 9
MFRC522 mfrc522(SS_PIN, RST_PIN);
MFRC522::MIFARE_Key key;
const int greenLEDpin = 6;
const int buzzerPin = 5;
const int redLEDpin = 7;
void setup() {
  Serial.begin(9600);
  SPI.begin(); // Init SPI bus
  mfrc522.PCD_Init(); // Init RC522
  pinMode(redLEDpin, OUTPUT); //sets redLEDpin as OUTPUT
  pinMode(greenLEDpin, OUTPUT); //sets greenLEDpin as OUTPUT
  pinMode(buzzerPin, OUTPUT); //sets buzzerPin as OUTPUT
  Serial.println("Put your RFID tag close to the reader...");
  Serial.println();
}

void loop () {
  if ( ! mfrc522.PICC_IsNewCardPresent() )
    {return;}
  // Verify if the UID has been readed
  if ( ! mfrc522.PICC_ReadCardSerial() )
    {return;}
  //Show UID on serial monitor
  Serial.print("UID tag :");
  // Sound the buzzer when a card is read
  tone(buzzerPin, 2000);
  delay(100);
  noTone(buzzerPin);
  delay(100);
  String content="";
  byte letter;
  for (byte i = 0; i < mfrc522.uid.size; i++) {
    Serial.print(mfrc522.uid.uidByte[i] < 0x10 ? " 0" : " ");
    Serial.print(mfrc522.uid.uidByte[i], HEX);
    content.concat(String(mfrc522.uid.uidByte[i] < 0x10 ? " 0" : " "));
    content.concat(String(mfrc522.uid.uidByte[i], HEX));
  }
}

```

```

Serial.println();
Serial.print("Status : ");
content.toUpperCase();
if (content.substring(1) == "00 0D A3 7C")
{ Serial.print("You're on time");
  digitalWrite(greenLEDpin, HIGH);
  delay(500);
  digitalWrite(greenLEDpin, LOW);}
else
{ Serial.print("You are Late");
  digitalWrite(redLEDpin, HIGH);
  delay(500);
  digitalWrite(redLEDpin, LOW);}
}

```

TEST de la carte SD

```

#include <SPI.h>
#include <SD.h>
File myFile;
void setup() {
  Serial.begin(9600);
  while (!Serial) { ; }
  Serial.print("Initializing SD card...");
  if (!SD.begin(4)) {
    Serial.println("initialization failed!");
    while (1);}
  Serial.println("initialization done.");
  if (SD.exists("DATA1.txt")) {
    Serial.println("DATA1.txt exists.");
  } else {
    Serial.println("DATA1.txt doesn't exist.");
  }
  // open a new file and immediately close it:
  Serial.println("Creating DATA1.txt...");
  myFile = SD.open("DATA1.txt", FILE_WRITE);
  myFile.println("Arduino");
  myFile.println("SD");
  if (myFile) {
    while (myFile.available()) { //execute while file is available
      Serial.write(myFile.read());
      Serial.println("File exists");
    }
    myFile.close(); //close file

  } else
  { Serial.println("error opening DATA1.txt");}
}
void loop() {
  // nothing happens after setup finishes.
}

```

Test de l'écran LCD et horloge RTC DS1307

```

#include <Wire.h>
#include "RTClib.h"
#include <LiquidCrystal.h>
RTC_DS1307 RTC;
LiquidCrystal lcd (3, 2, A0, A1, A2, A3);
void setup () {
  Serial.begin(9600);
  lcd.begin(16, 2);
  Wire.begin();
  RTC.begin();
  if (! RTC.isrunning()) {
    Serial.println("RTC is NOT running!");
    RTC.adjust(DateTime(F(__DATE__), F(__TIME__)));
  }
}

void loop () {
  DateTime now = RTC.now();
  lcd.setCursor(0,0);
  lcd.print("TIME: ");
  lcd.print(now.hour(), DEC);
  lcd.print(":");
  lcd.print(now.minute(), DEC);
  lcd.print(":");
  lcd.print(now.second(), DEC);
  lcd.setCursor(0,1);
  lcd.print("DATE: ");
  lcd.print(now.year(), DEC);
  lcd.print("/");
  lcd.print(now.month(), DEC);
  lcd.print("/");
  lcd.print(now.day(), DEC);
  delay(1000);}

```

Programme du système de pointage à base de RFID sans la carte SD

```

#include <SPI.h> //include library code for SPI comms
#include <MFRC522.h> //include library code for RFID
#include <LiquidCrystal.h> //include library code for LCD1602
#include <RTClib.h> //include library code for the RTC
#include <Wire.h> //include library code for the RTC

```

Annexe

```
const int SS_PIN = 10;
const int RST_PIN = 9;
const int redLEDpin = 7;
const int greenLEDpin = 6;
const int buzzerPin = 5;
LiquidCrystal lcd(3, 2, A0, A1, A2, A3);
//create instance RTC
RTC_DS1307 rtc;
//create MFRC522 instance
MFRC522 mfrc522(SS_PIN, RST_PIN);
char daysOfTheWeek[7][12] = {"Su", "Mo", "Tu", "We", "Th", "Fr",
"Sa"};
// Define check in time
const int checkInHour = 20;
const int checkInMinute = 13;
//Variable to hold user check in
int userCheckInHour;
int userCheckInMinute;

void verifyCheckIn() {
    if ((userCheckInHour < checkInHour) || ((userCheckInHour ==
checkInHour) && (userCheckInMinute <= checkInMinute))) {
        digitalWrite(greenLEDpin, HIGH);
        delay(500);
        digitalWrite(greenLEDpin, LOW);
        lcd.setCursor(8, 0);
        lcd.print("on time");
        delay(5000);
        lcd.clear();}
    else {
        digitalWrite(redLEDpin, HIGH);
        delay(500);
        digitalWrite(redLEDpin, LOW);
```

Annexe

```
    lcd.setCursor(8, 0);
    lcd.print("late");
    delay(5000);
    lcd.clear();}}

void setup()
{
    Serial.begin(9600);    //initiate a serial communication
    lcd.begin(16,2);
    while(!Serial);
    SPI.begin();           //initiate SPI bus
    mfrc522.PCD_Init();    //initiate MFRC522µ

    pinMode(redLEDpin, OUTPUT);
    pinMode(greenLEDpin, OUTPUT);
    pinMode(buzzerPin, OUTPUT);
    // Setup for the RTC
    if (!rtc.begin()) {
        lcd.println("Couldn't find RTC");
        Serial.flush();
        while (1); delay(10);}
    if (!rtc.isrunning()) {
        lcd.println("RTC is NOT running!");
        rtc.adjust(DateTime(F(__DATE__), F(__TIME__)));}

    lcd.setCursor(2, 0);
    lcd.print( "Welcome to" );
    lcd.setCursor(0, 1);
    lcd.print("Control System");
    delay(2000);
    lcd.clear();}
```



```

void loop()
{
    lcd.setCursor(0, 0);
    lcd.print("Scan your card...");
    lcd.setCursor(0, 1);
    lcd.print("Card no:");
    lcd.setCursor(0, 2);
    lcd.print("Status:");
    lcd.setCursor(0, 1);
    DateTime now = rtc.now();
    lcd.print(now.day(), DEC);
    lcd.print ("/");
    lcd.print (now.month(), DEC);
    lcd.print ("/");
    lcd.print (now.year(), DEC);
    lcd.print (" ");
    lcd.print (now.hour(), DEC);
    lcd.print (":");
    if (now.minute() < 10) {
        lcd.print ("0");
    }
    lcd.print (now.minute(), DEC);
    lcd.print (":");
    if (now.second() < 10) {
        lcd.print ("0");
    }
    lcd.print (now.second(), DEC);
    lcd.print (" ");
    lcd.print (daysOfTheWeek[now.dayOfTheWeek()]);
    // Look for new cards
    if ( ! mfrc522.PICC_IsNewCardPresent())
    { return; }
    // Select one of the cards
    if ( ! mfrc522.PICC_ReadCardSerial())

```

```

{ return; }
lcd.print("UID tag:");
tone(buzzerPin, 2000);
delay(100);
noTone(buzzerPin);
delay(100);
String content = "";
byte letter;
for (byte i = 0; i < mfrc522.uid.size; i++)
{
    Serial.print(mfrc522.uid.uidByte[i] < 0x10 ? " 0" : " ");
    Serial.print(mfrc522.uid.uidByte[i], HEX);
    content.concat(String(mfrc522.uid.uidByte[i] < 0x10 ? "0":
" "));
    content.concat(String(mfrc522.uid.uidByte[i], HEX));
    lcd.clear();
    lcd.setCursor(0, 0);
    for (byte i = 0; i < mfrc522.uid.size; i++){
        lcd.print(mfrc522.uid.uidByte[i] < 0x10 ? "0" : " ");
        lcd.print(mfrc522.uid.uidByte[i], HEX);}
    }
    lcd.setCursor(0, 0);
    lcd.print("Status : ");
    content.toUpperCase();
    if (content.substring(1) == "")
    {
        lcd.setCursor(0,1);
        lcd.print(now.day(), DEC);
        lcd.print ('/');
        lcd.print (now.month(), DEC);
        lcd.print ('/');
        lcd.print (now.year(), DEC);
        lcd.print (' ');
    }

```

```

    lcd.print (now.hour(), DEC);
    lcd.print (':');
    lcd.print (now.minute(), DEC);
    lcd.print (':');
    lcd.print (now.second(), DEC);
    lcd.print (' ');
    lcd.println(daysOfTheWeek[now.dayOfTheWeek()]);
    // Save check in time;
    userCheckInHour = now.hour();
    userCheckInMinute = now.minute();
    verifyCheckIn();
    delay(2000);
}
else {
    lcd.setCursor(0,1);
    lcd.print(now.day(), DEC);
    lcd.print ('/');
    lcd.print (now.month(), DEC);
    lcd.print ('/');
    lcd.print (now.year(), DEC);
    lcd.print (' ');
    lcd.print (now.hour(), DEC);
    lcd.print (':');
    lcd.print (now.minute(), DEC);
    lcd.print (':');
    lcd.print (now.second(), DEC);
    lcd.print (' ');
    lcd.println(daysOfTheWeek[now.dayOfTheWeek()]);
    userCheckInHour = now.hour();
    userCheckInMinute = now.minute();
    verifyCheckIn();
    delay(2000); lcd.clear();}}

```