

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 : Automatique

Spécialité : Automatique et Informatique Industriel

Thème

Réalisation d'un système de pointage portable

Préparé par :

BOUKHALFA Ali

HAMDAOUI Laala

Dirigé par :

Mme Mezzah

Examiné par :

Mr Hanfoug

Mr Bessad

Année universitaire : 2022/2023

Remerciement

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

Je remercie mes parents, mes sœurs, 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 sincères respects et remerciements, à mon encadrante Mme Mezzah ; pour sa disponibilité et sa présence, son aide, sa gentillesse, ses compétences et son orientation.

Je remercie le propriétaire du magasin des composants électroniques qui nous a proposé une solution pour résoudre un problème dans notre travail

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

Merci

Remerciement

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

Je remercie ma mère, mes frères et sœurs, 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 sincères respects et remerciements, à mon encadrante Mme Mezzah ; pour sa disponibilité et sa présence, son aide, sa gentillesse, ses compétences et son orientation.

Je remercie le propriétaire du magasin des composants électroniques qui nous a proposé une solution pour résoudre un problème dans notre travail

Mes gratitudeux 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 sœurs ; des exemples, 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.

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

A mes amis, quin'ont jamais cessé de m'encourager, Que la joie soit leur allié et que le succès embellisse leur vie

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

Dédicace

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

Ma mère, aucune dédicace ne saurait exprimer mon respect et mon amour éternel que je leur porte, ma considération pour toutes ces années de sacrifices qu'elle a consentie 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 frères et sœurs ; des exemples, 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 mon père, paix à leur âme ; le fruit de leur conseils précieux, a qui m'a éclairé le chemin et m'a montré la voix. Puisse Allah le toutpuissant, l'avoir en sa sainte miséricorde.

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.

A mes amis, qui n'ont jamais cessé de m'encourager, Que la joie soit leur allié et que le succès embellisse leur vie

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

TABLE DES MATIERE

LISTE DES FIGURES***LISTE DES TABLEAUX******LISTE DES ABREVIATION*****INTRODUCTION GNERALE**

1. Introduction	2
-----------------------	---

CHAPITRE I : SYSTEMES DE GESTION DE PRESENCE

1.1. Introduction	4
1.2. Fonctionnalités d'un système de présence	5
1.3. Architecture d'un système de présence	5
1.4. Types des systèmes de présence.....	6
1.4.1. Système de présence basé sur un code-barres	6
1.4.2. Système de présence biométrique	8
1.4.3. Système de présence intelligent (ou système d'IA)	12
1.4.4. Système de présence basé sur RFID.....	13
1.4.4.1. Composants fondamentaux de la technologie RFID.....	14
1.4.4.2. Architecture et fonctionnement de la technologie RFID	16
1.4.4.3 domaines d'application de la technologie RFID	17
1.5. Conclusion.....	18

CHAPITRE II : SYSTEME DE POINTAGE BASE SUR LA RFID

2.1. Introduction	20
2.2. Désignation du système de présence RFID portable proposé	21
2.3. Outils de développement matériel et logiciel.....	21
2.3.1. Outils de développement matériel.....	22
2.3.1.1 Carte Arduino UNO R3.....	22
2.3.1.2 Les étiquettes RFID	23
2.3.1.3 Le Module RFID... ..	24
2.3.1.4 Le Module Data Logger Shields	26
2.3.1.5 L'afficheur LCD I2C 16x2.....	27
2.3.2. Outils de développement logiciel.....	28
2.3.2.1 Arduino IDE.....	28

2.3.2.2	Le langage de programmation	30
2.3.2.3	La communication série	33
2.3.2.4	Les bibliothèques.....	34
2.4	Conclusion.....	42

CHAPITRE III : IMPLEMENTATION ET VALIDATION

3.1.	Introduction	44
3.2.	Implémentation matérielle.....	44
3.3.	Implémentation logiciel.....	52
3.4.	Teste et validation	66
3.5.	Conclusion.....	71

CONCLUSION GENERALE	73
----------------------------------	-----------

REFERENCES

LISTE DES FIGURES

Figure 1.1: code-barres	6
Figure 1.2: lecteur code-barres	6
Figure 1.3: Reconnaissance d’empreinte digitale	7
Figure 1.4: Identification par géométrie de la main.....	8
Figure 1.5: Identification biométrique par IRIS	9
Figure1.6 : Reconnaissance faciale	9
Figure1.7 : Reconnaissance vocale.....	10
Figure 1.8: Reconnaissance par la signature	10
Figure 1.9 : Système de présence intelligent (Reconnaissance facial)	11
Figure 1.10 : L’IA dans le travail.....	12
Figure 1.11 : Système de présence basé sur RFID	12
Figure 1.12 : Composants fondamentaux de la technologie RFID.....	13
Figure 1.13: Composant du Tag RFID	13
Figure 1.14: Différents supports de tags RFID	14
Figure 1.15 : Différents lecteurs RFID	15
Figure 1.16: Communication entre Lecteur et Tag RFID	16
Figure 1.17: Domaines d’application de la technologie RFID	17
Figure 2.1: Carte Arduino UNO R3	22
Figure 2.2: Les étiquettes RFID	24
Figure 2.3 : Module RFID-RC522	24
Figure 2.4 : Le Module data logger Shields	26
Figure 2.5 : L’afficheur LCD I2C 16x2	27
Figure 2.6 : Logiciel Arduino IDE	29
Figure 2.7 : Les bibliothèques du logiciel Arduino IDE	29

Figure 3.1 : Schéma synoptique du projet	45
Figure 3.2 : Module data logger Shields et la carte Arduino UNO.....	46
Figure 3.3 : Montage du Shields sur l'Arduino	46
Figure 3.4 : Shield avec l'Arduino.....	46
Figure 3.5 : Schéma électronique RFID et Arduino	47
Figure 3.6 : Câblage du module RFID-RC522 avec l'Arduino	48
Figure 3.7 : Schéma électrique LCD I2C avec Arduino	49
Figure 3.8 : Câblage du l'afficheur avec la carte Arduino.....	49
Figure 3.9 : Schéma, boutons avec Arduino	50
Figure 3.10 : câblage des boutons avec l'Arduino.....	50
Figure 3.11 : Alimentation avec batterie 9V	50
Figure 3.12 : Schéma électrique globale(fritzing)	51
Figure 3.13 : Le circuit globale du system de pointage basé sur RFID.....	51
Figure 3.14 : Teste de module RFID-RC522	52
Figure 3.15 : Teste de lecteur carte SD et l'RTC-DS1307	53
Figure 3.16 : Teste de l'LCD et les boutons.....	54
Figure 3.17 : Organigramme de fonctionnement générale.....	55
Figure 3.18 : fonctionnement de 'scancart'	60
Figure 3.19 : fonctionnement de 'listuid'	64
Figure 3.20 : Affichage de notre système.....	68
Figure 3.21 : carte SD n'est pas initialisé.....	68
Figure 3.22 : carte SD est initialisé	68
Figure 3.23 : Scanne du l'UID	69
Figure 3.24 : fichier 'listuid.txt'	69
Figure 3.25 : Affichage de fonction scan carte	70
Figure 3.26 : fichier liste de présence.....	70
Figure 3.27 : fichiers criés dans la carte SD.....	70

LISTE DE TABLEAUX

Tableau 2.1 : Les pins et leur description pour la carte Arduino UNO.....	23
Tableau 2.2 : Les broches et leur description pour un module RFID	25
Tableau 2.3 : Fonctions utiliser dans le langage de programmation Arduino-IDE	30
Tableau 2.4 : Variables utilisées dans le langage de programmation Arduino-IDE	31
Tableau 2.5 : Structure utilisées dans le langage de programmation Arduino-IDE.....	32
Tableau 3.1 : Liste des composants utilisés	39
Tableau 3.2 : Connexion entre les broches Arduino et RFID.....	47
Tableau 3.3 : Connexion entre les broches Arduino et LCD.....	48
Tableau 3.4 : Connexion entre les broches Arduino et les boutons	50

LISTE DES ABREVIATIONS

IA : Intelligence Artificielle

RQ : Quick Responce

ADN : Acide DésoxyriboNucléique

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

RFID : Radio Frequency Identification

SD (carte) : Secure Digital

RTC : Real-Time Control

SPI : Serial Peripheral Interface

USB : Universal Serial Bus

LCD : Liquid cristal display / Ecran à cristaux liquides.

I2C : Inter Integrated Circuit Bus

SCL : Serial Clock

SDA : Serial Data

MOSI : Multiple Output, Signal Input

MISO : Multiple Input, Signal Output

RST : Réinitialisation

INTRODUCTION GENERALE

Introduction

La présence des étudiants est un élément crucial pour les établissements d'enseignement, car elle garantit leur participation et leur engagement. De plus, la présence est souvent utilisée pour évaluer leur participation aux examens, évaluations et travaux de cours.

Dans le cadre de notre projet, nous développons un système portable de prise de présence basé sur la technologie RFID (Radio Frequency Identification) afin d'automatiser le processus fastidieux et chronophage de prise de présence manuelle des étudiants. En intégrant des étiquettes RFID dans leurs cartes d'identité, il sera possible enregistrer rapidement et facilement leur présence en les faisant simplement passer leur carte sur un lecteur RFID. Les données de présence peuvent ensuite être gérées efficacement. De plus, ce système portable permet de suivre la présence des étudiants lors de différentes activités et facilite la gestion administrative.

La technologie RFID offre une solution pratique et efficace pour la prise de présence des étudiants. Le système que nous proposons permettra de collecter et de stocker les données de présence des étudiants. Il utilise une carte Arduino et un ensemble de modules pour créer et sauvegarder les fichiers d'archives, ainsi qu'une interface utilisateur pour l'exploitation et la configuration du système.

Ce mémoire est divisé en trois chapitres. Le premier présente les notions générales sur les systèmes de pointage et les différentes technologies utilisées. Nous expliquons en détail le principe de l'identification par radiofréquence et les composants des systèmes RFID. Le deuxième chapitre décrit la structure et les fonctionnalités du système de pointage que nous proposons. Enfin, le troisième chapitre présente les étapes de réalisation et de validation du système.

CHAPITRE 1

SYSTEMES DE GESTION

DE PRESENCE

1.1 Introduction

La gestion de la présence des étudiants est un aspect important de la gestion des établissements d'enseignement. Elle peut être utilisée pour suivre l'assiduité des étudiants, améliorer la qualité de l'enseignement et la performance des enseignants, ainsi que pour assurer la sécurité des campus. La technologie a permis de développer des systèmes de présence efficaces et automatisés pour les établissements d'enseignement, offrant une solution pour suivre et gérer la présence des étudiants. Dans ce chapitre, nous allons examiner les fonctionnalités et l'architecture générale d'un système de présence, ainsi que les différents types de systèmes de présence disponibles sur le marché, allant des systèmes de présence à base de code-barres aux systèmes de présence biométriques et intelligents.

Dans la première section, nous allons passer en revue les fonctionnalités de base d'un système de présence, qui comprend la collecte des données de présence, la gestion des absences, la génération de rapports de présence et la communication avec les étudiants et les enseignants.

La deuxième section sera consacrée à l'architecture d'un système de présence, qui comprendra la description des différents modules et composants qui composent le système, ainsi que leur interconnexion.

Dans la troisième section, nous allons passer en revue les différents types de systèmes de présence disponibles sur le marché. Nous examinerons les avantages et les inconvénients de chaque type de système, allant des systèmes de présence à base de code-barres aux systèmes de présence biométriques et intelligents.

1.2 Fonctionnalités d'un système de présence

Les fonctionnalités d'un système de présence pour les étudiants peuvent varier en fonction des besoins spécifiques de l'établissement, mais en général, elles incluent :

Suivi des heures de présence : la fonction principale d'un système de présence est de suivre les heures de présence des étudiants. Cela permet de savoir quand les étudiants sont arrivés en cours, combien de temps ils ont assisté et quand ils sont partis.

Gestion des absences : un système de présence peut également aider à gérer les absences des étudiants, qu'il s'agisse d'absences justifiées ou non. Les données recueillies peuvent être utilisées pour planifier les cours et les examens, ainsi que pour établir des rapports sur l'assiduité.

Gestion des horaires : un système de présence peut faciliter la gestion des horaires en permettant aux étudiants de consulter et de gérer leur emploi du temps en ligne. Les responsables peuvent également planifier les cours en conséquence.

Rapports et analyses : les données collectées par un système de présence peuvent être utilisées pour générer des rapports et des analyses sur l'assiduité des étudiants, les performances académiques, etc. Ces informations peuvent aider les établissements à prendre des décisions éclairées en matière de gestion des étudiants.

Sécurité : un système de présence peut également contribuer à améliorer la sécurité sur le campus en identifiant rapidement les étudiants présents en cas d'urgence.[1]

1.3 Architecture d'un système de présence

L'architecture d'un système de présence peut varier en fonction des besoins et des technologies disponibles, mais elle comprend généralement trois composants principaux :

Les périphériques de saisie de présence : ces périphériques permettent de collecter les données de présence, tels que les lecteurs de cartes d'identification, les scanners de codes-barres, les lecteurs de codes QR, les capteurs de proximité, les caméras de reconnaissance faciale.

La base de données : les données de présence collectées par les périphériques sont stockées dans une base de données centrale. Cette base de données peut être hébergée

localement ou dans le cloud, et peut être gérée par l'établissement d'enseignement ou un fournisseur de services tiers.

L'interface utilisateur : l'interface utilisateur permet aux utilisateurs du système de présence de visualiser et d'interagir avec les données de présence. Cette interface peut être un logiciel installé sur un ordinateur ou une application mobile, et permet aux étudiants, aux enseignants et aux responsables de l'établissement de consulter les rapports d'assiduité, de saisir les absences, de gérer les horaires...

Ces différents modules et composants sont interconnectés pour former un système de présence fonctionnel. Le module de capture collecte les données de présence, qui sont ensuite stockées dans le module de stockage. Ensuite ces données peuvent être accessibles via l'interface utilisateur. [2]

1.4 Types de systèmes de présence

Les systèmes de présence sont utilisés dans de nombreux établissements d'enseignement pour suivre la présence des étudiants et améliorer la gestion des ressources humaines. Il existe plusieurs types de systèmes de présence, chacun ayant ses avantages et ses inconvénients. Voici une liste des types de systèmes de présence les plus courants

1.4.1 Système de présence à base de code-barres

Un code-barres est une représentation lisible utilisée 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 : [3]

- ✓ 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.

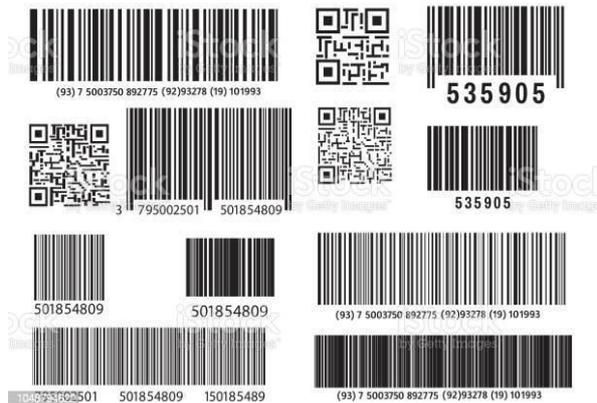


Figure1.1 : code-barres



Figure1.2 : lecteur code-barres [35]

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. 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].

Le système de présence basé sur les codes-barres est relativement peu coûteux et facile à mettre en place. Il peut également être utilisé pour automatiser le processus de suivi de la présence, en générant des rapports de présence en temps réel. Cependant, ce système de présence est sensible aux erreurs de saisie manuelle, telles que la mauvaise lecture du code-barres ou l'utilisation de codes-barres incorrects. De plus, les codes-barres peuvent être facilement reproduits ou dupliqués, ce qui peut entraîner des fraudes ou des erreurs dans la gestion de la présence. [4]

1.4.2 Système de présence 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 » [5]

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.

Types courants de la biométrie

- 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.



Figure1.3 : Reconnaissance d'empreinte digitale [35]

- 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.

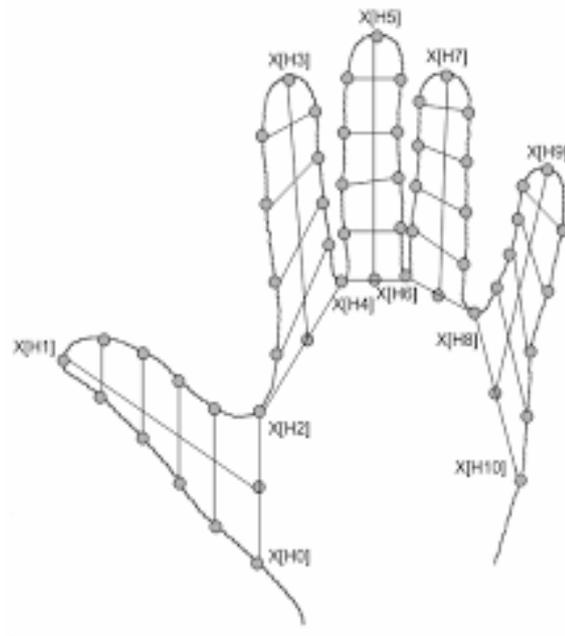


Figure 1.4 : Identification par géométrie de la main [35]

L'image scannée est convertie en un modèle numérique et associée 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.

- Identification biométrique par IRIS

Ce système d'authentification biométrique utilise des scanners spécialisés et des caméras infrarouges pour capturer une image précise de l'iris. Les étapes comprennent la localisation de la pupille, la détection de l'iris et des paupières, l'exclusion des paupières et des cils, et la conversion de l'iris en blocs de données numériques représentant l'image.



Figure 1.5 : Identification biométrique par IRIS [35]

Pour vérifier l'identité, les données collectées sont comparées en utilisant les mêmes méthodes de correspondance. Ce système est considéré comme extrêmement fiable et précis en raison de sa capacité à capturer des détails minutieux de l'iris et à effectuer des comparaisons précises avec les enregistrements biométriques existants.[6]

- **Reconnaissance faciale**

Les méthodes de reconnaissance faciale se basent sur l'emplacement et la forme des différentes caractéristiques du visage, telles que les yeux, les sourcils, le nez, les lèvres et le menton. Un module d'acquisition d'images, équipé d'une caméra, est utilisé pour confirmer l'identité d'une personne en se basant sur les données disponibles. L'utilisateur doit simplement se positionner devant l'appareil, puis une détection automatique du visage est réalisée et l'image est ensuite envoyée à un logiciel pour être numérisée.

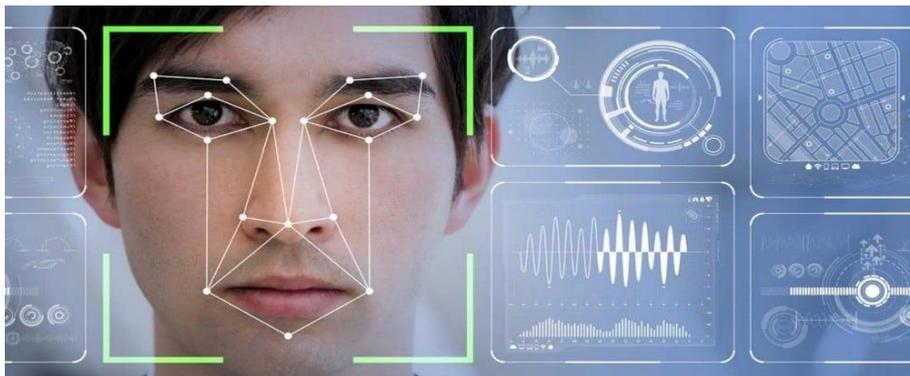


Figure1.6 : Reconnaissance faciale [35]

- **Reconnaissance vocale**

La reconnaissance vocale est une méthode d'authentification biométrique qui se base sur les caractéristiques vocales pour identifier les individus. Elle fait appel à des systèmes automatisés de reconnaissance vocale.



Figure 1.7 : Reconnaissance vocale [35]

- **Reconnaissance par la signature**

La reconnaissance par signature est un processus utilisé pour identifier et vérifier l'identité d'une personne en analysant sa signature manuscrite. Chaque individu a une signature unique qui peut être utilisée comme une caractéristique distinctive pour l'authentification.



Figure 1.8 : Reconnaissance par la signature [35]

Le système de présence biométrique présente de nombreux avantages par rapport aux systèmes de présence manuels ou basés sur les codes-barres. Tout d'abord, il offre une sécurité accrue, car les caractéristiques biométriques sont uniques à chaque individu et difficiles à falsifier ou à copier. De plus, il est rapide et facile à utiliser, car il ne nécessite pas de carte d'identification ou de badge. Enfin, il permet une collecte de données précise et automatisée, réduisant ainsi les risques d'erreurs humaines. [7]

Cependant, le système de présence biométrique peut être coûteux à mettre en place, car il nécessite des investissements en termes de matériel et de logiciels. De plus, certains employés ou étudiants peuvent être réticents à utiliser cette technologie en raison de préoccupations de confidentialité.

1.4.3 Système de présence intelligent (ou système d'IA)

Un système de présence avec IA fait référence à un système qui utilise l'intelligence artificielle (IA) pour gérer la présence des individus. Ce type de système utilise des techniques d'apprentissage automatique et de reconnaissance pour suivre et identifier la présence des personnes dans un environnement donné.

L'IA peut être utilisée pour analyser des données provenant de capteurs tels que des caméras, des capteurs de mouvement ou des dispositifs de suivi biométrique. L'algorithme d'IA peut apprendre à reconnaître les personnes spécifiques en se basant sur des caractéristiques uniques telles que la forme du visage, les empreintes digitales, les caractéristiques vocales, etc.



Figure 1.9 : Système de présence intelligent (Reconnaissance faciale)

Le système de présence avec IA peut être utilisé dans divers domaines, tels que la gestion du personnel dans les entreprises, le suivi de la présence des élèves dans les écoles, le contrôle d'accès sécurisé, etc. L'IA permet d'améliorer la précision de la reconnaissance, d'automatiser les processus de suivi de présence et de fournir des données en temps réel pour une gestion plus efficace. [6]



Figure 1.10 : L'IA dans le travail [35]

1.4.4 Système de présence basé sur RFID

Un système de présence basé sur la technologie RFID (Radio Frequency Identification) utilise des étiquettes RFID pour suivre la présence des étudiants. Chaque étudiant reçoit une étiquette RFID qui est lue à l'entrée de la salle de classe par un lecteur RFID. Les données de présence sont ensuite enregistrées dans une base de données centrale pour permettre l'analyse ultérieure. Ce type de système est plus précis que les systèmes à base de code-barres et moins sujet aux erreurs de saisie, mais il peut être plus coûteux à mettre en place.



Figure1.11 : Système de présence basé sur RFID [35]

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. [8]

1.4.4.1 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)

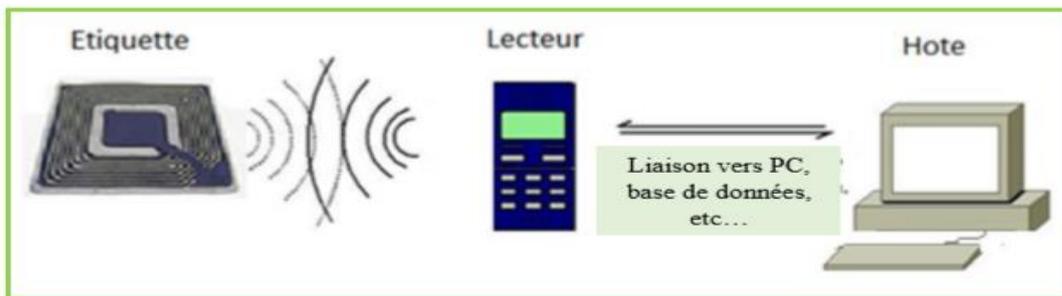


Figure1.12 : Composants fondamentaux de la technologie RFID [35]

a) 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. [9]

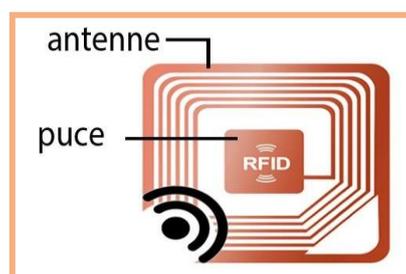


Figure 1.13: Composant du Tag RFID [18]

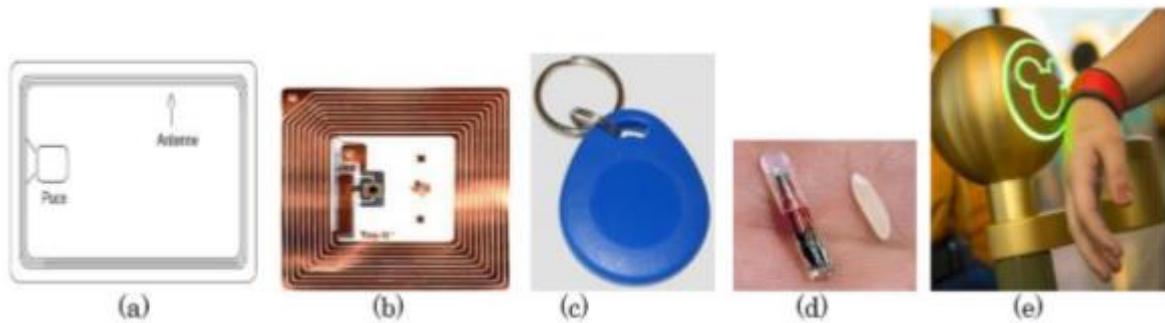


Figure 1.14: Différents supports de tags RFID [9]

Il existe deux grandes catégories de Tags RFID

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.

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]

b) Lecteur/Interrogeur

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. [10]

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 1.15: Différents lecteurs RFID [10]

1.4.4.2 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 (interrogateurs) 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. [11]

Le tag RFID est équipé 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 place leur 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.

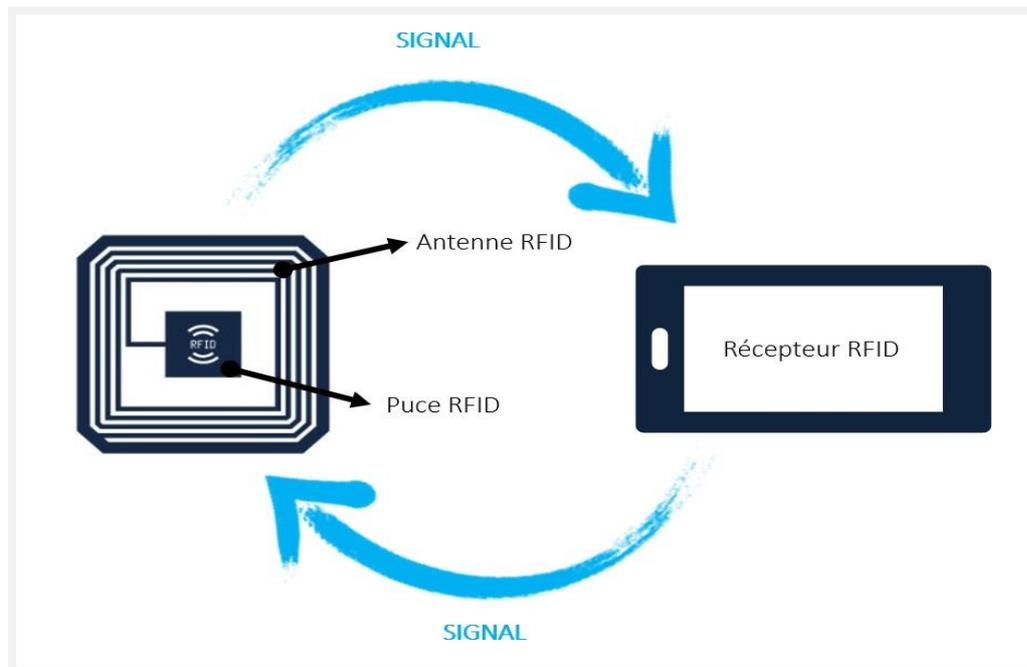


Figure 1.16: Communication entre Lecteur et Tag RFID [11]

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.[11]

1.4.4.3 Domaines d'application de la technologie RFID

Les systèmes d'identification par radiofréquence (RFID) sont utilisés dans de nombreuses applications essentielles de la vie quotidienne, notamment :

- ✓ Dans les établissements d'enseignement
- ✓ 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...

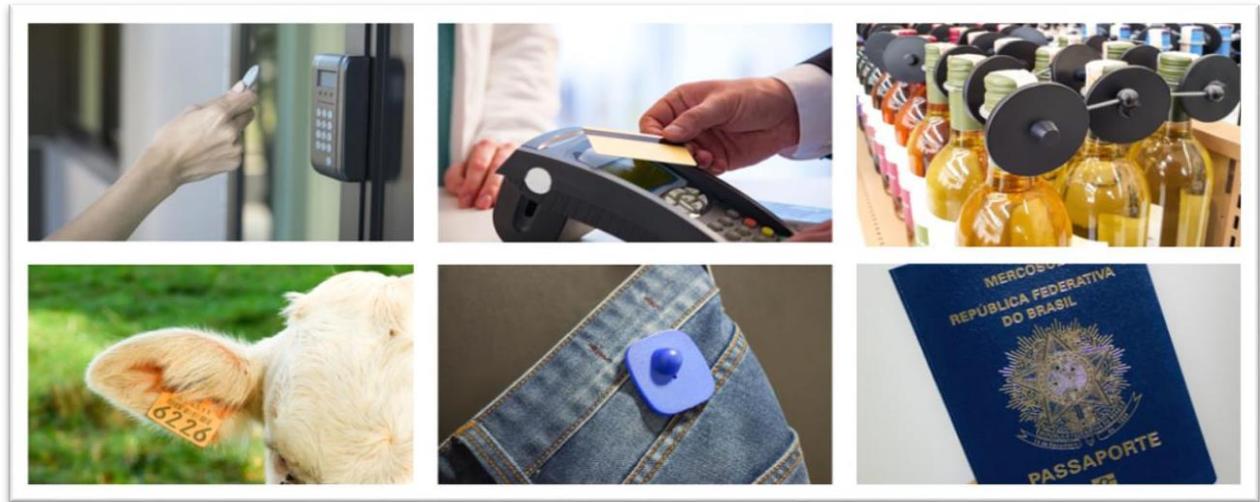


Figure1.17 : Domaines d'application de la technologie RFID [35]

1.5 Conclusion

En conclusion, ce chapitre a permis de présenter les différentes fonctionnalités et architectures des systèmes de présence ainsi que les différents types de systèmes utilisés dans les établissements d'enseignement. Nous avons vu que ces systèmes peuvent être basés sur des technologies telles que les codes-barres, l'identification par radiofréquence, la biométrie et l'intelligence artificielle. Chaque type de système a ses avantages et ses inconvénients, et le choix du système le plus approprié dépendra des besoins spécifiques de chaque établissement.

Nous avons également vu que les systèmes de présence ont un impact significatif sur la gestion des ressources humaines et la planification des activités d'enseignement, en permettant une collecte efficace des données de présence des étudiants. Cependant, il est important de prendre en compte les préoccupations liées à la vie privée et à la sécurité des données des étudiants, en particulier dans le cas de systèmes de présence biométriques et intelligents.

Dans le prochain chapitre, nous nous concentrerons sur les différentes méthodes de collecte de données de présence utilisées dans les systèmes de présence, en examinant les avantages et les inconvénients de chaque méthode. Nous verrons également comment ces données peuvent être utilisées pour améliorer la gestion des ressources humaines et optimiser les activités d'enseignement dans les établissements d'enseignement.

CHAPITRE 2

SYSTÈME DE PRÉSENCE

RFID PORTABLE

2.1 Introduction

Les dispositifs d'identification par radiofréquence (RFID) sont des équipements utilisés pour détecter les ondes radio. La technologie RFID est conçue pour l'identification automatisée d'objets. Les étiquettes RFID sont fixées sur des produits ou des appareils électroniques individuels, et lorsqu'une demande de transfert est effectuée, elles fournissent des informations sur le produit, comme ses caractéristiques.

Les étiquettes RFID sont des petits appareils électroniques qui comportent une puce et une antenne. Pour clarifier, on peut dire que les dispositifs RFID fonctionnent comme des codes à barres ou des bandes magnétiques sur les cartes de crédit. Pour chaque objet, le RFID crée une caractéristique unique qui peut être identifiée par d'autres appareils. Les informations peuvent être lues à partir des codes à barres ou des bandes magnétiques, tout comme RFID peut être lu par des lecteurs et les informations peuvent être reçues ou modifiées via eux. Ces dernières années, les systèmes d'identification automatique ont été couramment utilisés dans les industries, les professions et les autres organisations pour collecter des informations sur les personnes, les animaux, les marchandises et les produits en transit. La technologie d'identification par ondes radioélectriques est une technologie récente qui révolutionne la productivité des entreprises [12] [13].

Les lecteurs de tags RFID peuvent scanner rapidement des milliers de mètres par seconde, et le produit est détecté lorsqu'il est déplacé d'un endroit à un autre. Les applications de la technologie RFID incluent la réforme commerciale des intermédiaires, le soutien de la chaîne de gestion, la liste d'inventaire des produits, la régénération des animaux dans des cas rares, etc. Une question controversée sur l'utilisation des tags RFID concerne leur utilisation pour l'identification humaine, en particulier dans les systèmes de présence des professeurs et des étudiants.

2.2 Désignation du système de présence RFID portable proposé :

Le système portable RFID proposé pour la gestion des présences est une solution qui permet de suivre la présence des étudiants dans les salles de cours à l'aide de la technologie RFID (Radio Frequency Identification). Il s'agit d'un système portable, facilement transportable d'une salle de classe à l'autre, qui utilise des lecteurs RFID pour scanner les étiquettes RFID portées par les étudiants.

Le système se compose de plusieurs éléments : les étiquettes RFID portées par les étudiants, les lecteurs RFID, le logiciel de gestion des présences et l'interface utilisateur. Les étiquettes RFID sont attachées aux cartes étudiantes ou aux badges portés par les étudiants et contiennent des informations telles que le nom et le numéro d'étudiant. Les lecteurs RFID sont disposés dans chaque salle de classe et communiquent avec les étiquettes RFID pour enregistrer la présence des étudiants.

Le logiciel de gestion des présences permet de stocker les données de présence des étudiants dans une base de données centrale, de générer des rapports de présence pour chaque cours, de gérer les absences et les retards et d'envoyer des notifications aux enseignants ou aux étudiants en cas de problèmes. L'interface utilisateur est conviviale et permet aux enseignants et aux étudiants d'accéder facilement aux informations de présence [14].

Le système portable RFID pour la gestion des présences offre de nombreux avantages par rapport aux méthodes de suivi de présence traditionnelles, telles que les feuilles de présence papier ou les systèmes de codes-barres. Il est plus précis, plus rapide et plus efficace, ce qui permet aux enseignants de passer plus de temps à enseigner et aux étudiants de mieux suivre leur présence. De plus, le système peut être facilement transporté d'une salle de classe à l'autre, ce qui le rend très pratique pour une utilisation dans différents contextes.

2.3 Outils de développement matériel et logiciel :

Les outils matériels et logiciels utilisés dans le développement du système de présence RFID portable proposé sont essentiels pour le succès du système. Dans cette section, nous discuterons des outils qui ont été utilisés dans le processus de développement.

2.3.1 Outils de développement matériel

2.3.1.1 La carte Arduino UNO R3

Arduino est une plateforme open-source de développement électronique basée sur du matériel et un logiciel simple et facile à utiliser. Les cartes Arduino sont équipées de microcontrôleurs programmables qui permettent de contrôler des composants électroniques et de créer des projets interactifs. La plus simple à utiliser et la meilleure pour l'apprentissage est l'Arduino UNO R3. [15][16].

L'Arduino UNO R3 est une carte électronique programmable basée sur un microcontrôleur ATmega328P, qui offre des entrées/sorties numériques et analogiques pour interagir avec différents capteurs, actionneurs et autres périphériques électroniques. Elle est conçue pour être facile à utiliser et à programmer, même pour les débutants en électronique et en informatique. Elle peut être programmée en utilisant l'environnement de développement intégré (IDE) d'Arduino. La carte UNO R3 est largement utilisée dans l'enseignement, la recherche, les projets de bricolage et les projets d'Internet des objets. [17]

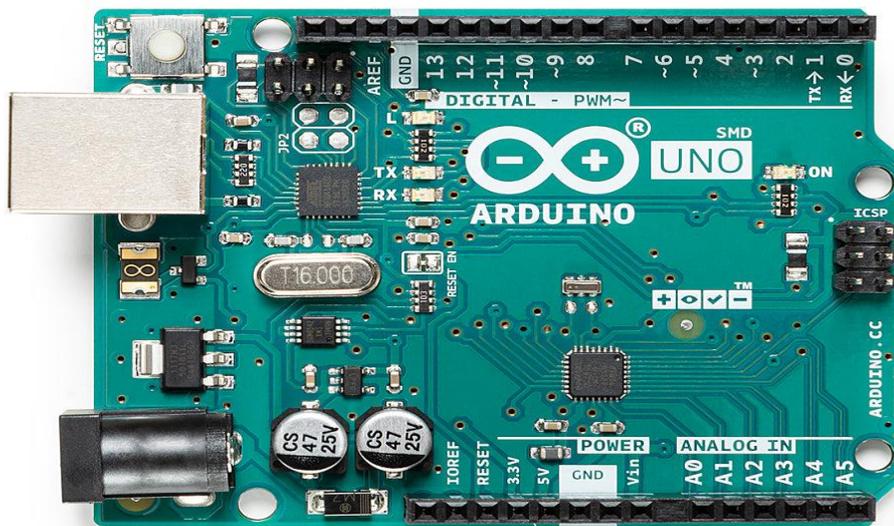


Figure2.1 : Carte Arduino UNO R3 [17]

Le microcontrôleur exécute le programme stocké dans sa mémoire interne, ce qui permet à la carte Arduino de contrôler des composants électroniques tels que des LED, des moteurs, des capteurs et d'autres dispositifs. Les programmes peuvent être développés pour effectuer une grande variété de tâches, allant de la simple commande de LED à la surveillance de capteurs et à la communication avec d'autres appareils via des protocoles tels que le Wi-Fi et le Bluetooth.

La carte dispose d'un bouton de réinitialisation, d'un résonateur céramique de 16 MHz (CSTCE16M0V53-R0), peut être alimentée à partir du port USB de votre ordinateur, de la plupart des chargeurs USB ou d'un adaptateur secteur (9 volts recommandés, pointe de fût de 2,1 mm, centre positif). Si aucun bloc d'alimentation n'est branché sur la prise d'alimentation, l'alimentation proviendra de la carte USB, mais dès que vous branchez une alimentation, la carte l'utilisera automatiquement. [15]

Pin	Description
0,1,2,4,7,8,12,13	Pin d'entrée/sortie numérique
3	Pin d'entrée/sortie numérique/ PWM (modulation de largeur d'impulsion)
5,6,9,10,11,	Pin d'entrée/sortie numérique/ PWM
A0, A1, A2, A3	Pin d'entrée analogique
A4	Pin d'entrée analogique/ I2C (Inter-Integrated Circuit)
A5	Pin d'entrée analogique/ I2C
3.3V, 5V	Pin de sortie de tension régulée
GND	Pin de terre
VIN	Tension d'entrée de la carte Arduino UNO

Tableau2.1 : Les pins et leur description pour la carte Arduino UNO

Les pins 0 à 13 sont des pins numériques qui peuvent être configurés comme des entrées ou des sorties, tandis que les pins A0 à A5 sont des entrées analogiques pour la lecture de signaux de capteurs ou autres composants électroniques. Les pins marquées PWM sont capables de générer des signaux PWM pour contrôler la vitesse des moteurs, la luminosité des LED, etc.

2.3.1.2 Les étiquettes RFID

L'une des méthodes d'identification les plus couramment utilisées consiste à intégrer un numéro de série ou une suite de données dans une puce électronique et à la relier à une petite antenne. Ce couple (puce + antenne) est ensuite encapsulé dans un support appelé RFID Tag ou RFID Label. Ces "tags" peuvent être intégrés dans des objets ou collés sur des produits, et sont souvent imprimés sur un support pliable et adhésif. Les données inscrites sur les étiquettes sont normalisées selon les directives d'EPC Global (Electronic Product Code). [18]



Figure2.2 : Les étiquettes RFID

2.3.1.3 Le module RFID

Le module RFID RC522 est un lecteur de cartes RFID basé sur le circuit intégré MFRC522 de NXP Semi-conducteurs. Il est souvent utilisé dans les projets électroniques pour lire et écrire des données sur des cartes RFID.



Figure2.3 : Module RFID-RC522

Le fonctionnement du module RFID RC522 est relativement simple. Ce dernier est composé d'un circuit émettant une énergie électromagnétique via une antenne et d'un système électronique qui reçoit et décode les informations envoyées par la puce RFID, puis les transmet au dispositif de collecte des données. En plus de lire les étiquettes RFID, ce dispositif est

capable d'écrire leur contenu. Le lecteur RFID joue un rôle crucial dans la lecture des étiquettes à radiofréquence et la transmission des informations qu'elles contiennent (telles que le code EPC, les informations d'état ou les clés de chiffrement) au niveau suivant du système (le middleware). Cette communication entre le lecteur et l'étiquette s'effectue en quatre étapes : premièrement, le lecteur transmet l'énergie nécessaire pour activer l'étiquette ; deuxièmement, il envoie une requête pour interroger les étiquettes à proximité ; troisièmement, il écoute les réponses et élimine les doublons ou les collisions entre réponses ; enfin, il transmet les résultats obtenus aux applications concernées. [18]

Le module RC522 utilise l'interface de communication SPI pour communiquer avec le microcontrôleur ou l'ordinateur auquel il est connecté. Il prend en charge les protocoles de communication ISO/IEC 14443A/MIFARE, ce qui permet la lecture et l'écriture de cartes MIFARE Classic, MIFARE Ultralight et d'autres cartes RFID compatibles avec ce protocole.[19]

Voici un tableau donnant les broches et leur description pour un module RFID :

Pin	Description
SDA	Données série
SCK	Horloge série
MOSI	Maître en esclave sortie
MISO	Maître en esclave entrée
IRQ	Interruption, non utilisée
GND	Masse
RST	Réinitialisation, actif bas
3.3V	Alimentation (5V)

Tableau2.2 : Les broches et leur description pour un module RFID

2.3.1.4 module data logger Shields

Le module Data Logger Shield est un composant électronique qui se connecte directement sur une carte Arduino et permet d'ajouter des fonctions d'enregistrement automatique de données. Il intègre une carte mémoire SD pour stocker les données collectées, une horloge temps réel pour marquer les enregistrements avec une précision temporelle et une surface libre

de prototypage pour ajouter d'autres composants électroniques pour réaliser des projets plus avancés.

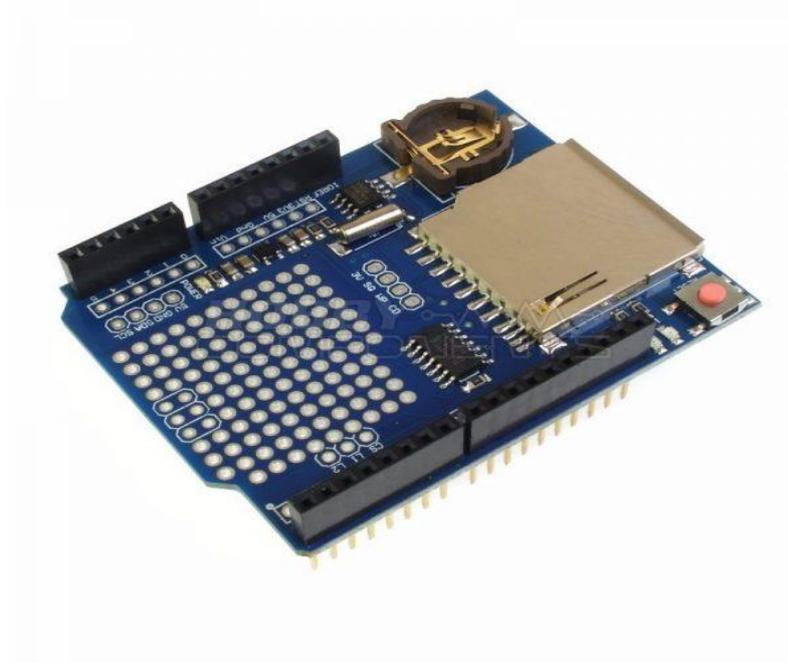


Figure 2.4 : Le Module data logger Shields [20]

Une fois connecté à une carte Arduino, le module Data Logger Shield permet d'enregistrer les données collectées à des intervalles réguliers sur une carte mémoire SD qui peuvent ensuite être relus et analysés sur un ordinateur à l'aide de logiciels d'analyse ou de tableurs. Le module dispose également d'une horloge en temps réel (RTC) à quartz avec une pile intégrée qui conserve l'heure hors tension, permettant une horodatage précis des enregistrements.

Le module Data Logger Shield se branche directement sur une carte Arduino et dispose de broches d'entrée analogiques et numériques permettant de connecter différents capteurs ou périphériques d'acquisition de données. Une fois que les capteurs sont connectés et que les paramètres de collecte de données ont été configurés à l'aide d'un programme Arduino, le module commence à collecter automatiquement des données et à les stocker sur la carte mémoire SD à des intervalles réguliers.[20]

Le module utilise la bibliothèque de gestion de fichiers SD pour stocker les données sur la carte mémoire SD. Cette bibliothèque permet de créer, lire et écrire des fichiers sur la carte mémoire SD à l'aide de fonctions standard Arduino.

2.3.1.5 L'afficheur LCD I2C 16x2

Un afficheur LCD I2C (Inter-Integrated Circuit) est un dispositif d'affichage à cristaux liquides (LCD) qui utilise une interface de communication série I2C pour communiquer avec un microcontrôleur ou un autre dispositif.



Figure 2.5 : L'afficheur LCD I2C 16x2 [21]

L'I2C est un protocole de communication série bidirectionnel qui permet aux périphériques de communiquer entre eux à l'aide de seulement deux fils de données, SDA (Serial Data) et SCL (Serial Clock). L'afficheur LCD I2C utilise cette interface pour transmettre les données d'affichage au microcontrôleur et pour recevoir les commandes de contrôle de l'affichage.

Le fonctionnement d'un afficheur LCD I2C peut varier en fonction du modèle et du fabricant, mais en général, le dispositif est équipé d'un contrôleur intégré qui gère l'affichage. Les données sont envoyées au contrôleur via l'interface I2C, qui décode les commandes de contrôle et affiche les données sur l'écran.

Les afficheurs LCD I2C sont populaires en raison de leur faible encombrement et de leur facilité d'utilisation. Ils sont souvent utilisés dans les projets électroniques pour afficher des messages, des valeurs de capteurs, des graphiques, etc. Les avantages de l'utilisation d'un

afficheur LCD I2C incluent une faible consommation d'énergie (Alimentation 5V), une interface simple et une grande flexibilité dans la personnalisation de l'affichage. [21]

2.3.2 Outils de développement logiciel

Les outils logiciels nécessaires pour développer notre système de présence portable basé sur RFID comprennent :

2.3.2.1 Arduino IDE

Le logiciel Arduino IDE est une interface de développement intégrée (IDE) qui permet de programmer facilement la carte Arduino en utilisant un langage de programmation simplifié basé sur le langage « Processing ». Lorsque vous écrivez un programme dans l'IDE, celui-ci est traduit en langage C et transmis au compilateur avr-gcc, qui effectue la traduction finale en langage compréhensible par le microcontrôleur.

Lorsque vous êtes prêt à télécharger votre programme sur la carte Arduino, vous branchez la carte à l'ordinateur via un câble USB et appuyez sur le bouton "Téléverser". L'IDE compile alors le code et le transfère sur la carte via la connexion USB. En quelques secondes, la carte redémarre et commence à exécuter le programme que vous avez écrit [15].

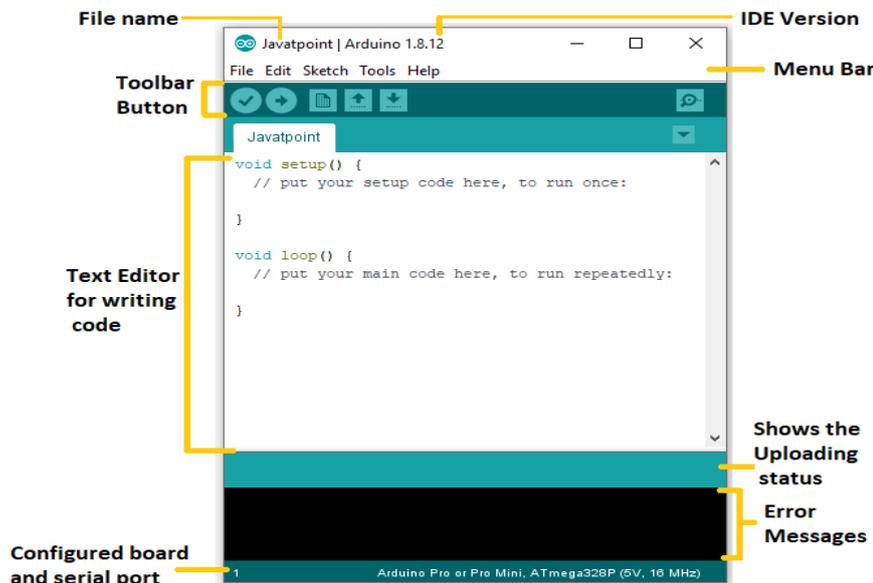


Figure 2.6: Logiciel Arduino IDE [15]

La bibliothèque est un élément important dans le fonctionnement du logiciel Arduino IDE. Elle est composée de fichiers qui contiennent des codes pré-écrits qui peuvent être utilisés pour

simplifier la programmation de projets Arduino. Les bibliothèques fournissent une interface simplifiée pour effectuer des tâches spécifiques, telles que la lecture de données depuis un capteur ou la communication avec un écran LCD.

Lorsqu'une bibliothèque est incluse dans le code Arduino, le compilateur de l'IDE vérifie le code pour détecter les fonctions de la bibliothèque et les utilise pour générer le code binaire pour le microcontrôleur. Les bibliothèques peuvent être téléchargées depuis des sources en ligne ou créées à partir de zéro. Certaines bibliothèques sont incluses avec le logiciel Arduino IDE par défaut, tandis que d'autres doivent être téléchargées séparément [22].

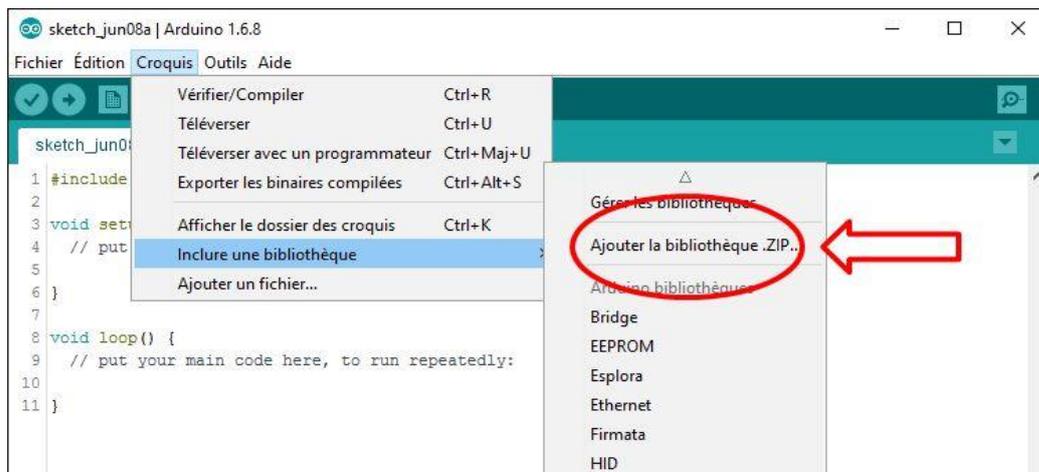


Figure2.7 : Les bibliothèques du logiciel Arduino IDE

2.3.2.2 Le langage de programmation

Dans le contexte d'Arduino IDE, le langage C++ est utilisé pour écrire des programmes pour les cartes Arduino. Les programmes écrits dans Arduino IDE sont appelés des "esquisses" ou "sketches" en anglais. Les sketches Arduino sont écrits en C++ avec des bibliothèques spéciales pour interagir avec les composants électroniques. La plupart des fonctions courantes utilisées dans les sketches Arduino, telles que `digitalWrite()` et `pinMode()`, sont définies dans la bibliothèque standard d'Arduino.

Les développeurs d'Arduino ont simplifié le processus de programmation en fournissant des fonctions d'abstraction de bas niveau pour les tâches courantes, ce qui signifie qu'ils n'ont pas besoin de comprendre toutes les complexités du C++ pour écrire des programmes pour les cartes Arduino. Les développeurs peuvent également créer leurs propres bibliothèques pour étendre les fonctionnalités d'Arduino et les partager avec la communauté.[24]

Ce langage de programmation peut être divisé en trois parties principales : les fonctions, les valeurs (variables et constantes) et la structure.

Les fonctions sont des blocs de code qui effectuent une tâche spécifique. Elles peuvent être prédéfinies (fournies avec le logiciel Arduino) ou créées par l'utilisateur pour répondre à ses besoins spécifiques. Par exemple, une fonction peut être utilisée pour allumer une LED connectée à la carte Arduino.

Fonction	Description
pinMode()	Cette fonction définit le mode de fonctionnement d'une broche (entrée ou sortie).
digitalRead()	Cette fonction lit l'état d'une broche numérique (haut ou bas).
digitalWrite()	Cette fonction envoie un signal haut ou bas à une broche numérique.
analogRead()	Cette fonction lit la valeur d'une broche analogique (entre 0 et 1023).
analogWrite()	Cette fonction envoie une valeur de signal analogique à une broche PWM (de 0 à 255).
delay()	Cette fonction interrompt le programme pour une durée déterminée en millisecondes.
millis()	Cette fonction renvoie le nombre de millisecondes écoulées depuis le début du programme.
Serial.begin()	Cette fonction initialise la communication série à une certaine vitesse (bauds).
Serial.print()	Cette fonction envoie des données série sous forme de texte lisible par l'homme.
Serial.read()	Cette fonction lit les données série entrantes.

Tableau2.3 : Fonctions utiliser dans le langage de programmation Arduino-IDE

Les valeurs (variables et constantes) sont des éléments qui stockent des informations. Les variables peuvent être modifiées tout au long du programme, tandis que les constantes conservent leur valeur initiale. Les valeurs peuvent être de différents types, tels que des entiers, des flottants ou des caractères.

Variable	Description
int	Entier (nombre entier)
float	Nombre à virgule flottante
boolean	Valeur booléenne (vrai ou faux)
char	Caractère unique
String	Chaîne de caractères
byte	Octet (nombre entier de 0 à 255)
unsigned int	Entier non signé (nombre entier positif uniquement)

Tableau 2.4 : Variables utilisées dans le langage de programmation Arduino-IDE

La structure fait référence à la manière dont le programme est organisé, c'est-à-dire comment les fonctions et les valeurs sont utilisées pour résoudre le problème ou réaliser l'objectif du programme. La structure d'un programme peut varier considérablement en fonction de sa complexité et de ses besoins spécifiques.

Structure	Description
setup()	Une fonction préprogrammée qui s'exécute une seule fois au démarrage de la carte Arduino, utilisée pour l'initialisation de la configuration matérielle.
loop()	Une autre fonction préprogrammée qui s'exécute en boucle continue après la fin de la fonction setup(). Elle est utilisée pour exécuter des tâches répétitives ou pour surveiller des capteurs.

if/else	Une structure de contrôle qui permet d'exécuter une ou plusieurs actions en fonction d'une condition donnée.
while	Une structure de contrôle qui permet de répéter une action tant qu'une condition est vraie.
for	Une structure de contrôle qui permet d'exécuter une boucle en spécifiant une valeur initiale, une condition et une instruction d'incrément.
switch/case	Une structure de contrôle qui permet d'exécuter une action en fonction de la valeur d'une variable donnée.
do/while	Une structure de contrôle qui permet d'exécuter une action au moins une fois, puis de répéter l'action tant qu'une condition est vraie.

Tableau 2.5 : Structures utilisées dans le langage de programmation Arduino-IDE

En utilisant ces trois éléments, les programmeurs peuvent écrire des programmes pour contrôler les différentes fonctionnalités de la carte Arduino. Ils peuvent utiliser des fonctions pour allumer et éteindre des LED, lire des capteurs, communiquer avec d'autres appareils et bien plus encore. Les variables et les constantes permettent de stocker et de manipuler les données nécessaires au programme, tandis que la structure organise le code pour réaliser l'objectif souhaité.

En somme, le langage de programmation Arduino est une combinaison de fonctions, de valeurs et de structure, qui permet aux programmeurs de créer des programmes pour contrôler les différentes fonctionnalités de la carte Arduino.[24]

2.3.2.3 La communication série

La communication série est un protocole de communication de données où les bits de données sont transmis un à un, en série, sur une ligne de transmission. C'est une méthode couramment utilisée pour la communication entre les microcontrôleurs, les capteurs, les périphériques électroniques et les ordinateurs.

Le fonctionnement de la communication série consiste à transmettre les données sous forme de bits sur une seule ligne de transmission. Les données sont encodées en utilisant des protocoles de communication tels que RS-232, RS-485, USB, etc. Les données sont transmises en série, ce qui signifie que chaque bit est envoyé l'un après l'autre, dans un ordre préétabli.

Pour établir une connexion série entre deux périphériques, les deux appareils doivent utiliser les mêmes paramètres de communication, tels que la vitesse de transmission des données (baud rate), le nombre de bits de données, le nombre de bits de parité et le nombre de bits d'arrêt. Ces paramètres sont souvent définis dans le code de programmation des microcontrôleurs ou dans les logiciels de communication sur l'ordinateur.

Pour établir une connexion série avec un microcontrôleur Arduino, on peut utiliser la bibliothèque de communication série intégrée, appelée "Serial". Les instructions de base pour établir une connexion série avec un Arduino sont les suivantes :

1. Inclure la bibliothèque "**Serial.h**" à l'aide de la directive `#include`.
2. Appeler la fonction **Serial.begin()** dans la fonction **setup()** pour initialiser la communication série avec la vitesse de transmission des données (baud rate) souhaitée. Par exemple, pour une vitesse de transmission de 9600 bauds, on peut utiliser la commande **Serial.begin(9600)**.
3. Utiliser les fonctions **Serial.print()** ou **Serial.println()** pour envoyer des données sur la ligne de transmission série. Par exemple, pour envoyer la chaîne de caractères "Hello World" sur la ligne de transmission série, on peut utiliser la commande **Serial.println("Hello World")**.
4. Utiliser la fonction **Serial.available()** pour vérifier si des données sont disponibles sur la ligne de transmission série, puis la fonction **Serial.read()** pour lire les données reçues. Par exemple, pour lire une chaîne de caractères envoyée sur la ligne de transmission série, on peut utiliser le code suivant :

```
1 if (Serial.available()) {  
2     String message = Serial.readString();  
3     Serial.println("Le message reçu est : " + message);  
4 }
```

En utilisant ces instructions, il est possible d'établir une connexion série entre un microcontrôleur Arduino et un ordinateur ou un autre périphérique électronique compatible avec la communication série.[25]

2.3.2.4 Les bibliothèques

Les bibliothèques sont des collections de fonctions et de procédures prédéfinies dans un langage de programmation particulière qui peuvent être incluses dans un programme pour effectuer des tâches spécifiques. Dans le contexte de l'Arduino, les bibliothèques sont des ensembles de fichiers d'en-tête (.h) qui fournissent des fonctions permettant d'interagir avec des périphériques externes, tels que des capteurs, des modules de communication, des écrans, etc.

a) La bibliothèque « LiquidCrystal.h »

La bibliothèque "LiquidCrystal.h" est une bibliothèque standard utilisée pour contrôler des écrans LCD alphanumériques. Elle fournit des fonctions pour initialiser et contrôler l'écran, ainsi que pour écrire du texte et positionner le curseur sur l'écran.[26]

Voici un exemple de programme utilisant la bibliothèque "LiquidCrystal.h" pour afficher un message sur un écran LCD :

```

1  #include <LiquidCrystal.h>
2
3  // initialisation de L'écran
4  LiquidCrystal lcd(12, 11, 5, 4, 3, 2);
5
6  void setup() {
7  // définition du nombre de colonnes et de lignes de L'écran
8  lcd.begin(16, 2);
9  // affichage du message sur la première ligne
10  lcd.print("Bonjour !");
11  }
12
13  void loop() {
14  // ne rien faire dans la boucle principale
15  }

```

Dans ce programme, la bibliothèque "LiquidCrystal.h" est incluse à l'aide de la directive « **#include** ». La variable « **lcd** » est ensuite initialisée avec les broches utilisées pour connecter l'écran à la carte Arduino. La fonction « **lcd.begin(16, 2)** » définit le nombre de colonnes et de lignes de l'écran, ici 16 et 2 respectivement. Enfin, la fonction « **lcd.print("Bonjour !")** » affiche le message "Bonjour !" sur la première ligne de l'écran.

Le fonctionnement de la bibliothèque repose sur une communication série entre la carte Arduino et l'écran LCD. Les données sont envoyées à l'écran par l'intermédiaire de broches spécifiques, généralement les broches 4 à 7 de la carte Arduino. La bibliothèque prend en charge les tâches de bas niveau nécessaires pour communiquer avec l'écran, ce qui facilite grandement le développement d'applications utilisant des écrans LCD.[27]

b) La bibliothèque « MFRC522.h »

La bibliothèque "MFRC522.h" est une bibliothèque standard utilisée pour interagir avec des modules de lecture et d'écriture de cartes RFID basés sur le chipset MFRC522 de NXP. Elle fournit des fonctions pour initialiser et contrôler le module RFID, ainsi que pour lire et écrire des données sur des cartes RFID.

Le fonctionnement de la bibliothèque repose sur une communication série entre la carte Arduino et le module RFID. Les données sont échangées en utilisant le protocole SPI (Serial Peripheral Interface), qui utilise des broches spécifiques de la carte Arduino pour transmettre les données. La bibliothèque prend en charge les tâches de bas niveau nécessaires pour communiquer avec le module RFID, ce qui facilite grandement le développement d'applications utilisant des cartes RFID [29].

Voici un exemple de programme utilisant la bibliothèque "MFRC522.h" pour lire l'identifiant d'une carte RFID et l'afficher sur le moniteur série :

```

1 #include <SPI.h>
2 #include <MFRC522.h>
3
4 #define RST_PIN 9
5 #define SS_PIN 10
6

```

```

7 MFRC522 mfrc522(SS_PIN, RST_PIN); // Crée un objet MFRC522
8
9 void setup() {
10   Serial.begin(9600); // Initialise le port série
11   SPI.begin(); // Initialise la communication SPI
12   mfrc522.PCD_Init(); // Initialise le module RFID
13   Serial.println("Lecteur RFID MFRC522 connecte");
14 }
15
16 void loop() {
17   // Vérifie si une carte est présente
18   if (mfrc522.PICC_IsNewCardPresent() &&
19 mfrc522.PICC_ReadCardSerial()) {
20     // Affiche l'identifiant de la carte sur le moniteur série
21     Serial.print("Carte detectee : ");
22     for (byte i = 0; i < mfrc522.uid.size; i++) {
23       Serial.print(mfrc522.uid.uidByte[i] < 0x10 ? "0" : "");
24       Serial.print(mfrc522.uid.uidByte[i], HEX);
25     }
26     Serial.println();
27   }
}

```

Dans ce programme, la bibliothèque "**MFRC522.h**" est incluse à l'aide de la directive « **#include** ». Les variables « **RST_PIN** » et « **SS_PIN** » sont définies pour spécifier les broches utilisées pour connecter le module RFID à la carte Arduino. L'objet **MFRC522** est ensuite créé en utilisant ces broches.

La fonction « **setup()** » initialise le port série, la communication SPI et le module RFID. La fonction « **loop()** » vérifie s'il y a une carte RFID présente dans le champ de lecture du module. La fonction « **mfrc522.PICC_IsNewCardPresent()** » renvoie une valeur booléenne qui indique si une nouvelle carte a été détectée. Si c'est le cas, la fonction « **mfrc522.PICC_ReadCardSerial()** » est appelée pour lire l'identifiant de la carte.

Ensuite, l'identifiant de la carte est affiché sur le moniteur série en utilisant la fonction « **Serial.print()** ». La boucle « **for** » parcourt les octets de l'identifiant de la carte (contenus dans l'objet « **mfrc522.uid.uidByte[]** ») et les affiche en format HEX, avec un "0" devant si l'octet est inférieur à 0x10 pour avoir un affichage cohérent.

Enfin, la fonction « **Serial.println()** » est utilisée pour passer à la ligne suivante sur le moniteur série après l'affichage de l'identifiant de la carte.[29]

c) La bibliothèque « **SD.h** »

La bibliothèque "SD.h" est une bibliothèque standard utilisée pour interagir avec des cartes SD et SDHC (Secure Digital High Capacity) via une interface SPI (Serial Peripheral Interface) sur une carte Arduino. Elle fournit des fonctions pour initialiser et contrôler la carte SD, ainsi que pour lire et écrire des fichiers sur la carte SD.

Le fonctionnement de la bibliothèque repose sur une communication série entre la carte Arduino et la carte SD. Les données sont échangées en utilisant le protocole SPI, qui utilise des broches spécifiques de la carte Arduino pour transmettre les données. La bibliothèque prend en charge les tâches de bas niveau nécessaires pour communiquer avec la carte SD, ce qui facilite grandement le développement d'applications utilisant des cartes SD [30].

Voici un exemple de programme utilisant la bibliothèque "SD.h" pour écrire des données dans un fichier sur une carte SD :

```

1  #include <SD.h>
2
3  const int chipSelect = 10;
4
5  File dataFile;
6
7  void setup() {
8      Serial.begin(9600);
9
10     while (!Serial) {
11         ;
12     }
13

```

```
14 Serial.print("Initialisation de la carte SD...");
15
16 if (!SD.begin(chipSelect)) {
17     Serial.println("La carte SD n'est pas disponible");
18     return;
19 }
20
21 Serial.println("La carte SD est initialisée");
22
23 dataFile = SD.open("donnees.txt", FILE_WRITE);
24
25 if (dataFile) {
26     Serial.println("Ecriture des donnees...");
27     dataFile.println("Premiere ligne de donnees");
28     dataFile.println("Deuxieme ligne de donnees");
29     dataFile.close();
30     Serial.println("Donnees ecrites dans le fichier");
31 } else {
32     Serial.println("Erreur d'ouverture du fichier");
33 }
34 }
35
36 void loop() {
37     // rien ici
38 }
```

Dans ce programme, la bibliothèque "SD.h" est incluse à l'aide de la directive « **#include** ». La variable `chipSelect` est définie pour spécifier la broche utilisée pour connecter la carte SD à la carte Arduino. La fonction « **SD.begin()** » initialise la carte SD avec la broche de sélection spécifiée.

La fonction « **setup()** » initialise le port série et la carte SD. Si la carte SD n'est pas disponible, le programme se termine. Si la carte SD est disponible, un fichier nommé "donnees.txt" est ouvert en mode écriture avec la fonction « **SD.open()** ». Les données sont ensuite écrites dans le fichier à l'aide de la fonction « **dataFile.println()** ». Enfin, le fichier est fermé avec la fonction « **dataFile.close()** ».

La fonction « **loop()** » ne contient rien dans cet exemple, car le programme n'a pas besoin de faire autre chose que d'écrire les données dans le fichier.[31]

En utilisant cette bibliothèque, il est possible de développer des applications utilisant des cartes SD pour stocker des données, des fichiers, des images, des vidéos, etc.

d) La bibliothèque « **RTClib.h** »

La bibliothèque "RTClib.h" est une bibliothèque standard utilisée pour interagir avec des modules de temps réel (RTC) sur une carte Arduino. Elle fournit des fonctions pour initialiser et contrôler le module RTC, ainsi que pour lire l'heure et la date actuelles.

Le fonctionnement de la bibliothèque repose sur une communication série entre la carte Arduino et le module RTC. Les données sont échangées en utilisant le protocole I2C, qui utilise deux broches spécifiques de la carte Arduino pour transmettre les données. La bibliothèque prend en charge les tâches de bas niveau nécessaires pour communiquer avec le module RTC, ce qui facilite grandement le développement d'applications utilisant des modules RTC.[32]

Voici un exemple de programme utilisant la bibliothèque "RTClib.h" pour afficher l'heure et la date actuelles sur le moniteur série :

arduinoCopy code

```

1  #include <RTClib.h>
2
3  RTC_DS3231 rtc;
4
5  void setup () {
6    Serial.begin(9600);
7    while (!Serial) {}
8    if (!rtc.begin()) {
9      Serial.println("Module RTC non trouvé !");

```

```

10  while (1);
11  }
12  if (rtc.lostPower()) {
13      Serial.println("RTC perdu l'alimentation, veuillez ajuster
14  l'heure !");
15      rtc.adjust(DateTime(F(__DATE__), F(__TIME__)));
16  }
17  }
18
19  void loop () {
20      DateTime now = rtc.now();
21      Serial.print(now.day(), DEC);
22      Serial.print('/');
23      Serial.print(now.month(), DEC);
24      Serial.print('/');
25      Serial.print(now.year(), DEC);
26      Serial.print(' ');
27      Serial.print(now.hour(), DEC);
28      Serial.print(':');
29      Serial.print(now.minute(), DEC);
30      Serial.print(':');
31      Serial.print(now.second(), DEC);
32      Serial.println();
33      delay(1000);
    }

```

Dans ce programme, la bibliothèque "RTCLib.h" est incluse à l'aide de la directive « **#include** ». La fonction « **setup()** » initialise le port série et le module RTC. Si le module RTC n'est pas détecté, le programme s'arrête. Si le module RTC a perdu l'alimentation, l'heure et la date actuelles sont initialisées avec la fonction « **rtc.adjust()** ».

La fonction « **loop()** » lit l'heure et la date actuelles avec la fonction « **rtc.now()** » et les affiche sur le moniteur série avec la fonction « **Serial.print()** ». Le délai de 1000 millisecondes permet de rafraîchir l'affichage de l'heure toutes les secondes.

En utilisant cette bibliothèque, il est possible de développer des applications utilisant des modules RTC pour afficher l'heure et la date actuelles, programmer des événements en fonction de l'heure, etc. [33]

2.4 Conclusion

En conclusion, ce chapitre a présenté le système de présence RFID portable proposé, ainsi que les outils de développement matériels et logiciels utilisés pour sa conception et son développement. Le microcontrôleur, le lecteur RFID, les étiquettes RFID, le module carte SD, le module RTC et l'affichage LCD sont les principaux composants matériels utilisés pour la réalisation de ce système.

Du côté logiciel, l'Arduino IDE, le langage de programmation, la communication série et les bibliothèques RFID, LCD, carte SD et RTC sont des outils indispensables pour le développement de ce système.

La conception et le développement de ce système de présence RFID portable sont importants car ils permettent d'améliorer la gestion de la présence dans les entreprises, les écoles et autres organisations. Ce système offre une solution efficace pour enregistrer les présences et les absences des employés, des élèves ou des membres d'une organisation, en utilisant la technologie RFID.

En somme, ce système permet de gagner du temps, de réduire les erreurs de saisie, d'assurer une gestion plus efficace des présences et d'obtenir des données en temps réel sur la présence des personnes. Le développement de ce système est donc un atout majeur pour les organisations souhaitant améliorer leur gestion de la présence et optimiser leur productivité.

CHAPITRE 3

IMPLIMENTATION ET

VALIDATION

3.1. Introduction

Ce chapitre se concentre sur la réalisation et la conception d'un système de pointeuse RFID, en abordant la conception matérielle, la programmation logicielle ainsi que les tests de validation. L'objectif est de développer un système fiable et précis pour gérer la présence des utilisateurs. La conception inclut le choix des composants matériels, la programmation des protocoles de communication et la création d'une interface utilisateur conviviale. Les tests visent à évaluer les performances du système et à garantir sa conformité aux spécifications initiales. En somme, ce chapitre présente les étapes clés pour mettre en place un système de pointeuse RFID efficace.

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 composants	Description
Carte Arduino	Un Arduino UNO
Module RFID	Un Module RC522
Datta logger Shields (Lecteur Carte SD +Horloge RTC)	Carte SD Module DS1307
Afficheur LCD	Ecran LCD 16*2 (I2C)
Bouton	2 Boutons
Batterie	9V (non rechargeable)

Résistance	2 Résistances (10Kohm)
Fils de connexion	(Male -male) (male-femelle)
Plaque d'essai	(Maquette moyenne)

Tableau 3.1 : Liste des composants utilisés

Schéma synoptique

Le système repose sur une carte Arduino UNO et peut être divisé en quatre parties principales : l'interface d'affichage à cristaux liquides (LCD), la section d'interface du module RFID, la section d'interface de l'horloge en temps réel et la section d'interface de la carte SD. Ce sont les composants fondamentaux du système

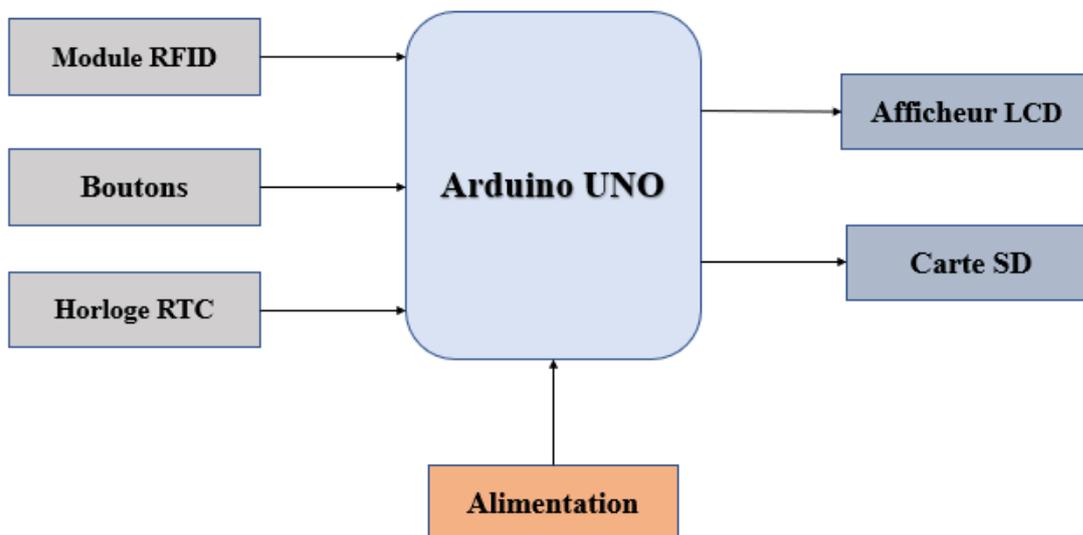


Figure 3.1 : Schéma synoptique du projet

Data Logger Shield

Pour enregistrer et stocker les données collectées, nous utilisons un module Data Logger Shield équipé d'un RTC (Real-Time Clock) intégré et d'une interface pour carte SD. Ce Shield se connecte directement à la carte Arduino UNO, offrant ainsi une solution pratique pour enregistrer les informations sur une carte SD.

Horloge temps réel (RTC) : Le Data Logger Shield utilise généralement le port I2C de l'Arduino pour se connecter à l'horloge temps réel (RTC). Sur la plupart des cartes Arduino, les broches A4 (SDA) et A5(SCL) sont utilisées pour la communication I2C avec le RTC.

Carte SD : Le Data Logger Shield utilise le bus SPI (Serial Peripheral Interface) pour communiquer avec la carte SD. Les broches utilisées peuvent varier selon le modèle du Shields, mais généralement les broches 10 (SS), 11 (MOSI), 12 (MISO) et 13 (SCK) de l'Arduino sont utilisées pour la communication SPI avec la carte SD.



Figure3.2 : Module data logger Shields et la carte Arduino UNO [34]

Connectez le module Data Logger Shield à la carte Arduino : Insérez soigneusement le module Data Logger Shield sur le dessus de la carte Arduino, en veillant à aligner correctement les broches.

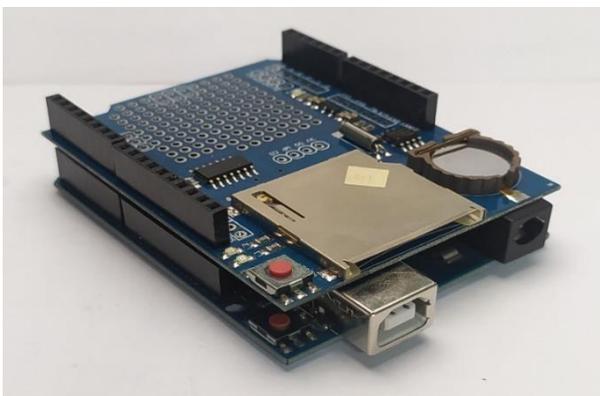


Figure3.3 : Montage du Shields sur l'Arduino [34]

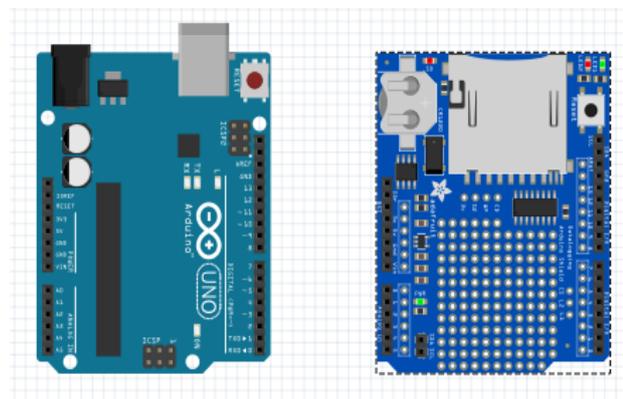


Figure3.4 : Shield avec l'Arduino(fritzing) [34]

Module RFID

Pour alimenter la carte Arduino UNO, il suffit de la connecter à l'ordinateur à l'aide du câble USB fourni. Cette alimentation permet de démarrer le fonctionnement de la carte et de la rendre opérationnelle pour exécuter le programme que nous avons développé.

La fonction principale du module lecteur RFID est de lire les données installées à l'intérieur de l'étiquette RFID (Radio Frequency Identification) pour permettre l'identification sans contact.

Pour connecter un module RFID (RFID-RC522) à une carte Arduino, voici une configuration courante des broches (pins) utilisées :

PINS	CÂBLAGE À ARDUINO
SDA	Digital 10
SCK	Digital 13
MOSI	Digital 11
MISO	Digital 12
IRQ	Ne pas connecter
GND	GND
RST	Digital 9
3.3V	5V

Tableau 3.2: Connexion entre les broches Arduino et RFID

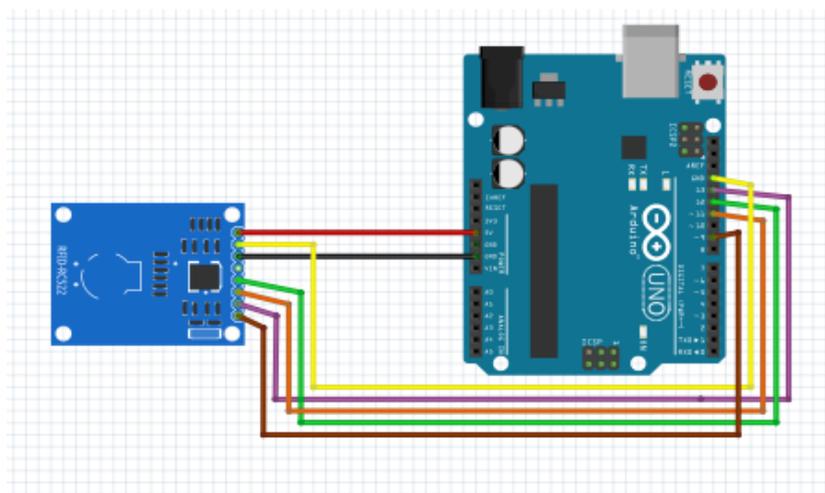


Figure3.5 : Schéma électronique RFID et Arduino [34]

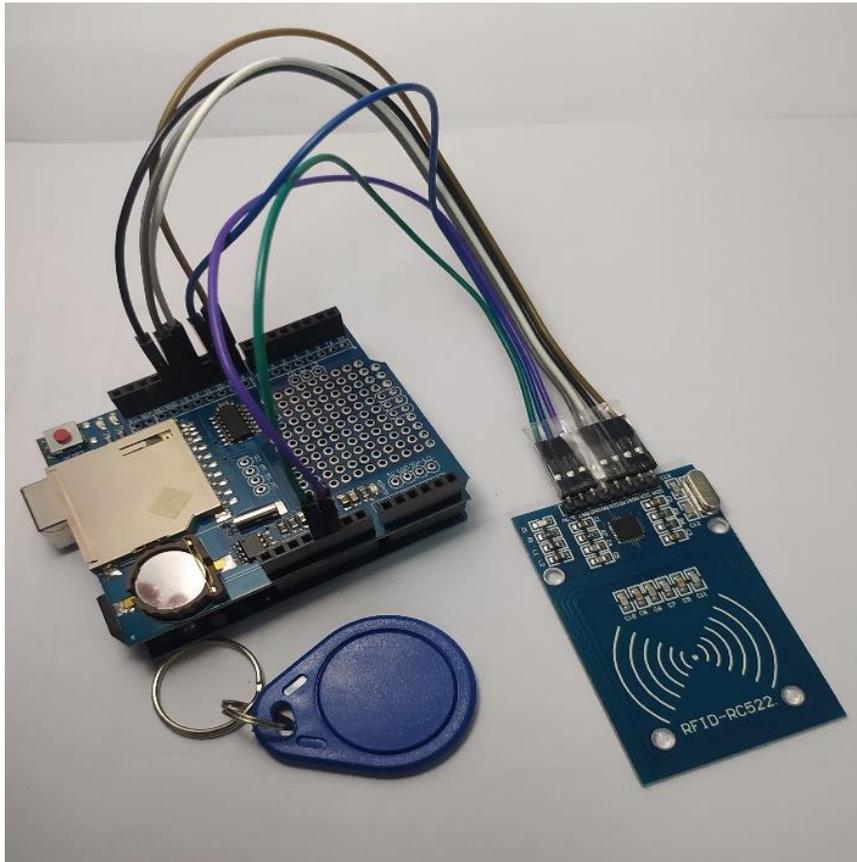


Figure 3.6 : Câblage du module RFID-RC522 avec l'Arduino [34]

Afficheur LCD

Nous utilisons un afficheur LCD I2C 16x2 pour afficher le menu et autres opérations en temps réel. L'afficheur LCD offre une interface conviviale pour afficher du texte et des symboles, et le protocole I2C facilite la connexion avec la carte Arduino en utilisant seulement quelques broches.

PINS	CÂBLAGE À ARDUINO
Vcc	5V
GND	GND
SDA	A4
SCL	A5

Tableau 3.3: Connexion entre les broches Arduino et LCD

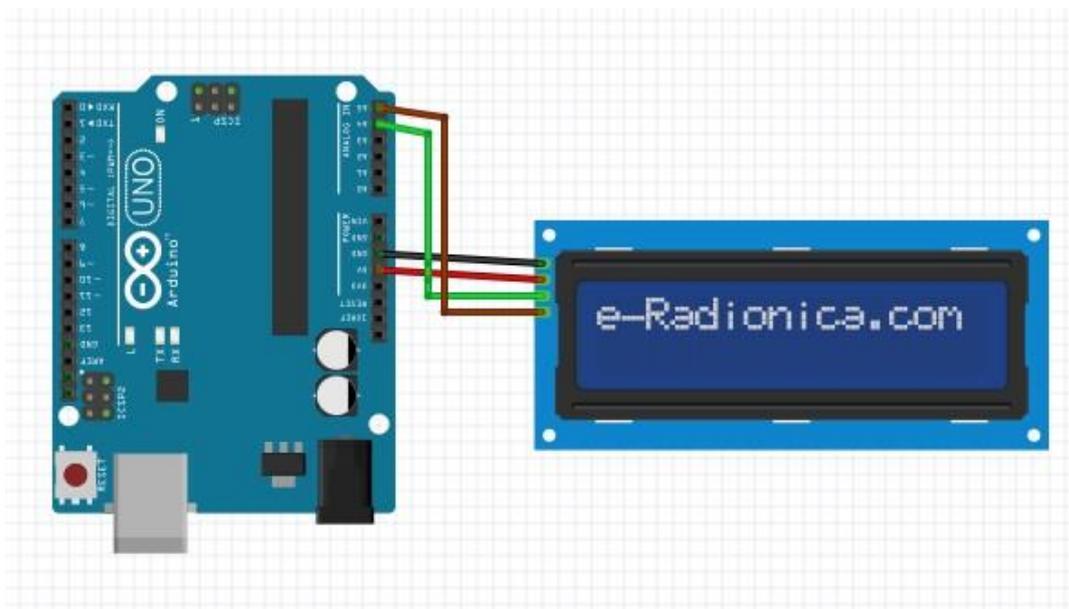


Figure 3.7 : Schéma électrique LCD I2C avec Arduino [34]

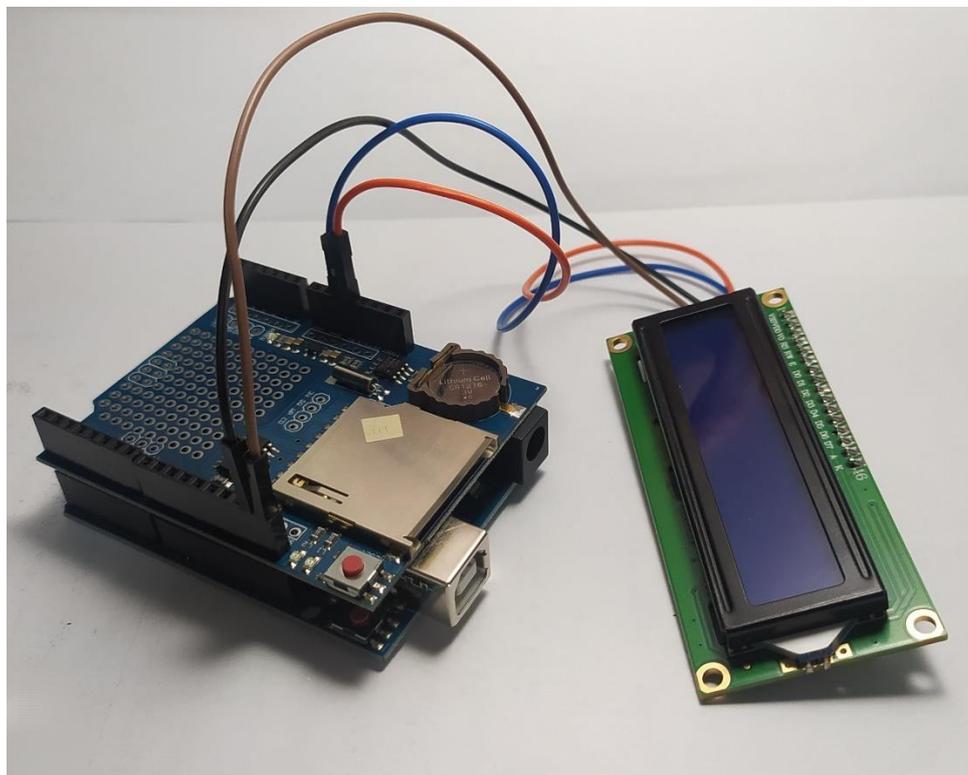


Figure3.8 : Câblage du l'afficheur avec la carte Arduino [34]

Boutons

Dans notre système, nous avons inclus la présentation de deux boutons pour faciliter la navigation dans le menu et la sélection des options. Ces boutons sont essentiels pour offrir une interaction conviviale avec notre interface utilisateur.

PINS	CÂBLAGE À ARDUINO
Bouton 1	Port2
Bouton 2	Port3
GND	GND

Tableau 3.4: Connexion entre les broches Arduino et les boutons

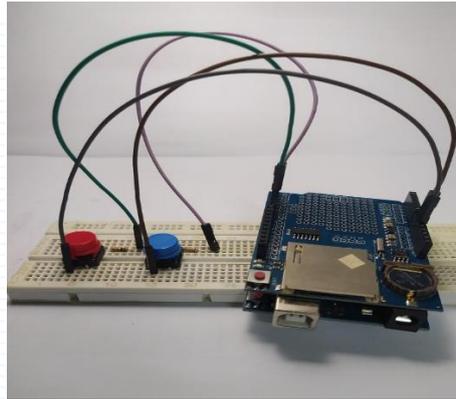
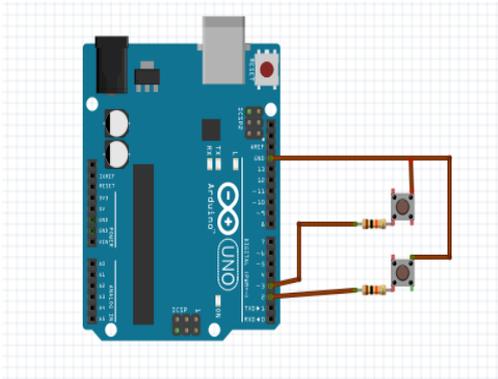


Figure 3.9: Schéma, boutons avec Arduino [34]

Figure 3.10 : câblage des boutons avec l'Arduino [34]

Batterie (non rechargeable)

L'alimentation de notre projet Arduino avec une batterie de 9V offre une solution portable et autonome pour alimenter le microcontrôleur. La tension nominale de la batterie fournit une puissance suffisante pour alimenter l'Arduino et ses composants connectés. Cependant, il est important de vérifier la tension de fonctionnement recommandée de notre modèle d'Arduino afin de s'assurer que la batterie de 9V est compatible. En utilisant une batterie de 9V, nous pouvons profiter des avantages de la portabilité et de l'autonomie de l'Arduino pour notre projet, que ce soit sur le terrain ou dans des environnements où l'accès à une source d'alimentation secteur est limité.



Figure 3.11 : Alimentation avec batterie 9V [34]

Schéma électronique du montage global

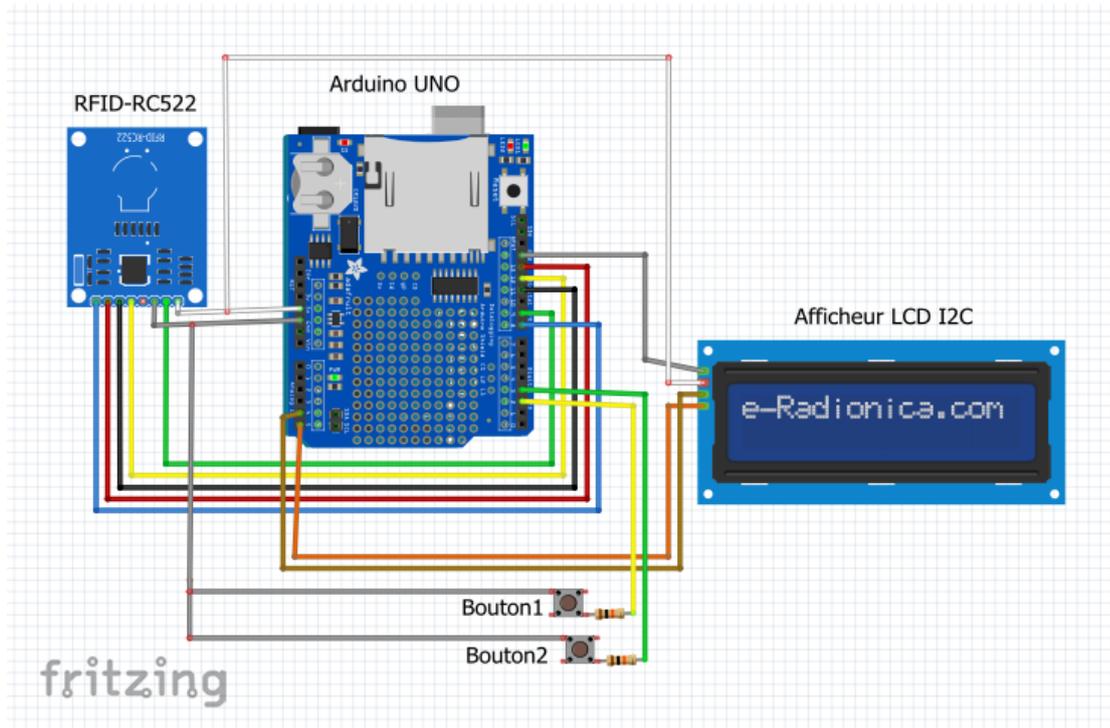


Figure3.12 : Schéma électrique globale (fritzing) [34]

Schéma du montage global

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

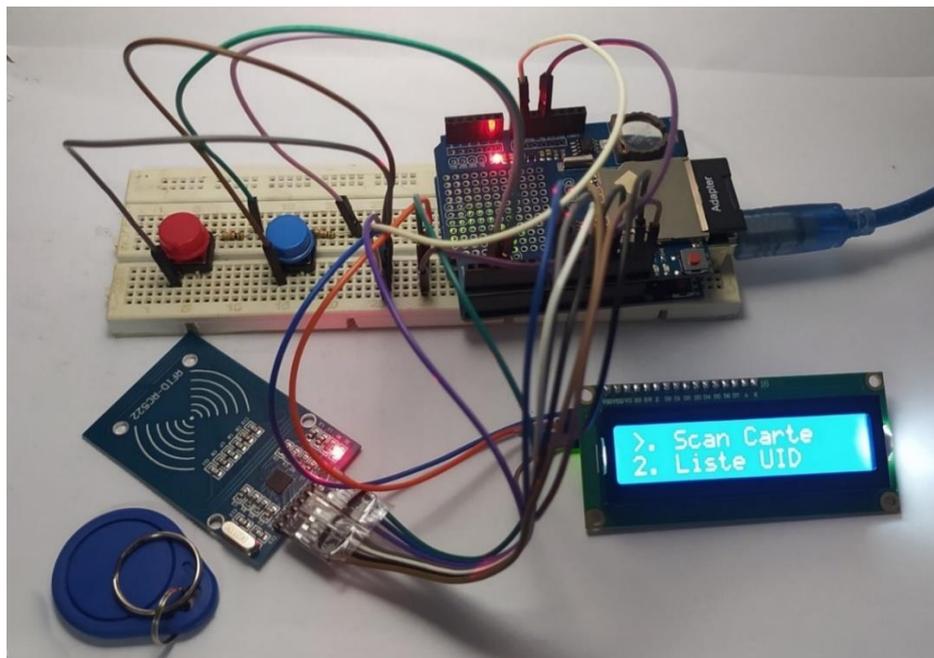


Figure 3.13 : Le circuit globale du system de pointage basé sur RFID [34]

3.3. Implémentation logiciel (Arduino IDE)

La création du matériel ne suffit pas pour assurer le bon fonctionnement du système ; les logiciels jouent un rôle crucial à cet égard. Dans ce projet, la programmation se fait en utilisant Arduino IDE. Une fois la conception et le développement du système terminés, un logiciel d'application sera développé. Un groupe principal de logiciels d'application doit être pris en compte, notamment le processus de la carte qui sera insérée dans l'appareil portable. Lorsque la carte RFID est insérée dans le lecteur RFID, celui-ci passe en mode lecture. Le lecteur identifie ensuite la carte et enregistre les informations dans la carte SD, qui peuvent être récupérées en mode écriture.

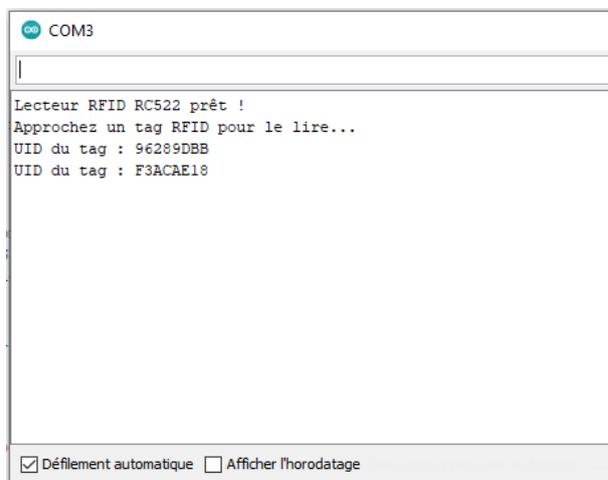
Teste des modules

La phase de test des modules est cruciale dans le cadre de ce projet. Notre première étape consistera à vérifier et confirmer le bon fonctionnement de chaque module, afin de nous assurer qu'ils remplissent leurs fonctions prévues. Parmi les composants à tester, nous avons le module RFID, le module de carte Micro SD, le module LCD et l'Horloge RTC DS1307.

Arduino UNO avec le RFID

Dans le cadre de notre projet, nous avons effectué des tests approfondis du module RFID-RC522. Le module est largement utilisé pour la lecture et l'écriture des cartes RFID. Nous avons connecté le module à Arduino et avons procédé à des essais pour vérifier sa compatibilité et sa performance. Nous avons testé la lecture des cartes RFID, la détection de la présence des cartes. Les résultats ont été prometteurs, le module RFID-RC522 a démontré une excellente capacité de lecture et d'écriture des cartes RFID, ce qui en fait un composant essentiel pour notre projet.

Voici le résultat du teste du module sur le moniteur série :



```

COM3
Lecteur RFID RC522 prêt !
Approchez un tag RFID pour le lire...
UID du tag : 96289DBB
UID du tag : F3ACAE18
  
```

Défilement automatique Afficher l'horodatage

Figure3.14 : Teste de module RFID-RC522 [34]

Arduino UNO avec le Shield (Lecteur carte SD et RTC DS1307)

Nous avons également réalisé des tests des modules de carte SD et de l'horloge RTC DS1307. Nous avons connecté le module à Arduino et avons effectué des tests pour vérifier sa capacité à lire et écrire des données sur la carte. Nous avons également effectué des tests sur le module de l'horloge RTC-DS1307, qui utilisée pour suivre l'heure et fournir une synchronisation horaire au système. Nous avons configuré l'heure correcte et avons testé sa capacité à lire et mettre à jour l'heure. Les résultats des tests ont été concluants, montrant que les modules de carte SD et d'horloge RTC DS1307 offrent de bonnes performances et une précision de réponse, confirmant leur importance dans la réussite de notre projet.

Voici le résultat du teste sur le moniteur série :

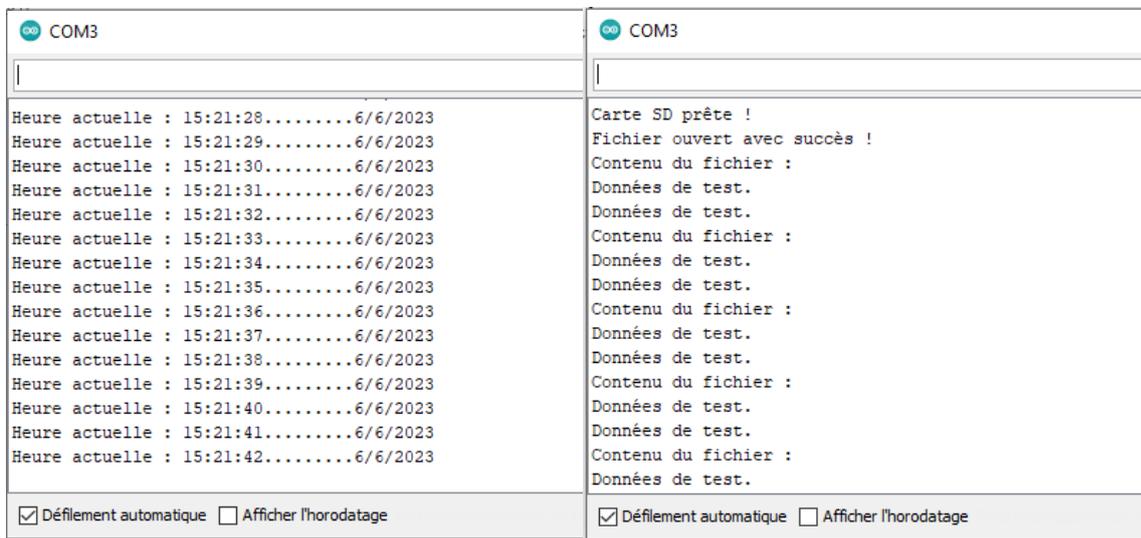


Figure 3.15 : Teste de lecteur carte SD et l’RTC-DS1307 [34]

Arduino UNO avec l’afficheur et les boutons

Dans le cadre de notre projet, nous avons procédé au test d'un afficheur I2C 16x2 et de deux boutons, l'un pour la navigation entre les deux fonctions et l'autre pour la sélection. Nous avons connecté l’LCD à l’Arduino et avons effectué des tests pour vérifier sa compatibilité et son bon fonctionnement. Nous avons également connecté les deux boutons à l’Arduino et avons programmé leur fonctionnement en conséquence et nous avons testé la navigation entre les deux fonctions à l'aide du premier bouton, en changeant l'affichage sur l'afficheur, et avons utilisé le deuxième bouton pour effectuer des sélections. Les résultats des tests ont été positifs, démontrant que l'afficheur I2C 16x2 et les boutons fonctionnent correctement et répondent aux commandes, ce qui les rend essentiels pour l'interaction utilisateur dans notre projet.



Figure 3.16 : Teste de l’LCD et les boutons [34]

Remarque 1 :

On a rencontré un problème lors du branchement du module RFID et du lecteur de carte SD avec l'Arduino, plus précisément dans le port (MISO) des deux composants ce qui causé l’arrêt du fonctionnement des deux modules. Après faire une recherche on trouve que le problème est dans le lecteur SD qui a une erreur de fabrication, On a corrigé ce problème avec l’élimination du lecteur carte SD et l’utilisation du Datta logger Shield qui constituer d’un lecteur SD et une horloge RTC Pour éviter le problème.

Remarque 2 :

On a rencontré un autre problème avec le module RFID, lié à l'alimentation. Normalement, le module RFID nécessite une tension de 3.3V, mais lorsque nous l'avons branché à la sortie 3.3V de l'Arduino, il ne fonctionnait pas correctement. En revanche, lorsque nous l'avons alimenté avec une tension de 5V, il a fonctionné normalement. Il est possible que la sortie 3.3V de l'Arduino ne délivre pas exactement 3.3V comme prévu.

Organigramme générale

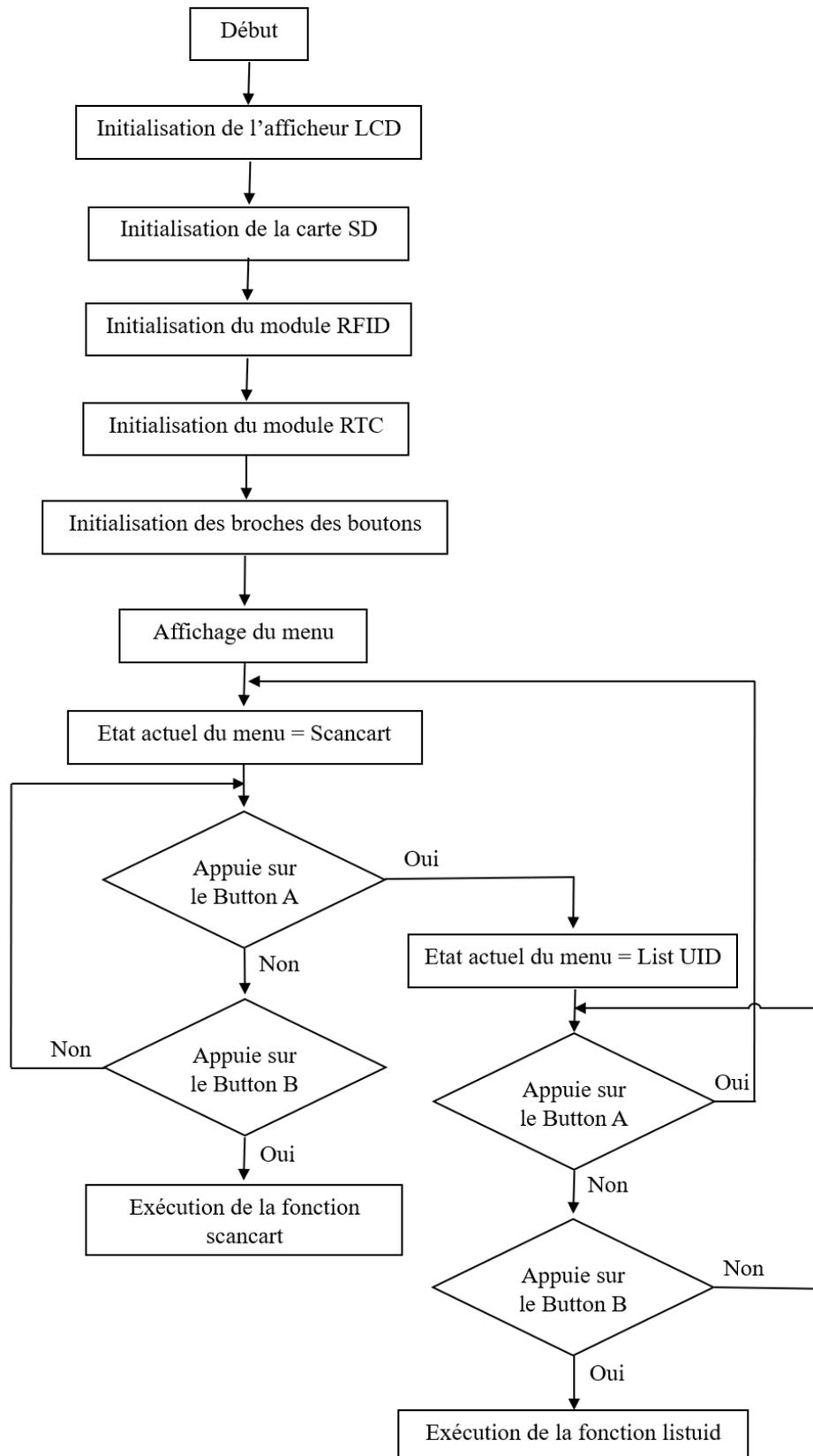


Figure3.17 : Organigramme de fonctionnement générale

Déroulement du programme

```

1  #include <SPI.h>
2  #include <MFRC522.h>
3  #include <SD.h>
4  #include <Wire.h>
5  #include <LiquidCrystal_I2C.h>
6  #include <RTCLib.h>
7
8  // Création de l'objet LiquidCrystal_I2C
9  LiquidCrystal_I2C lcd(0x27, 16, 2);
10
11 // Définition de la broche utilisée pour la sélection de la carte SD
12 const int chipSelect = 10;
13
14 // Création d'une instance du module RFID MFRC522 avec les broches de
15 // communication SPI (SDA, SCK)
16 MFRC522 mfrc522(8, 9);
17
18 // Création d'une instance du module RTC (Real-Time Clock) DS1307
19 RTC_DS1307 rtc;
20
21 File listuid_file;
22 File listpr_file;
23
24 int num = 1 ;
25
26 // Définition des broches des boutons
27 const int BUTTON_A_PIN = 2;
28 const int BUTTON_B_PIN = 3;
29
30 // Énumération des états du menu
31 enum MenuState {
32     SCANCART,
33     LISTUID
34 };
35
36 // Variables du menu
37 MenuState currentMenuState = SCANCART;
38 bool selectPressed = false;

```

Cette partie du code commence par inclure les bibliothèques nécessaires au programme. Ensuite, un objet est créé pour gérer l'affichage sur un écran LCD utilisant la bibliothèque "LiquidCrystal_I2C.h" avec une adresse I2C spécifique. Une variable est définie pour la sélection de la carte SD. Des instances des modules RFID MFRC522 et RTC DS1307 sont créées. Deux fichiers sont déclarés pour stocker les informations des cartes RFID et les personnes ayant accédé à l'emplacement. Une variable est initialisée. Les broches des boutons sont définies. Une énumération est utilisée pour définir les états du menu. Des variables sont initialisées pour gérer l'état courant du menu et l'état de pression du bouton de sélection.

```

1 // Fonction d'affichage du menu
2 void displayMenu() {
3     lcd.clear();
4     lcd.setCursor(0, 0);
5     lcd.print("1. Scan Cart");
6     lcd.setCursor(0, 1);
7     lcd.print("2. List UID");
8
9     lcd.setCursor(0, currentMenuState);
10    lcd.print(">");
11 }
12
13 // Fonction pour gérer les actions du bouton A
14 void handleButtonA() {
15     if (currentMenuState == SCANCART) {
16         currentMenuState = LISTUID;
17     } else {
18         currentMenuState = SCANCART;
19     }
20     displayMenu();
21 }
22
23 // Fonction pour gérer les actions du bouton B
24 void handleButtonB() {
25     selectPressed = !selectPressed;
26 }

```

Cette partie contient trois fonctions pour gérer les actions des boutons et l'affichage du menu sur l'écran LCD. La fonction "displayMenu()" affiche les options du menu avec la sélection actuelle indiquée par une flèche. La fonction "handleButtonA()" est appelée lors de la pression du bouton A pour changer l'option sélectionnée. Elle met à jour la variable "currentMenuState" et actualise l'affichage. La fonction "handleButtonB()" est appelée lors de la pression du bouton B pour inverser l'état de la variable "selectPressed", qui indique si le bouton de sélection est pressé ou non.

```

1 void setup() {
2     lcd.begin(16, 2);
3     lcd.setBacklight(HIGH);
4
5     lcd.clear();
6     lcd.setCursor(0, 0);
7     lcd.print("Initialisation");
8     lcd.setCursor(0, 1);
9     lcd.print("de la carte SD..");
10    delay(1000);

```

```

11 // Initialisation de la carte SD
12 while (!SD.begin(chipSelect)) {
13     Serial.println("Échec de l'initialisation de la carte SD !");
14     lcd.clear();
15     lcd.setCursor(0, 0);
16     lcd.print("Initialisation");
17     lcd.setCursor(0, 1);
18     lcd.print("echouee!");
19     delay(1000);
20     lcd.clear();
21     lcd.setCursor(0, 0);
22     lcd.print("Reinitialisation");
23     lcd.setCursor(0, 1);
24     lcd.print("de la carte SD..");
25     delay(1000);
26 }
27 Serial.println("Carte SD initialisée avec succès.");
28 lcd.clear();
29 lcd.setCursor(0, 0);
30 lcd.print("Carte");
31 lcd.setCursor(0, 1);
32 lcd.print("initialisee.");
33 delay(1000);
34
35 // Initialisation du module RFID
36 SPI.begin();
37 mfrc522.PCD_Init();
38
39 // Initialisation du module RTC
40 Wire.begin();
41 while(!rtc.begin()) {
42     Serial.println("Échec de l'initialisation du module RTC !");
43     lcd.clear();
44     lcd.setCursor(0, 0);
45     lcd.print("Initialisation");
46     lcd.setCursor(0, 1);
47     lcd.print("du RTC echouee!");
48     delay(1000);
49 }
50
51 if (!rtc.isrunning()) {
52     Serial.println("Module RTC non trouvé !");
53     lcd.clear();
54     lcd.setCursor(0, 0);
55     lcd.print("module RTC");
56     lcd.setCursor(0, 1);
57     lcd.print("non trouve!");
58     delay(1000);
59     // rtc.adjust(DateTime(F(__DATE__), F(__TIME__))); // Décommenter cette
60     // ligne pour ajuster la date et l'heure du RTC
61 }
62
63
64 pinMode(BUTTON_A_PIN, INPUT_PULLUP);
65 pinMode(BUTTON_B_PIN, INPUT_PULLUP);
66
67 displayMenu();

```

La fonction "setup()" est exécutée au démarrage du programme et est utilisée pour initialiser les composants et périphériques nécessaires. La première étape consiste à configurer l'écran LCD en spécifiant ses dimensions et à allumer le rétro-éclairage. Ensuite, un message d'initialisation de la carte SD est affiché sur l'écran LCD, et la carte SD est initialisée. Si l'initialisation échoue, un message d'erreur est affiché et une tentative d'initialisation est répétée après un délai. Une fois la carte SD initialisée avec succès, un message de confirmation est affiché. Ensuite, le module RFID et le module RTC sont initialisés. Si l'initialisation du module RTC échoue, un message d'erreur est affiché, et une nouvelle tentative est effectuée après un délai. Si le module RTC ne fonctionne pas, un message d'erreur est affiché, et la date et l'heure peuvent être ajustées manuellement. Enfin, les broches des boutons sont configurées et le menu est affiché sur l'écran LCD.

```

1  void loop() {
2      if (digitalRead(BUTTON_A_PIN) == LOW) {
3          handleButtonA();
4          delay(200); // Anti-rebond
5      }
6
7      if (digitalRead(BUTTON_B_PIN) == LOW) {
8          handleButtonB();
9          delay(200); // Anti-rebond
10     }
11
12     if (selectPressed) {
13         if (currentMenuState == SCANCART) {
14             scancart();
15         } else if (currentMenuState == LISTUID) {
16             listuid();
17         }
18         selectPressed = false;
19         displayMenu();
20         delay(200);
21     }
22
23

```

La fonction "loop()" est exécutée en continu après la fonction "setup()". Elle contient une boucle qui gère les actions des boutons et l'affichage du menu. La première condition vérifie si le bouton A est pressé et appelle la fonction "handleButtonA()" pour gérer l'action du bouton. La deuxième condition vérifie si le bouton B est pressé et appelle la fonction "handleButtonB()". La troisième condition vérifie si le bouton de sélection est pressé en utilisant la variable "selectPressed". Si c'est le cas, elle exécute la fonction correspondant à l'option sélectionnée dans le menu, réinitialise la variable "selectPressed", actualise l'affichage

du menu, et ajoute un délai pour éviter les rebonds. Le programme boucle en continu en attendant les interactions de l'utilisateur avec le menu en appuyant sur les boutons.

Organigramme de la fonction 'scancart' :

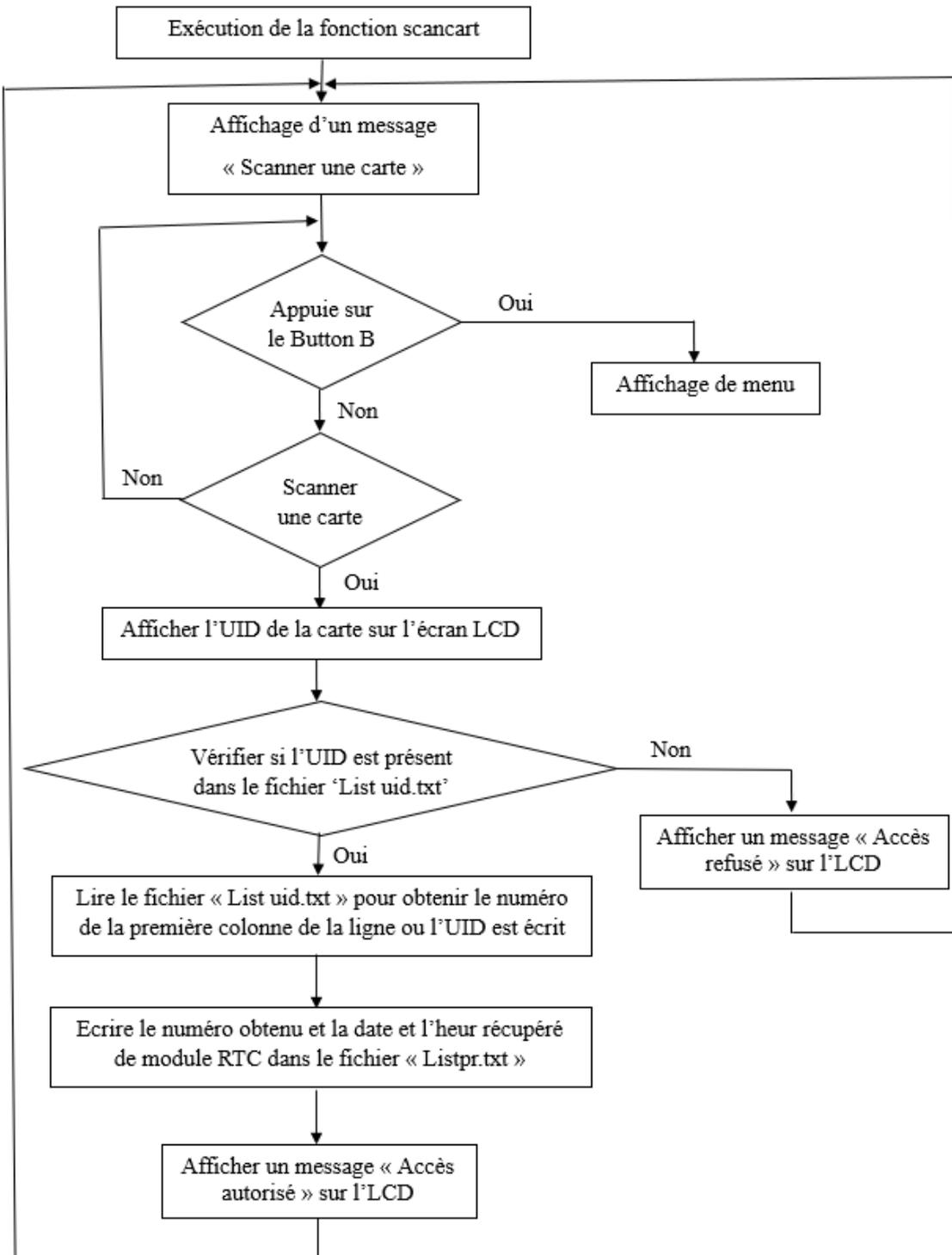


Figure 3.18 : fonctionnement de 'scan carte'

```

1 void scancart(){
2     lcd.clear();
3     lcd.setCursor(0, 0);
4     lcd.print("Scanner");
5     lcd.setCursor(0, 1);
6     lcd.print("une carte :");
7     while (true) {
8         // Vérifier si Le bouton B est appuyé pour sortir de La fonction
9         if (digitalRead(BUTTON_B_PIN) == LOW) {
10            currentMenuState = SCANCART;
11            displayMenu();
12            return;
13        }
14
15        // Attente de La détection d'une carte RFID
16        if (mfrc522.PICC_IsNewCardPresent() && mfrc522.PICC_ReadCardSerial()) {
17            lcd.clear();
18
19            String content = "";
20
21            // Lecture de L'UID de La carte
22            for (byte i = 0; i < mfrc522.uid.size; i++) {
23
24                content.concat(String(mfrc522.uid.uidByte[i] < 0x10 ? "0" : ""));
25                content.concat(String(mfrc522.uid.uidByte[i], HEX));
26            }
27            lcd.print("UID tag :");
28            lcd.setCursor(0, 1);
29            lcd.print(content);
30            delay(1000);
31
32
33            // Vérification de L'UID dans Le fichier listuid.txt
34            listuid_file = SD.open("listuid.txt", FILE_READ);
35            bool uid_found = false;
36            int line_num = 1; // Numéro de ligne courant
37            int line_to_write = -1; // Numéro de ligne où L'UID est trouvé
38
39            if (listuid_file) {
40                while (listuid_file.available()) {
41                    String line = listuid_file.readStringUntil('\n');
42                    if (line.indexOf(content) != -1) {
43                        uid_found = true;
44                        line_to_write = line_num;
45                        lcd.clear();
46                        lcd.setCursor(0, 0);
47                        lcd.print("Acces");
48                        lcd.setCursor(0, 1);
49                        lcd.print("autorise.");
50                        delay(1000);
51                        break;
52                    }
53                    line_num++;
54                }
55                listuid_file.close();

```

```

56     } else {
57         lcd.clear();
58         lcd.setCursor(0, 0);
59         lcd.print("Erreur");
60         lcd.setCursor(0, 1);
61         lcd.print("d'écriture.");
62         delay(1000);
63     }
64
65     if (!uid_found) {
66         lcd.clear();
67         lcd.setCursor(0, 0);
68         lcd.print("Acces");
69         lcd.setCursor(0, 1);
70         lcd.print("refuse.");
71         delay(1000);
72     } else if (line_to_write != -1) {
73         // Écriture du numéro de ligne et de la date dans le fichier listpr.txt
74         DateTime now = rtc.now();
75         String line_to_write_str = String(line_to_write) + " " + String(now.day())
76 + "/" + String(now.month()) + ", " + String(now.hour()) + ":" +
77 String(now.minute());
78
79         listpr_file = SD.open("listpr.txt", FILE_WRITE);
80         if (listpr_file) {
81
82             listpr_file.println(line_to_write_str);
83             listpr_file.close();
84         }
85         else {
86             lcd.clear();
87             lcd.setCursor(0, 0);
88             lcd.print("erreur");
89             lcd.setCursor(0, 1);
90             lcd.print("d'écriture.");
91             delay(1000);
92         }
93     }
94
95     delay(500);
96     lcd.clear();
97     lcd.setCursor(0, 0);
98     lcd.print("Scanner");
99     lcd.setCursor(0, 1);
100    lcd.print("une carte :");
101 }
}
}

```

La fonction "scancart()" est utilisée lorsque l'utilisateur choisit l'option "Scan Cart" dans le menu. Son rôle est de scanner une carte RFID, vérifier si son UID est autorisé (présent dans le fichier "listuid.txt") et enregistrer la date et l'heure de l'accès autorisé dans le fichier "listpr.txt". Tout d'abord, un message est affiché sur l'écran LCD pour indiquer à l'utilisateur de scanner

une carte. Ensuite, la fonction entre dans une boucle infinie en attendant la détection d'une carte RFID à l'aide de `"mfr522.PICC_IsNewCardPresent()"`. Lorsqu'une carte est détectée, son UID est lu avec `"mfr522.PICC_ReadCardSerial()"` et stocké dans la variable `"content"`. L'UID est ensuite affiché sur l'écran LCD.

La fonction vérifie ensuite si l'UID est présent dans le fichier `"listuid.txt"`. Si c'est le cas, l'accès est autorisé et un message est affiché sur l'écran LCD. Si l'UID n'est pas trouvé, un message `"Accès refusé"` est affiché.

Si l'UID est trouvé dans le fichier `"listuid.txt"`, la fonction enregistre le numéro de ligne correspondant à l'UID dans la variable `"line_to_write"`. Ensuite, la date et l'heure de l'accès autorisé sont enregistrées dans le fichier `"listp.txt"`. Le numéro de ligne et la date sont stockés dans la variable `"line_to_write_str"` et écrits dans le fichier avec `"listpr_file.println()"`.

Enfin, un délai de 500 millisecondes est ajouté avant de réinitialiser l'affichage et d'attendre la détection d'une nouvelle carte RFID. Si le bouton B est appuyé à n'importe quel moment pendant l'exécution de la fonction, l'affichage est réinitialisé et la fonction se termine.

Organigramme de la fonction 'listuid' :

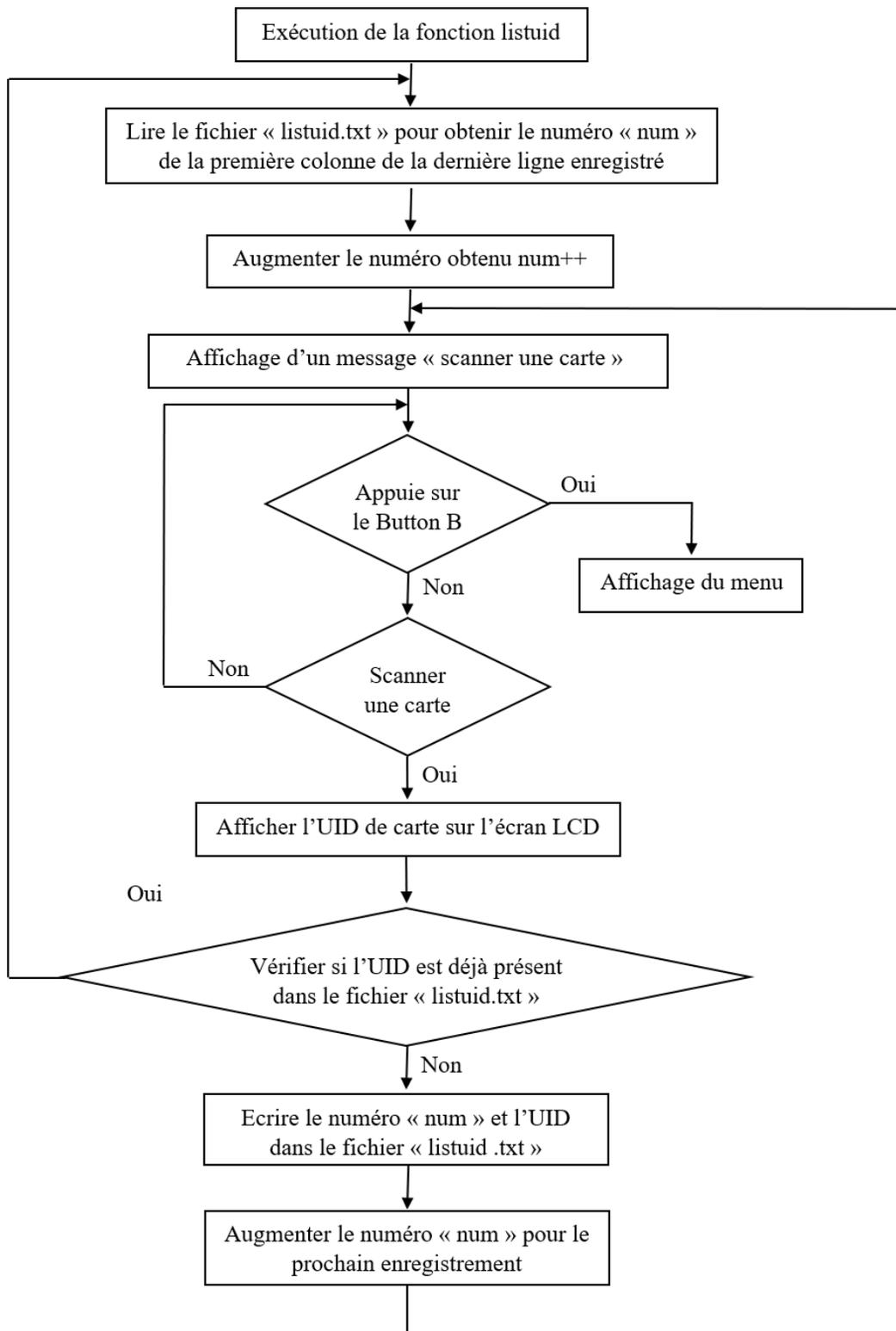


Figure 3.19 : Fonctionnement de 'liste uid'

```

1 void listuid(){
2     lcd.clear();
3     lcd.setCursor(0, 0);
4     lcd.print("Scanner");
5     lcd.setCursor(0, 1);
6     lcd.print("une carte :");
7     File file;
8     //Récupération du dernier numéro dans Le fichier
9     file=SD.open("listuid.txt", FILE_READ);
10    if(file){
11        while (file.available()){
12            String line=file.readStringUntil('\n');
13            int spaceIndex=line.indexOf(' ');
14            if(spaceIndex !=-1){
15                int currentNum=line.substring(0,spaceIndex).toInt();
16                if(currentNum >=num){
17                    num=currentNum +1 ;
18                }
19            }
20        }
21        file.close();
22    }
23    while(true){
24        // Vérifier si Le bouton B est appuyé pour sortir de La fonction
25        if (digitalRead(BUTTON_B_PIN) == LOW) {
26            currentMenuState = SCANCART;
27            displayMenu();
28            return;
29        }
30        // Attente de La détection d'une carte RFID
31        if (mfrc522.PICC_IsNewCardPresent() && mfrc522.PICC_ReadCardSerial()) {
32            lcd.clear();
33            String content = "";
34            // Lecture de L'UID de La carte
35            for (byte i = 0; i < mfrc522.uid.size; i++) {
36                content.concat(String(mfrc522.uid.uidByte[i] < 0x10 ? "0" : ""));
37                content.concat(String(mfrc522.uid.uidByte[i], HEX));
38            }
39            lcd.print("UID tag :");
40            lcd.setCursor(0, 1);
41            lcd.print(content);
42            delay(1000);
43            // Vérification de L'UID dans Le fichier Listuid.txt
44            file = SD.open("listuid.txt", FILE_READ);
45            bool uidfound = false;
46
47            if (file) {
48                while (file.available()) {
49                    String line = file.readStringUntil('\n');
50                    if (line.indexOf(content) != -1) {
51                        uidfound = true;
52                        break;
53                    }
54                }
55                file.close();

```

```

56     }
57     if (uidfound) {
58         // lcd.clear();
59         // lcd.setCursor(0, 0);
60         // lcd.print("carte deja");
61         // lcd.setCursor(0, 1);
62         // lcd.print("dans la liste.");
63         // delay(2000);
64     } else {
65         //écrire l'UID dans le fichier
66         file=SD.open("listuid.txt", FILE_WRITE);
67         if (file){
68             file.print(num);
69             file.print(" ");
70             file.println(content);
71             file.close();
72             // lcd.clear();
73             // lcd.setCursor(0, 0);
74             // lcd.print("carte");
75             // lcd.setCursor(0, 1);
76             // lcd.print("enregistree.");
77             // delay(1000);
78         }
79         num++;
80         delay(500);
81     }
82     delay(500);
83     lcd.clear();
84     lcd.print("Scanner");
85     lcd.setCursor(0, 1);
86     lcd.print("une carte :");
87 }
88 }
89 }

```

La fonction "listuid()" est utilisée lorsque l'utilisateur choisit l'option "List UID" dans le menu. Elle permet de scanner une carte RFID et d'ajouter son UID dans le fichier "listuid.txt" s'il n'est pas déjà présent. Tout d'abord, un message s'affiche sur l'écran LCD pour indiquer à l'utilisateur de scanner une carte. Ensuite, la fonction parcourt le fichier "listuid.txt" pour récupérer le dernier numéro de ligne et le mettre à jour. Cela permet d'ajouter le nouvel UID à la suite des enregistrements existants.

La fonction entre ensuite dans une boucle infinie en attendant la détection d'une carte RFID à l'aide de "mfrc522.PICC_IsNewCardPresent()". Lorsqu'une carte est détectée, son UID est lu avec "mfrc522.PICC_ReadCardSerial()" et affiché sur l'écran LCD. Puis, la fonction vérifie si l'UID est déjà présent dans le fichier "listuid.txt". Sinon, l'UID est ajouté dans le fichier avec son numéro de ligne correspondant, en utilisant les fonctions "file.print()" et "file.println()". Le numéro de ligne est également incrémenté pour la prochaine carte à ajouter.

Enfin, un délai de 500 millisecondes est ajouté avant de réinitialiser l'affichage et d'attendre la détection d'une nouvelle carte RFID. Si le bouton B est pressé à n'importe quel moment pendant l'exécution de la fonction, l'affichage est réinitialisé et la fonction se termine.

Remarque 3 :

Nous avons rencontré un problème lors de l'exécution de notre programme et après plusieurs recherches, nous avons découvert qu'il s'agissait d'un bogue causé par des instructions d'affichage dans l'écran LCD de la fonction "liste uid". Pour résoudre ce problème, nous avons commenté cette partie du code.

En commentant ces lignes, nous désactivons temporairement ces instructions d'affichage dans l'écran LCD. Cela devrait aider à résoudre le bogue que nous avons rencontré

3.4 Teste et validation

Dans ce chapitre, nous présenterons en détail les résultats obtenus tout au long du processus, depuis l'affichage du menu (les deux fonction 'Scan carte' et 'List uid') sur l'LCD jusqu'à la sauvegarde du fichier de présence dans la carte SD.

Premièrement, lorsque la carte Arduino est mise sous tension commence par initialiser le port série et l'affichage LCD. S'il y a un échec d'initialisation de la carte SD, un message d'erreur est affiché sur le LCD, puis affiche un message de réinitialisation jusqu'à l'initialisation de la carte SD, après l'écran LCD affiche le menu, deux fonction 'Scan carte' et 'List uid'. Puis il y'aura un bouton pour la navigation entre les fonctions et un autre pour la sélection. Comme montre dans les figures suivantes :

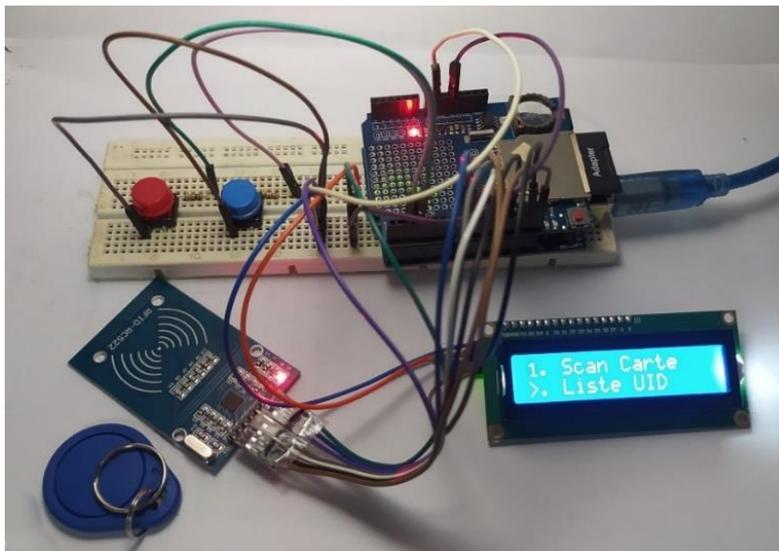


Figure 3.20 : Affichage de notre système



Figure3.21 : carte SD n'est pas initialisé



Figure3.22 : carte SD est initialisé

Lors de la sélection de la deuxième fonction 'liste uid' l'Arduino attend la présence d'une carte RFID et vérifie si le fichier "listuid.txt" existe. Si le fichier n'existe pas, il est créé.

Ensuite, il vérifie si l'UID scanné est déjà présent dans le fichier "listuid.txt". Si l'UID n'est pas déjà présent, il est écrit dans le fichier avec le numéro correspondant. Le numéro est ensuite incrémenté pour la prochaine carte. Si l'UID est déjà présent dans le fichier ne le réécrire pas.

Cela permet de créer une liste d'UID correspondant aux étudiants. Comme montre dans les figures suivantes :



Figure 3.23 : Scanne du l'UID

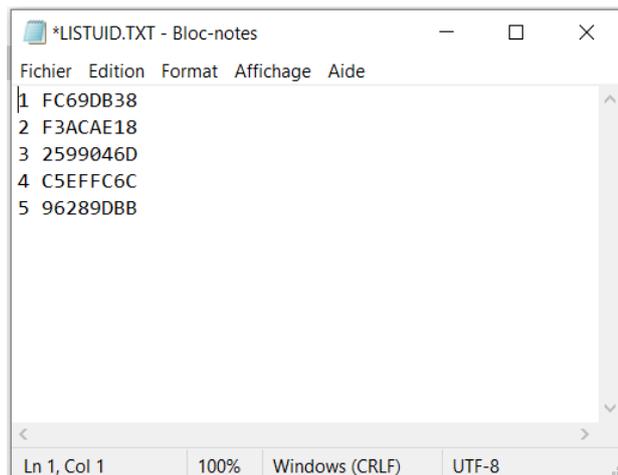


Figure 3.24 : fichier 'listuid.txt'

Lors de la sélection de la première fonction 'scan carte' Le système se met en attente de la détection d'une carte RFID. Une fois qu'une carte est détectée, son UID est extrait et affiché sur l'écran LCD. Ensuite, le code vérifie si cet UID est présent dans le fichier "listuid.txt" situé sur la carte SD. Si l'UID correspond, l'accès est autorisé et un message approprié est affiché sur l'écran LCD.

Ensuite, l'Arduino ouvre le fichier "listegr.txt" (le fichier de présence) et écrit la date et l'heure à la ligne correspondant au numéro de l'UID scannée dans le fichier "listuid.txt".

Si l'UID n'est pas trouvé dans le fichier, un message d'accès refusé est affiché. Lorsque l'UID est trouvé, le numéro de ligne correspondant ainsi que la date actuelle sont enregistrés dans le fichier "listpr.txt" sur la carte SD.



Figure 3.25 : Affichage dans 'scan carte' [34]

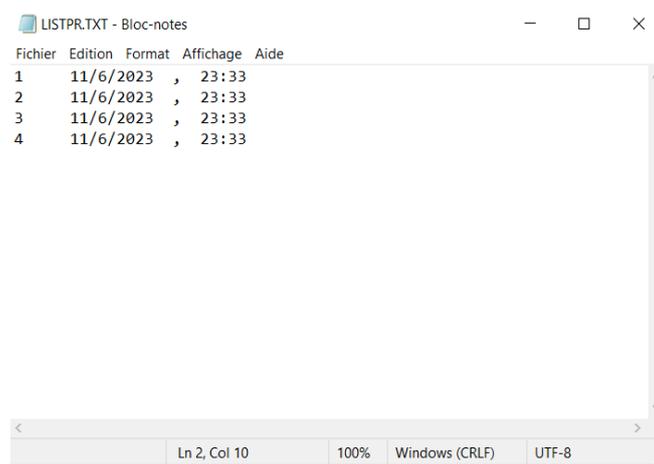


Figure 3.26 : fichier liste de présence [34]

Après un court délai, l'écran LCD est réinitialisé pour afficher à nouveau le message d'attente de détection de carte RFID. Ce résumé met l'accent sur les fonctionnalités clés du programme et sur le flux général de son exécution.



Figure 3.27 : fichiers criés dans la carte SD [34]

3.4 Conclusion

En conclusion, ce chapitre a décrit la conception et la réalisation d'un système de pointage utilisant la technologie RFID, en utilisant une carte Arduino UNO. Les différentes étapes de mise en place du système ont été présentées, notamment l'interconnexion des composants et la programmation de la carte à microcontrôleur ARDUINO. Des tests ont été effectués pour évaluer le fonctionnement du système, et les résultats obtenus ont été discutés.

Ce projet a permis de développer un système de pointage automatisé basé sur la technologie RFID, offrant une solution pratique et efficace pour enregistrer la présence des étudiants. L'utilisation de la carte Arduino UNO a facilité l'interconnexion des composants et la programmation du système. Les tests réalisés ont démontré que le système fonctionnait de manière fiable et précise, en identifiant de manière unique les étudiants portant des étiquettes RFID et en enregistrant leurs données de présence.

CONCLUSION
GENERALE

Conclusion générale

L'objectif de cette recherche est d'introduire la technologie RFID dans le domaine de l'éducation afin de résoudre les problèmes liés à la méthode de prise de présence manuelle. Actuellement, les instructeurs doivent consigner manuellement la présence de leurs étudiants à chaque cours, une pratique encore largement répandue dans la plupart des institutions du monde.

Cependant, ces méthodes de présence manuelles et électroniques présentent plusieurs inconvénients tels que la perte de temps, les risques d'erreurs, la nécessité d'efforts humains, la fatigue et l'inefficacité. Le processus de pointage est crucial tant dans les établissements d'enseignement que dans les lieux de travail. Cette recherche se concentre sur surmonter les principales limites des méthodes de prise de présence existantes en proposant un système entièrement automatisé basé sur la technologie RFID.

Le système de présence basé sur la RFID offre une méthode pratique de prise de présence en utilisant un lecteur RFID pour identifier de manière unique chaque étudiant portant une étiquette RFID et enregistrer ses données de présence.

REFERENCES

- [1] M. F. Omar et al, "Smart Attendance System for Universities", International Conference on Information and Communication Technology for the Muslim World (ICT4M), 2018.
- [2] Sijie Wang et al, "Development of an Attendance Management System for Lectures and Laboratories Based on the Internet of Things", Journal of Physics: Conference
- [3] Nor Aini BT Zakaria, "The development of student attendance system", Université technologie PETRONAS, Series.
- [4] Hiroko Kato ·Keng T. Tan ·Douglas Chai, Barcodes for Mobile Devices
- [5] Frédéric Massicotte, « LA BIOMÉTRIE, SA FIABILITÉ ET SES IMPACTS SUR LA PRATIQUE DE LA DÉMOCRATIE LIBÉRALE » Université de QUÉBEC à MONTRÉAL, Novembre 2007.
- [6] Kennedy O. Okokpujie, Etinosa Noma-Osaghae, et al, "Design and Implementation of a Student Attendance System Using Iris Biometric Recognition", International Conference on Computational Science and Computational Intelligence, 2017.
- [7] Y. Zhang, L. Sun, W. Li, and J. Zhang. "A Smart Attendance System Based on Facial Recognition.", 5th International Conference on Computational Intelligence and Applications (ICCIA), Wuhan, China, 2020.
- [8] Pierre GEORGET, Directeur Général de Gencod EAN France « L'identification par Radio Fréquence, Principes et applications » version 4.01, EPCglobal, mars 2004.
- [9] Davinder P, Twinkle K, Preet K. "The RFID Technology and its applications". International Journal of Electronics, Communication & Instrumentation Engineering Research and Development. September 2012.
- [10] Saber Zenagui K et Ouadah A. « Intégration d'une solution RFID dans la gestion de pointage Odo », Mémoire de fin d'étude, Université Abou Bakr Belkaid– Tlemcen 2016- 2017.
- [11] M. Alkhudairi, M. Shahzad, M. J. Mughal, and F. I. Khan. "Design and Implementation of RFID Based Student Attendance System." International Conference on Communication, Computing and Digital Systems (C-CODE), Islamabad, 2018.
- [12] Bhatt, H. and Glover, B. RFID Essentials. O'Reilly Pub, New York. (2006).
- [13] Ahson, S.A. and Ilyas, M. RFID Handbook Applications, Technology, Security, and Privacy (Hardcover). CRC Press, Boca Raton, Florida. (2008).
- [14] S. S. Ahmed, M. A. Ahmed, and M. S. Alam, "Design and Implementation of a Portable RFID Based Student Attendance System", IEEE 15th International Multitopic Conference, Islamabad, Pakistan, 2012, pp. 135-140.
- [15] Banzi, Massimo, et al. "Getting Started with Arduino." Maker Media, Inc., 2011
- [16] N. Gatkal, S. Nalawade, et D. Nimbalkar, « Arduino in agriculture », Dr. Annasaheb Shinde College of Agricultural Engineering and Technology, MPKV, Rahuri, Maharashtra, India.
- [17] Amanpreet Sandhu, "An Overview of IoT Hardware Development Platforms", International Journal on Emerging Technologies V.11, N.5, pages 155-163, 2020.
- [18] Youssef Bachoti, Bassim Belhaj Sendague, Joao Gabriel Rodrigues Oliveira, « Projet RFID », Telecom sud paris, janvier 2011.
- [19] U. Farooq, M. ul Hasan, M. Amar, et al. "RFID based security and access control system", Journal IACSIT Int. J. Eng. Technol.V6, N4, pages 309-314, 2014.
- [20] Guide de mise en marche du shield data-logger GT1046 ,GO TRONIC
- [21] Warren Gay , Custom Raspberry Pi Interfaces , I2C LCD Displays
- [22] Mert Arduino, "How to Add Library in Arduino IDE", 2021.
- [23] Bjarne Stroustrup, "The C++ Programming Language", Pearson Education, 2013.
- [24] "Arduino Programming Language (based on Wiring)" (2022). Arduino Reference. Retrieved from <https://www.arduino.cc/reference/en/> (accessed May 18, 2023)
- [25] Documentation de la bibliothèque "SPI.h" sur le site Arduino : Site web :

- <https://www.arduino.cc/reference/en/language/functions/communication/spi/> (accessed May 18, 2023)
- [26] Simon Monk, "Programming Arduino: Getting Started with Sketches", 2016.
- [27] Documentation de la bibliothèque "LiquidCrystal.h" sur le site Arduino : Site web : <https://www.arduino.cc/en/Reference/LiquidCrystal> (accessed May 18, 2023)
- [28] Documentation officielle de la bibliothèque "MFRC522.h" sur le site de NXP : Site web : <https://www.nxp.com/docs/en/data-sheet/MFRC522.pdf> (accessed May 18, 2023)
- [29] Simon Monk, "Programming Arduino: Getting Started with Sketches, Second Edition". McGraw-Hill Education TAB,2015.
- [30] Michael Margolis, "Arduino Cookbook, Second Edition". O'Reilly Media, Inc,2011
- [31] Documentation officielle de la bibliothèque "SD.h" sur le site Arduino : Site web : <https://www.arduino.cc/en/Reference/SD> (accessed May 18, 2023)
- [32] Simon Monk, "Programming Arduino: Getting Started with Sketches", McGraw-Hill Education.
- [33] John Boxall, "Arduino Workshop: A Hands-On Introduction with 65 Projects", No Starch Press.
- [34] IMG Photographie personnelle prise par téléphone.
- [35] Google Images.

Résumé

L'objectif de ce projet est de mettre en place un système de pointage pour la gestion de la présence des étudiants à l'aide d'un Arduino UNO et d'un lecteur RFID-RC522. Ce système comprend différents composants tels qu'un afficheur LCD, une horloge RTC DS1307 et une carte SD.

En combinant ces composants, il est possible de créer un système de présence portable qui permet d'identifier de manière unique la fréquentation des étudiants à l'université en fonction de leur étiquette RFID. Les étudiants n'ont qu'à placer leur carte d'identité sur le lecteur et leur présence sera enregistrée instantanément.

L'horloge en temps réel du système permet d'enregistrer précisément l'heure de présence des étudiants. De plus, les données de présence seront sauvegardées sur une carte SD, offrant ainsi une solution de stockage portable et facilement accessible.

Summary

The objective of this project is to implement an attendance tracking system for student management using an Arduino UNO and an RFID-RC522 reader. This system includes various components such as an LCD display, a DS1307 RTC clock, and an SD card.

By combining these components, it is possible to create a portable attendance system that uniquely identifies students' attendance at the university based on their RFID tag. Students simply need to place their identification card on the reader, and their attendance will be instantly recorded.

The real-time clock of the system accurately records the students' time of attendance. Additionally, the attendance data will be saved on an SD card, providing a portable and easily accessible storage solution.

ملخص

الهدف من هذا المشروع هو انشاء نظام لتسجيل حضور الطلاب باستخدام أردو وبنوانو وقارئ ارفايدي هذا النظام يحتوي على عدة مكونات اهمها شاشة ال سي دي وساعة ارتي سي وبطاقة ذاكرة من خلال دمج هذه المكونات يمكننا انشاء نظام يعمل على تسجيل حضور الطلاب في الجامعة بطريقة فريدة من نوعها حيث يكفي على الطلاب ان يمرروا بطاقتهم على القارئ ليتم تسجيل حضورهم. يضمن نظام الساعة تسجيل وقت حضور الطلاب بدقة حيث يتم حفظ هذه البيانات في بطاقة الذاكرة مما يوفر حلا محمولا وسهل الوصول اليه للتخزين.