



UNIVERSITÉ CHEIKH ANTA DIOP  
FACULTÉ DES SCIENCES ET TECHNIQUES  
LABORATOIRE ALGÈBRE CRYPTOGRAPHIE GÉOMÉTRIE ALGÈBRE APPLICATION

---

# Mise en place d'une application WEB pour la gestion d'un e-parking

---

*Président du jury :*

PR. MAMADOU BARRY

*Membres du jury :*

*Soutenu par :*

AMINATA DIALLO

THIERNO MAMOUDOU SABALY

*Sous la direction de :*

Mr. HABIB NDIAYE

Année académique :2017-2018

## REMERCIEMENTS

**EN PRÉAMBULE À CE MÉMOIRE JE REMERCIE NOTRE CRÉATEUR ALLAH POUR TOUS SES BIENFAITS. JE VOUDRAIS EXPRIMER MA GRATITUDE À TOUTE MA FAMILLE QUI M'A SOUTENUE, ENCOURAGÉE ET ASSISTÉE, PLUS PARTICULIÈREMENT À MES PARENTS. PUIS, J'ADRESSE MES SINCÈRES REMERCIEMENTS À MON ENCADREUR MR HABIB NDIAYE POUR M'AVOIR GUIDÉE, ORIENTÉE, CONSEILLÉE, CRITIQUÉE AFIN DE TOUJOURS DONNER LE MEILLEUR DE MOI-MÊME. MAIS AUSSI D'AVOIR CRU EN MOI. ENFIN, JE REMERCIE MES AMIS POUR LEURS SOUTIENS INCONDITIONNELS ET LEURS ENCOURAGEMENTS. AINSI QUE TOUS CEUX QUI DE LOIN OU DE PRÈS ONT CONTRIBUÉ À LA RÉALISATION DE CE PROJET. À TOUS CES INTERVENANTS, JE PRÉSENTE MES REMERCIEMENTS, MON RESPECT ET MA GRATITUDE POUR AVOIR FAIT DE MOI CE QUE JE SUIS.**

## DÉDICACES

**JE VOUDRAIS DÉDIER CE MÉMOIRE À TOUTES LES PERSONNES QUI ME SONT CHÈRES. MAIS AUSSI À TOUS LES PASSIONNÉS DE "L'INTERNET OF THING"(IOT). CES GENS QUI, NUIT ET JOUR, M'ONT SOUTENUE ET ONT CRU EN MOI. À TOUS CEUX QUI LIRONT CE MÉMOIRE, JE VOUS SOUHAITE DE TOUT CŒUR QUE LA LECTURE DE CE MÉMOIRE SOIT UNE AUBAINE OU SOURCE D'INSPIRATION POUR VOUS.**

AMINATA DIALLO

Thierno Mamoudou Sabaly

# Table des matières

<b>1</b>	<b>Introduction</b>	<b>6</b>
<b>2</b>	<b>Étude de cas</b>	<b>7</b>
<b>3</b>	<b>Approche expérimentale</b>	<b>8</b>
3.0.1	Définition du module Arduino . . . . .	8
3.0.2	Les différentes gammes de la carte Arduino . . . . .	8
3.0.3	Le Micro contrôleur ATMega2560 . . . . .	9
3.0.4	Broches de connexion . . . . .	10
3.0.5	Les ports de communications . . . . .	10
3.1	Partie programme . . . . .	12
3.1.1	Environnement de la programmation . . . . .	12
3.1.2	Structure générale du programme (IDE Arduino) . . . . .	12
3.1.3	Injection du programme . . . . .	12
3.1.4	Description du programme . . . . .	13
3.1.5	Les étapes de téléchargement du programme . . . . .	13
3.2	Les différents composants de notre système . . . . .	15
3.2.1	Les afficheurs LCD . . . . .	15
3.2.2	RFID . . . . .	16
3.2.3	Buzzer . . . . .	16
3.2.4	Servo moteur . . . . .	17
3.2.5	Capteur d’ultra-sons . . . . .	18
3.2.6	capteur d’infrarouge . . . . .	18

# Table des figures

3.1	Schéma des types de microcontrôleur . . . . .	10
3.2	Schéma récapitulatif des différents ports de communication . . . . .	11
3.3	Structure générale du programme . . . . .	12
3.4	Initialisation de la carte . . . . .	12
3.5	Initialisation de la carte . . . . .	14
3.6	Schéma de l'écran LCD . . . . .	15
3.7	Schéma du clavier Matricielle keypad . . . . .	16
3.8	Schéma du buzzer actif . . . . .	17
3.9	Schéma du servomoteur . . . . .	17
3.10	Schéma du capteur d'ultrasons . . . . .	18
3.11	Schéma du capteur d'infrarouge . . . . .	18

# Liste des tableaux

3.1    Tableau récapitulatif des différents types d'Arduino . . . . . 9

# Chapitre 1

## Introduction

Actuellement le problème de stationnement s'accroît de plus en plus dans le quotidien de la population urbaine.

En effet, l'indiscipline de la population et l'absence de mobilité motive la plupart à stationner n'importe où n'importe comment. Dans un tel cadre la gestion des parkings constitue un véritable problème. Des lors comment assurer la gestion des parkings ?

En ce sens nous allons étudier les objectifs d'une application de gestion de e-parking, puis expliquer le fonctionnement de celle-ci et des différents composants qui la composent et enfin procéder à la réalisation de cette dernière.

# Chapitre 2

## Étude de cas

Le principe de notre application repose sur la gestion d'un e-parking. Et en ce sens lorsque l'individu veut accéder au parking un message de Bonjour l'accueil à partir de l'écran lcd. Il devra dans ce cas s'identifier grâce à son badge rfid près du lecteur. Un badge rfid est une carte dotée d'une puce magnétique qui stocke un identifiant. Et une identification se fera au niveau de la base de données afin de vérifier l'identifiant de l'individu. Et cet ainsi après comparaison de celui-ci dans la base de données au niveau des id valides trois cas se présenteront :

1. D'abord nous avons le cas où l'identifiant est dans la base de données des identifiants valide. Permettant ainsi l'accès à ce dernier au niveau du parking. Dès lors l'ordre sera donné au servomoteur qui sert de barrière électronique de se lever automatiquement pour un temps bien déterminé. Par mesure de sécurité la barrière va se remettre à sa position initiale lorsque le capteur ultrason ne va plus détecter la présence d'une voiture à une distance de 30 cm de celle-ci. L'individu pourra dans ce cas se garer au niveau des places inoccupées. Rappelons que le nombre de places du parking est délimité par des capteurs d'infrarouges qui lorsqu'une place est occupée il détecte cela et le signal au niveau de l'interface graphique.
2. Ensuite, lorsque l'identifiant se trouve dans la base de données mais dans la table des uid qui ne sont plus valides un message sera envoyé à ce dernier au niveau de l'écran lcd pour lui dire de se rendre au niveau de l'administrateur du système.
3. Et enfin dans le cas où l'identifiant n'est pas dans la base de données. Un message sera envoyé à ce dernier au niveau de l'écran lcd afin qu'il remette son badge au niveau du lecteur rfid. Ainsi après 3 tentatives d'échec un système d'alarme va se déclencher grâce au buzzers. L'identifiant devra suivre l'autre voie afin de permettre à la file d'avancer.



# Chapitre 3

## Approche expérimentale

### 3.0.1 Définition du module Arduino

Le module Arduino est un circuit imprimé en matériel libre(plateforme de contrôle) dont les plans de la carte elle-même sont publiés en licence libre. Un microcontrôleur programmé peut analyser et produire des signaux électriques de manière à effectuer des tâches très diverses. Arduino est utilisé dans beaucoup d'applications comme l'électrotechnique industrielle et embarquée, le modélisme, la domotique mais aussi dans des domaines différents comme l'art contemporain et le pilotage d'un robot, commande des moteurs, faire des jeux de lumière, communiquer avec l'ordinateur, commander des appareils mobiles (modélisme). Chaque module d'Arduino possède un régulateur de tension +5V et un oscillateur à quartz de 16MHz(ou un résonateur céramique de certains modèles). Pour programmer cette carte, on utilise le logiciel Arduino IDE.

### 3.0.2 Les différentes gammes de la carte Arduino

Carte Arduino	nom	taille	micro controleur	memoire flash	EEPROM	Alimentation
	mini	33x18x10	ATmega328	32 (2)	1	7 – 9
	mini PRO	33x18x10	Atmega168	16 (2)	0.5	3.3-12 / 5-12
	micro	50x18x18	Atmega32u4	32 (4)	1	7 – 12
	nano	44x20x18	Atmega168/328	16/32(2)	0.5/1	7 – 12
	uno	74x54x16	Atmega328	32 (0.5)	1	7 – 12
	due	102x53x16	AT91SAM3X8E	512		7 – 12
	leonardo	70x54x16	Atmega32u4	32 (4)	1	7 – 12
	Yun	72x53x17	Atmega32u4, Atheros AR9331	32 (4)	1	5
	Mega ADK	102x54x16	Atmega2560	256 (8)	4	7 – 12
	Mega 2560	107x53x15	Atmega2560	256 (8)	4	7 – 12
	Pro	53x52x10	Atmega168/328	16/32 (2)	0.5/1	3.3-12 / 5-12
	zero		ATSAMD21G18	256	16	
	ethernet		ATmega328	32(0.5)	1	7-12

TABLE 3.1 – Tableau récapitulatif des différents types d'Arduino

La mémoire EEPROM<sup>1</sup> est un type de mémoire morte. Une mémoire morte est un type de mémoire utilisée pour enregistrer des informations qui ne doivent pas être perdues lorsque l'appareil qui les contient n'est plus alimenté en électricité.

### 3.0.3 Le Micro contrôleur ATmega2560

Un microcontrôleur ATmega2560 est un circuit intégré, qui rassemble sur une puce plusieurs éléments complexes dans un espace réduit, au temps des pionniers de

1. Electrically-Erasable Programmable Read-Only Memory

l'électronique. Aujourd'hui, en soudant un grand nombre de composants encombrants tels que les transistors, les résistances et les condensateurs, tous peuvent être logés dans un petit boîtier en plastique noir muni d'un certain nombre de broches dont la programmation peut être réalisée en langage C. La figure suivante montre un microcontrôleur ATmega2560, qu'on trouve sur la carte Arduino.

FIGURE 3.1 – Schéma des types de microcontrôleur

### 3.0.4 Broches de connexion

La carte Mega2560 dispose de 16 entrées analogiques, chacune pouvant fournir une mesure d'une résolution de 10 bits(c-à-d sur 1024 niveaux soit de 0 à 1023) à l'aide de la très utile fonction **analogRead()** du langage Arduino. Par défaut, ces broches mesurent entre le 0V(valeur 0) et le 5V(valeur 1023), mais il est possible de modifier la référence supérieure de la plage de mesure en utilisant la broche AREF et l'instruction **analogReference()** du langage Arduino. Note : Les broches analogiques peuvent être utilisées en tant que broches numériques. Il y a deux autres broches disponibles sur la carte :

1. AREF : Tension de référence pour les entrées analogiques (si différent du 5V). Utilisée avec l'instruction `analogReference()`.
2. Reset : Mettre cette broche au niveau BAS entraîne la réinitialisation(le redémarrage) du micro-contrôleur. Typiquement, cette broche est utilisée pour ajouter un bouton de réinitialisation sur le circuit qui bloque celui présent sur la carte.

### 3.0.5 Les ports de communications

La carte Arduino Mega2560 dispose de toute une série de facilités pour communiquer avec un ordinateur, une autre carte Arduino, ou avec d'autres microcontrôleurs. L'ATmega2560 dispose de quatre UARTs <sup>2</sup>(émetteur-récepteur asynchrone universel en français) pour la communication série de niveau TTL(5V) et qui est disponible sur les broches 0(RX) et 1(TX). Un circuit intégré ATmega8U2 sur la carte assure la connexion entre cette communication série de l'un des ports série de l'ATmega 2560 vers le port USB de l'ordinateur qui apparaît comme un port COM virtuel pour les logiciels de l'ordinateur. Le code utilisé pour programmer l'ATmega8U2 utilise le driver standard USB COM, et aucun autre driver externe n'est nécessaire. Cependant, sous Windows, un fichier .inf est requis. Le logiciel Arduino inclut une fenêtre terminal série(ou moniteur série) sur l'ordinateur qui permet d'envoyer des textes simples depuis et vers la carte Arduino. Les LEDs RX et TX sur la carte

---

2. Universal Asynchronous Receiver Transmitter

clignotent lorsque les données sont transmises via le circuit intégré ATmega8U2 utilisé en convertisseur USB-vers-série et la connexion USB vers l'ordinateur(mais pas pour les communications série sur les broches 0 et 1). Une librairie Série Logicielle permet également la communication série(limitée cependant) sur n'importe quelle broche numérique de la carte UNO. L'ATmega2560 supporte également la communication par protocole I2C (ou interface TWI <sup>3</sup>( Interface "2 fils") et SPI :

1. Le logiciel Arduino inclut la librairie Wire qui simplifie l'utilisation du bus I2C. <sup>4</sup>
2. Pour utiliser la communication SPI(Interface Série Périphérique), la librairie est disponible.

FIGURE 3.2 – Schéma récapitulatif des différents ports de communication

---

3. Two Wire Interface

4. Voir la documentation pour les détails

## 3.1 Partie programme

Une telle carte d'acquisition qui se base sur la construction d'un microcontrôleur doit être dotée d'une interface de programmation. L'environnement de programmation open-source pour Arduino peut être téléchargé gratuitement (pour Mac OS X, Windows, et Linux).

### 3.1.1 Environnement de la programmation

FIGURE 3.3 – Structure générale du programme

Le logiciel de programmation de la carte Arduino sert d'éditeur de code (Langage proche du C). Une fois, le programme tapé ou modifié au clavier, il sera transféré et mémorisé dans la carte à travers de la liaison USB. Le câble USB alimente à la fois en énergie la carte et transporte aussi l'information.

### 3.1.2 Structure générale du programme (IDE Arduino)

Comme la plupart des langages de programmation, le code source est exécutable sur n'importe quel système d'exploitation basé sur la programmation en C. Les fonctions Setup et Loop sont obligatoires pour la création d'un programme en arduino. La fonction setup n'est exécutée qu'une seule fois, elle initialise et fixe les valeurs de démarrage du programme. La fonction Loop quant à elle s'exécute sans fin elle peut aussi permettre de contrôler la carte arduino

### 3.1.3 Injection du programme

Avant d'envoyer un programme dans la carte, il est nécessaire de sélectionner le type de la carte (Arduino UNO) et le numéro de port USB (COM3) comme à titre d'exemple cette figure suivante.

FIGURE 3.4 – Initialisation de la carte

### 3.1.4 Description du programme

Un programme arduino est une suite d'instructions élémentaires sous forme textuelle (ligne par ligne). La carte lit puis effectue les instructions les unes après les autres dans l'ordre défini par les lignes de code.

**Les commentaires** Les commentaires sont, en programmation informatique, des portions du code source ignorées par le compilateur ou l'interpréteur car ils ne sont pas censés influencer l'exécution du programme.

**Définition des variables** Pour notre montage, on va utiliser une sortie numérique de la carte qui est par exemple la 3e sortie numérique. Cette variable, doit être définie et nommée ici buzzer pin 3 ; la syntaxe est pour désigner un nombre entier est int.

Une programmation des interactions void loop : dans cette boucle, on définit les opérations à effectuer dans l'ordre digital write nom, état) est une autre des quatre fonctions relatives aux entrées – sorties numériques.

delay(temps en mili-seconde) est la commande d'attente entre deux instructions.

chaque ligne d'instruction est terminée par un point-virgule. Ne pas oublier les accolades qui encadrent la boucle.

```
void loop()
```

```
-----  
10 digital write ( moteur 1,HIGH); -----  
11delay(3000)-----  
12 digital Write(buzzer 1, LOW); -----  
13 delay (1000)-----  
  
14 -----
```

### 3.1.5 Les étapes de téléchargement du programme

1. On conçoit ou on ouvre un programme existant avec le logiciel IDE Arduino.
2. On vérifie ce programme avec le logiciel Arduino(compilation).
3. Si des erreurs sont signalées, on modifie le programme.
4. On téléverse le programme sur la carte.
5. On câble le montage électronique.
6. L'exécution du programme est automatique après quelques secondes.

7. On alimente la carte soit par le port USB, soit par une source d'alimentation autonome (pile 9 volts par exemple).
8. On vérifie que notre montage fonctionne.

FIGURE 3.5 – Initialisation de la carte

## 3.2 Les différents composants de notre système

La carte Arduino est généralement associée aux accessoires qui simplifient les réalisations.

### 3.2.1 Les afficheurs LCD

Les écrans LCD existent depuis 1971. Ils n'ont pas cessé de se développer depuis, et équipent maintenant bien des appareils à affichage embarqué (appareils photo, digicodes, montres, téléphones). LCD est l'abréviation anglaise de "liquid crystal display" qui veut dire : afficheur à cristaux liquides. Cette technologie permet de créer des écrans plats qui consomment peu d'énergie. Parmi les écrans lcd nous pouvons citer l'afficheur I2C, les afficheur 1\*16, 20\*4, le shield keypad LCD, etc... Nous nous intéresserons à l'écran LCD 16\*2. L'écran LCD 16\*2 nous permettra d'afficher les données saisies par l'utilisateur.

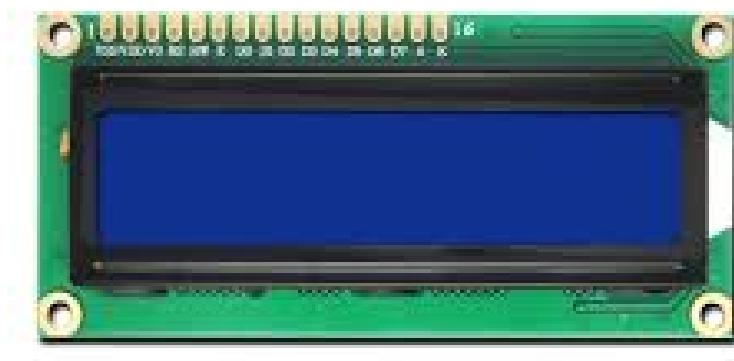


FIGURE 3.6 – Schéma de l'écran LCD



### 3.2.2 RFID

RFID, qui signifie identification par radio fréquence, est une méthode permettant de récupérer et traiter des données à distance. Le système est activé par un transfert d'énergie électromagnétique entre une étiquette radio et un émetteur RFID. L'étiquette radio, appelée tag, est composée d'une puce électronique et d'une antenne qui reçoit le signal émis par le lecteur équipé d'une technologie RFID. Les composants permettant de lire et répondre aux signaux La RFID est caractérisé par la Fréquence. Celle-ci permet d'établir la communication entre la puce et l'antenne. Il existe ainsi différentes puces, leur différence réside sur la fréquence de fonctionnement et la distance de lecture. Plus la fréquence est élevée, plus la distance de lecture s'agrandit. Trois types de fréquence sont utilisés par les puces RFID : Basse fréquence (125KHZ), Haute fréquence (13,56MHZ) et Très haute fréquence(UHF) :

1. RFID passive : C'est un ensemble de puce fonctionnant en lecture seule. N'ayant pas de batterie, elles se déplacent vers les lecteurs pour être activées par un signal électromagnétique. Cette alimentation permettra la lecture des données qu'il contient
2. RFID active : Contrairement à la précédente, cette technologie permet une lecture à longue distance des puces. Et ceci, grâce à la source d'énergie (pile ou batterie) qu'utilise ces puces.
3. RFID semi-passive : ça serait une sorte d'intermédiaire évidemment, ces puces possèdent des sources d'énergies mais leur alimente que par intervalle de temps.



FIGURE 3.7 – Schéma du clavier Matricielle keypad

### 3.2.3 Buzzer

Les buzzers sont des périphériques qui nous permettent d'émettre un signal sonore à partir d'une tonalité. Il existe deux types de buzzers à savoir le buzzer actif et le buzzer passif. Un buzzer actif va générer une tonalité à l'aide d'un oscillateur interne de sorte que tout ce qui est nécessaire est une tension CC. Un buzzer passif nécessite un signal CA pour faire un son. C'est comme un haut-parleur électromagnétique où un signal d'entrée changeant, plutôt que de produire une tonalité automatiquement. La méthode pour les identifier est la suivante : si vous appliquez une tension CC il vibre, alors nous avons un buzzer actif sinon un buzzer passif. En ce qui concerne les commandes nous allons contrôler le pitch. PWM sur l'Arduino peut être utilisé pour contrôler le pitch et le volume en même temps (qui peut ou peut ne pas être ce que vous voulez). Si nous voulons changer juste le volume ou juste le pitch je suppose que certains circuits externes seraient nécessaires pour changer l'amplitude sans

changer la tension, et vice versa. Ainsi ce buzzer va nous fournir la tonalité nécessaire afin de déclencher l'alarme.



FIGURE 3.8 – Schéma du buzzer actif

### 3.2.4 Servo moteur

Un servomoteur de modélisme se présente sous la forme d'un petit rectangle avec deux languettes sur les côtés pour la fixation et un axe décentré avec un bras(interchangeable) pour la liaison mécanique. Le servomoteur est un moteur esclave. Esclave parce que l'on lui ordonne et il obéit. On peut remarquer de petit gros servomoteurs, ils sont essentiellement constitués d'une boîte, d'un axe et d'une roue (ou barre) trouée. Il y a des servomoteur hydrauliques, mais nous utiliserons de ceux électriques dans le cadre de ce projet. Un servomoteur est constitué d'un moteur à courant continu. Il tourne lorsqu'il parcouru par de l'électricité. Ce moteur est associé à une série d'engrenages. Une série d'engrenage est un ensemble de roue dentées qui s'entraînent les unes des autres lors de leur rotation, on par le de train d'engrenage. Ce train met en rotation l'axe apparent à l'extérieur du moteur. Un servomoteur est caractérisé par un couple (kg.cm) exprimant la masse qu'il peut entraîner à une certaine distance de son axe. Ce couple est définit comme étant la force nécessaire pour faire tourner l'axe.



FIGURE 3.9 – Schéma du servomoteur

### 3.2.5 Capteur d'ultra-sons



FIGURE 3.10 – Schéma du capteur d'ultrasons

### 3.2.6 capteur d'infrarouge



FIGURE 3.11 – Schéma du capteur d'infrarouge