



الجمهورية الجزائرية الديمقراطية الشعبية
République Algérienne Démocratique et Populaire
وزارة التعليم العالي والبحث العلمي
Ministère de l'Enseignement Supérieure et de la Recherche Scientifique

جامعة وهران 2 محمد بن أحمد
Université d'Oran 2 Mohamed Ben Ahmed

معهد الصيانة و الأمن الصناعي
Institut de Maintenance et de Sécurité Industrielle

Département de Maintenance en Instrumentation

MÉMOIRE

Pour l'obtention du diplôme de Master

Filière : Génie Industriel

Spécialité : Génie Industrielle

Thème

Vision artificielle : Application à la conception d'un parking intelligent

Présenté et soutenu publiquement par :

DENDEN NOUR EL HOUDA

et

DEROUICHE OUSSAMA

Devant le jury composé de :

Nom et Prénom	Grade	Etablissement	Qualité
REGUIEG YSSAAD Sadek	MAA	IMSI-Univ. D'Oran2	Président
BELKHODJA LEILA	MCB	IMSI-Univ. D'Oran2	Encadreur
AISSANI Nassima	MCA	IMSI-Univ. D'Oran2	Examineur

Année 2020/2021

Résumé

La population urbaine en croissance rapide dans tout le monde crée de nombreux problèmes pour les villes, le stationnement des voitures étant l'un des problèmes majeurs. Le nombre de voitures augmente chaque jour de manière rapide, cela cause le problème de l'encombrement.

Le parking conventionnel est un des facteurs les plus importants pour régler cette situation, cependant il contient beaucoup des problèmes dont l'insécurité, la perte de temps, la pollution etc...

Les villes ont fortement besoin de systèmes de stationnement de voitures avancés. Dans ce sens, de nombreuses nouvelles technologies ont été développées pour aider à résoudre ces problèmes.

Dans ce mémoire, nous proposons un parking intelligent qui regroupe les technologies de l'électronique, de l'automatique de l'informatique et de la vision artificielle. Le parking intelligent joue un rôle majeur dans la création d'un meilleur environnement urbain en réduisant les émissions de CO2 et autres polluants.

Le but de ce travail est de réaliser un prototype d'un parking intelligent privé basant sur un accès pour les personnes normales et un accès pour les personnes handicapées par une technologie d'identification par radiofréquence (RFID). Ces scénarios seront automatisés via une carte "ArduinoUno" et logiciel « Arduino IDE » et simulés par logiciel « Fritzing ».

Mots clés :

Parking intelligent, ville intelligente, Internet des objets, ArduinoUno, Identification par radiofréquence (RFID), Vision artificielle

Abstract

The rapidly growing urban population all over the world creates many problems for cities, with parking of cars being one of the major problems. The number of cars is increasing rapidly everyday, this causes the congestion problem. The parking among the most important factors to solve this situation, the conventional parking lots contain many problems, large areas occupied, poor security, waste of time ...etc.

Cities have a great need for advanced car parking systems, to avoid these problems, many new technologies have been developed to help solve these problems.

In this thesis, we propose an intelligent parking lot that brings together the technologies of electronics, automation and IT. Smart parking plays a major role in creating a better urban environment by reducing CO2 emissions and other pollutants.

The aim of this work is to produce a prototype of a private smart car park based on Access for normal people and Access for people with disabilities by radio frequency identification technology (RFID), these scenarios will be automated via a card " Arduino Uno " and « Arduino IDE » software and simulated by « Fritzing » software.

Keywords :

Smart Parking, Smart City, Internet of Things, ArduinoUno, Radio frequency identification (RFID)

Remerciements

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

Nous tenons à remercier sincèrement Dr. BELKHODJA Leila, qui, en tant qu'encadrant, s'est toujours montrée à l'écoute et disponible tout au long de la réalisation de ce travail.

Nos remerciements vont également aux membres du jury pour nous avoir honoré de leurs appréciations pour ce travail.

Nous remercions également nos enseignants de l'Institut de Maintenance et de Sécurité Industrielle, en particulier ceux du Département de Maintenance en Instrumentation, de la Filière *Génie Industrielle*, pour leurs aides et leurs encouragements.

Enfin, nous remercions tous ceux qui nous ont soutenu, encouragé et donné envie de mener à bien ce travail.

Merci à tous et à toutes.

Dédicaces

Du profond de mon cœur, je dédie ce travail à tous ceux qui me sont chers,

A MA CHERE MERE

Aucune dédicace ne saurait exprimer mon respect, mon amour éternel et ma
Considération pour les sacrifices que vous avez consentis pour mon instruction et mon bien
être.

Je vous remercie pour tout le soutien et l'amour que vous me portez depuis mon enfance et

J'espère que votre bénédiction m'accompagne toujours.

Que ce modeste travail soit l'exaucement de vos vœux tant formulés, le fruit de vos
innombrables sacrifices.

Puisse Dieu, le tout Puissant, vous accorder santé, bonheur et longue vie.

A TOUS MES AMIS

En témoignage de l'amitié sincère qui nous a liées et des bons moments passés ensemble.

A tous ceux qui ont contribué de près ou de loin, pour la réussite de ce projet, veuillez trouver
ici le respect et la reconnaissance que j'éprouve pour vous.

Denden Nour El Houda

Dédicaces

Tous les mots ne sauraient exprimer la gratitude, le respect,
la reconnaissance...

Je dédie ce travail avec ma sincère gratitude
et mon respect

À

Mes chers parents, Aucun mot ne peut exprimer mes

Respects, mon amour éternel et ma considération

Pour les sacrifices que vous avez faits pour moi, je

Vous remercie pour tout le soutien et l'amour que

Vous m'avez apporté.

À

Mes chers frères et chères sœurs.

À

Tous mes proches et amis, qui m'ont toujours soutenu
et encouragé.

Derouiche Oussama

Sommaire

Liste des abréviations	I
Liste des tableaux	II
Liste des figures	III
Introduction générale	2
Chapitre I : Généralités	3
I.1. Introduction	4
I.2. Problématique de stationnement	4
I.3. Quelques définitions	5
I.3.1. Définition l'automobile	5
I.3.2. Internet des objets	5
I.3.3. La mobilité intelligente	6
I.3.4. Le stationnement	7
I.3.5. Définition du parking	8
I.3.6. Stationnements intelligents	8
I.4. Historique du parking	9
I.5. Les différents types de stationnement	10
I.6. Les différents types de parking	12
I.7. Les points forts du stationnement intelligent	14
I.8. Conclusion	15
Chapitre II : Matériels et méthodes	16
II.1. Introduction	17
II.2. Vision artificielle	17
II.3. La partie matérielle	19
II.3.1. La carte Arduino UNO	19
II.3.1.1. Présentation de la carte	19
II.3.1.2. Caractéristique de la carte Arduino UNO	20
II.3.1.3. Les différents composants de la carte Arduino UNO	21
II.3.2. Module RFID RC522	25
II.3.3. TCRT5000 Module de capteur infrarouge	28
II.3.4. Servomoteur	29
II.3.5. Module relais 5v	32
II.3.6. Afficheur LCD 16x2	32
II.3.7. Module I2C LCD, PCF88574	34
II.3.8. Buzzer	35
II.3.9. Capteur ultrason (HC-SR04)	36
II.3.10. Les composants	37
II.3.10.1. Breadboard (platine de montage)	37
II.3.10.2. Les fils de liaison	38
II.3.10.3. Résistances (R)	38
II.3.10.4. Diode LED	38
II.4. La partie logicielle	39
II.4.1. Le logiciel Arduino	39

II.4.1.1. Présentation de « IDE »	39
II.4.1.2. Programmer avec Arduino	40
II.4.1.3. Structure d'un programme	40
II.4.2. Le logiciel Fritzing	42
II.6. Conclusion	43
Chapitre III : Conception et réalisation	44
III.1.Introduction	45
III.2. Problématique	45
III.3. Structure du système du parking proposé	45
III.4. Conception Algorithmique	48
III.5. Principe de fonctionnement du système	49
III.5.1. Détection des véhicules à la barrière (RFID/Caméra)	49
III.5.2. Détection des véhicules à l'intérieur du parking	49
III.5.3. Système de détection avec la technologie RFID	49
III.5.4. Système de comptage.....	49
III.4. Matériel	50
III.6. Quelques simulations virtuelles des composants sous Fritzing.....	56
III.7. Présentation des programmes sur IDE.....	60
III.8. Système sur plaque d'essai.....	63
III.9. Implémentation sur maquette finale	64
III.10. Jeu d'essai.....	66
III.11. Conclusion.....	71
Conclusion générale	73
Bibliographie.....	74

Liste des abréviations

IoT: Internet of Things

IP: Internet Protocol

RFID: Radio Frequency Identification

USB: Universal Serial BUS

IDE: Integrated Development Environment.

EEPROM: Electrically erasable programmable read-only memory

LED: Light Emitting Diode

LCD: liquid crystal display

GND: GROUND

NXP : l'entreprise NXP SEMICONDUCTORS FRANCE.

VCC : Voltage Collector Collector

PWM : Pulse Width Modulation

PCB : Printed Circuit Board

Liste des tableaux

Tableau II.1 : Les Caractéristiques de la carte Arduino Uno	20
Tableau II.2 : Les Caractéristiques d'un Microcontrôleur ATmega328	22
Tableau II.3 : Les Caractéristiques du buzzer	35
Tableau II.4 : Les Caractéristiques d'un Capteur ultrason HC-SR04	36
Tableau III.1 : Ensemble des composants	50

Liste des figures

Figure I.1 : Stationnement intelligent.....	4
Figure I.2 : L'automobile	5
Figure I.3 : Des exemples sur l'internet des objets.....	6
Figure I.4 : La mobilité intelligente	7
Figure I.5 : Le stationnement des véhicules sur le bord de la route.....	7
Figure I.6 : Exemple de parking	8
Figure I.7 : Le stationnement intelligent des véhicules.....	9
Figure I.8 : le parking dans les années 50	10
Figure II.1 : La conversion de l'écriture manuscrite en texte dactylographié par la ROC	18
Figure II.2 : La Voiture autonome détectant les véhicules autour d'elle	18
Figure II.3 : Concept de reconnaissance faciale	19
Figure II.4 : La carte Arduino UNO.....	20
Figure II.5 : Les composants de la carte Arduino UNO	21
Figure II.6 : Microcontrôleur ATmega328	21
Figure II.7 : Datasheet d'ATmega328	23
Figure II.8 : Constitution de la carte Arduino UNO	25
Figure II.9 : Module RFID RC522	26
Figure II.10 : Utilisation d'un lecteur RFID avec Arduino UNO.....	27
Figure II.11 : Carte RFID	28
Figure II.12 : TCRT5000 Module de capteur infrarouge.....	28
Figure II.13 : TCRT5000 Module de capteur infrarouge de proximité pour détection d'obstacle.....	29
Figure II.14 : Vue d'un servomoteur	30
Figure II.15 : Un servomoteur avec une carte Arduino Uno.....	31
Figure II.16 : l'angle de l'axe du servomoteur en fonction de la largeur de l'impulsion	31
Figure II.17 : Vue d'un relais 5v	32
Figure II.18 : Un afficheur LCD 16x2	33
Figure II.19 : Schéma d'un afficheur LCD 16x2 avec Aduino Uno.....	34
Figure II.20 : Module convertisseur I2C LCD	35
Figure II.21 : Vue d'un Buzzer.....	35
Figure II.22 : Buzzer avec une carte Arduino Uno.....	36
Figure II.23 : Module HC-SR04 capteur ultrason	37
Figure II.24 : Breadboard	37
Figure II.25 : Les fils de liaison.....	38
Figure II.26 : Les résistances	38
Figure II.27 : Diode LED	39
Figure II.28 : L'interface visuelle du logiciel Arduino	40
Figure II.29 : L'exemple de structure de programme Arduino	41
Figure II.30 : L'interface de Fritzing	42
Figure III.1 : Schéma bloc de notre système	46
Figure III.2 : Structure schématique du parking intelligent proposé.....	47
Figure III.3 : La conception algorithmique de notre parking	48
Figure III.4 : Exemple de caméra de détection de plaques d'immatriculation	49
Figure III.5 : Les deux cartes Arduino UNO	51
Figure III.6 : Lecteur RFID RC522	51
Figure III.7 : Les cartes RFID	51
Figure III.8 : Capteur infrarouge TCRT5000.....	52

Figure III.9 : Servomoteur.....	52
Figure III.10 : Relais 5v	52
Figure III.11 : Afficheur LCD 16x2 avec I2C.....	53
Figure III.12 : Capteur ultrason HC-SR04.....	53
Figure III.13 : Buzzer.....	53
Figure III.14 : Les LED.....	54
Figure III.15 : Résistance	54
Figure III.16 : Un Breadboard.....	55
Figure III.17 : fils de liaison.....	55
Figure III.18 : Simulation d'un Servo Moteur	56
Figure III.19 : Simulation d'une Diode LED	56
Figure III.20 : Simulation d'un Buzzer.....	57
Figure III.21 : Simulation d'un Lecteur RFID	57
Figure III.22 : Simulation d'un capteur ultrason HC-SR04.....	58
Figure III.23 : Simulation d'un afficheur LCD avec I2C	58
Figure III.24 : Simulation d'un capteur infrarouge TCRC5000.....	59
Figure III.25 : Simulation d'un système compteur.....	59
Figure III.26 : Simulation d'un système Accès RFID	60
Figure III.27 : Sous-programme de l'afficheur LCD 16x2 avec I2C	60
Figure III.28 : Sous-programme du lecteur RFID	61
Figure III.29 : Sous-programme du servo-moteur.....	62
Figure III.30 : Sous-programme de l'Arduino Uno master	62
Figure III.31 : Sous-programme de l'Arduino Uno slave.....	63
Figure III.32 : Circuit sur plaque d'essai	63
Figure III.33 : Plancher de carton.....	64
Figure III.34 : Semi plan.....	64
Figure III.35 : Parking semi fini	65
Figure III.36 : Le prototype final de notre parking intelligent	65
Figure III.37 : Le véhicule devant la barrière 1 principale	66
Figure III.38 : La validation de la carte RFID 1 et l'entrée du véhicule.....	67
Figure III.39 : La barrière principale 1 descendre	67
Figure III.40 : Le stationnement du véhicule normal	68
Figure III.41 : La validation de la carte RFID 2 et l'entrée du véhicule.....	69
Figure III.42 : Le stationnement véhicule handicapé	69
Figure III.43 : Le véhicule devant la barrière principal ou secondaire.....	70

Introduction générale

Introduction générale

La population urbaine, en croissance rapide, crée de nombreux problèmes pour les villes. Par conséquent, le nombre de véhicules connaît une forte augmentation chaque jour dans le monde, ce qui cause le problème de la congestion. Pour régler cette situation, l'installation de parkings devient un facteur primordial.

Le parking de voitures est un sujet qui a toujours été très présent parce qu'il touche la mobilité de la plupart des personnes. Les conducteurs ont souvent de réels problèmes pour facilement trouver une place de stationnement car la demande est généralement beaucoup plus importante que l'offre.

Les parkings conventionnels représentent beaucoup d'inconvénients comme le manque de places de stationnement, le problème de sécurité et la perte de temps. Il a été calculé, dans les grandes villes comme Sydney, New York ou London, que le temps moyen pour trouver une place se situe entre 3.5 et 14 minutes, ce qui engendre un problème supplémentaire de pollution dans la ville lors de la recherche d'une place de stationnement. De plus, les moyens de paiement ne sont pas toujours pratiques. Il y a aussi d'autres problèmes plus spécifiques comme par exemple retrouver dans un parking souterrain, parking sombre, espace très vaste, etc...

L'Algérie, comme tous les pays du monde, connaît aussi ce phénomène, plus particulièrement dans les grandes villes. Une des solutions que nous proposons est de faire usage de parkings intelligents, appelés aussi Smart. Beaucoup de systèmes existent déjà dans les pays riches comme les Etats Unis, l'Allemagne et le Japon, mais ils sont très coûteux pour être les mettre en œuvre ici en Algérie.

Dans ce projet de fin d'études, nous proposons une solution de smart parking, pouvant minimiser les problèmes qui résultent du stationnement classique comme la perte de temps, le stress du conducteur, la pollution, etc... . Pour cela, nous allons procéder à la conception et à la réalisation d'un prototype de parking intelligent paramétrés par des capteurs, des démarches algorithmiques et des programmes d'automatisation de la tâche du stationnement. Nous organisons ce travail en trois chapitres comme suit :

- Le premier chapitre qui consiste à donner un aperçu sur le parking intelligent, sa définition, ses types, ses avantages et ses inconvénients.
- Le deuxième chapitre est consacré à la description des différents logiciels et matériels utilisés dans notre réalisation
- Le troisième chapitre comprend la partie pratique de notre travail, où nous présentons les logiciels de simulation utilisés à savoir IDE Arduino et Fritzing. Nous avons mentionné aussi les composants de base que nous avons utilisés lors de la réalisation d'une maquette de parking capteurs et actionneurs.

Enfin, on termine notre travail par une conclusion générale et perspective.

Chapitre I

Généralités

I. Généralités

I.1. Introduction :

L'augmentation importante du nombre de voitures qui circule dans la route surtout en ville entraîne des embouteillages. Beaucoup de voitures sont mal garées, les parkings sont mal gérés. Dans ce chapitre, nous introduisons le contexte en présentant l'une des solutions possibles pour réduire les problèmes de stationnement et le temps de recherche d'un espace libre.

Le stationnement intelligent (smart parking) est une passerelle importante vers une ville intelligente. Ceci aide à résoudre beaucoup de problèmes et besoins tels que la facilité de mouvement et d'accès au lieu de destination du conducteur en lui optimisant la recherche d'espaces libres et participant ainsi à minimiser la pollution.



Figure I.1: Stationnement intelligent

I.2. Problématique de stationnement :

Depuis la seconde moitié du 20^e siècle, les villes sont aménagées en fonction de plusieurs facteurs dont l'automobile. La ségrégation des activités et la faible densité du cadre bâti ont accru les besoins de déplacement en automobile et, par conséquent, la demande en stationnement. Rapidement, le stationnement sur rue n'a plus suffi à combler cette demande. Les municipalités ont alors adopté le principe selon lequel chaque activité doit assumer les besoins de stationnement qu'elle crée, par le biais de normes minimales de stationnement pour chaque nouvelle construction surtout dans les quartiers les plus intéressants qui connaissent toujours une forte demande de stationnement.

Il faut savoir aussi que le stationnement n'est pas seulement destiné pour les gens du quartier ou de la ville mais pour tout le monde, et la durée de l'arrêt n'est pas spécifique, ce qui peut

souvent provoquer des problèmes entre les propriétaires des véhicules. C'est ainsi que le stationnement intelligent est venu pour apporter une solution à ces problèmes.

I.3. Quelques définitions :

Dans ce qui suit nous présentons quelques définitions liées au smart parking :

I.3.1. Définition l'automobile :

Une automobile est un véhicule à roues, motorisé et destiné au transport terrestre de quelques personnes et de leurs bagages. L'abréviation populaire « voiture » est assez courante, bien que ce terme désigne de nombreux types de véhicules qui ne sont pas tous motorisés (figure 1.2).



Figure I.2: L'automobile

I.3.2. Internet des objets :

L'Internet des objets est un paradigme assez vaste et admet des acceptions multiples selon les acteurs et les environnements technologiques considérés (figure 1.3). Schématiquement, l'internet des objets est une extension de l'Internet actuelle à tous les objets pouvant communiquer de manière directe ou indirecte avec des équipements électroniques eux-mêmes connectés à Internet. L'utilisation des radios étiquettes (puces RFID) permet, entre autres, une telle extension. Associées à un objet, ces puces peuvent être lues par un dispositif (mobile ou non) et des informations relatives à l'objet seront retrouvées grâce à l'Internet [1].



Figure I.4 : La mobilité intelligente

Des panneaux d'implantation dynamiques sont utilisés pour guider les utilisateurs vers des aires de stationnement gratuites. Une application logicielle unique, disponible sur les Smartphone qui informe les utilisateurs sur les lieux disponibles, l'état de la circulation et les horaires des transports en commun, calcule le chemin en fonction du meilleur rapport coût / temps, réserve si nécessaire les moyens de transport en libre-service. Finalement, la mobilité intelligente semble se construire au fur et à mesure des usages et des innovations mais cela est loin de la freiner dans son développement [3].

I.3.4. Le stationnement :

Le stationnement consiste à laisser un véhicule immobilisé un certain temps afin qu'il se trouve dans un endroit pour s'arrêter soit avec paiement ou gratuitement. Le véhicule doit être garé dans un endroit spécial pour le stationnement afin qu'il ne perturbe pas la route des habitués et ne leur cause aucun problème.



Figure I.5: Le stationnement des véhicules sur le bord de la route

I.3.5. Définition du parking :

Un parking, appelé aussi parc de stationnement ou aire de stationnement, est un espace ou un bâtiment spécifiquement aménagé pour le stationnement des véhicules (figure 1.6). Il peut être public ou privé, en enclos, en élévation ou souterrain. On en trouve le plus souvent à côté des bâtiments publics (gare, aéroport), des lieux de travail, des centres commerciaux ou devant les grandes surfaces pour accueillir les usagers.

Le symbole signalétique presque universel signifiant « parking » est un « P » majuscule blanc sur un disque ou un carré bleu.

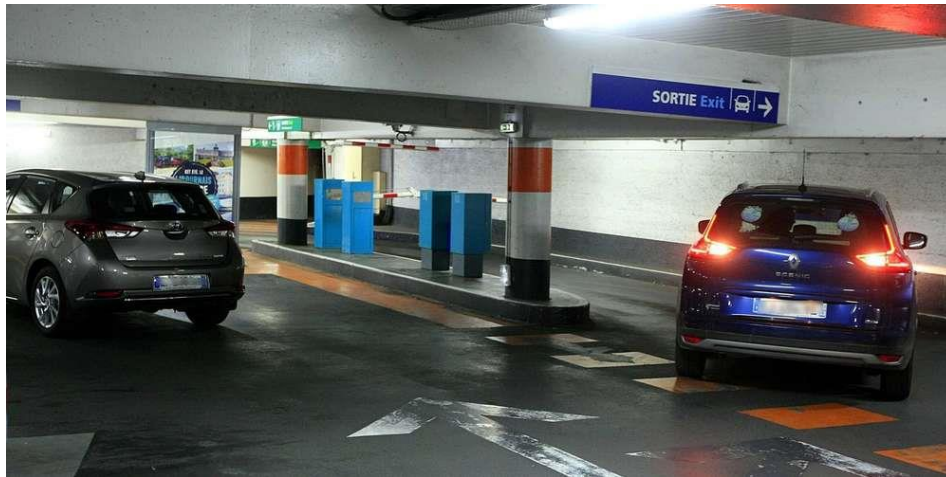


Figure I.6 : Exemple de parking

I.3.6. Stationnement intelligent :

Le stationnement intelligent, ou le Smart Parking, est une application de technologies modernes qui permet d'éliminer beaucoup de besoins en peu de temps, et ceci est limité à la facilité de circulation et à la sécurité routière, aux places de réservation et aux différents moyens de paiement. Le principe consiste à équiper chaque place de stationnement d'un capteur intelligent capable de détecter la présence d'un véhicule et d'informer en temps réel que la place est libre ou occupée. Le capteur est complètement autonome et ne nécessite donc aucune infrastructure à proximité, ce qui réduit les coûts d'investissement et surtout de maintenance. Il s'installe directement dans la chaussée, au centre de chaque place de stationnement, en moins de 10 minutes. Parmi ces technologies, vous pouvez réserver et payer de différentes façons, de manière intelligente et parmi ces méthodes, cartes à puce (smart cards), cartes sans contact, cartes de crédit/débit...etc. ces technologies rendent le stationnement plus attrayant et populaire, encourageant le développement et le paiement



Figure I.7 : Le stationnement intelligent des véhicules

I.4. Historique du parking :

Les premiers garages : datent de l'avant-première guerre mondiale en Europe et aux Etats-Unis comme par exemple celui de la rue Ponthieu à Paris édifié par Auguste Perret (1905). Durant les années 20 les projets d'infrastructures routières magnifient la voiture dans son rapport à la ville. Les parkings sur plusieurs niveaux sont généralement dérivés d'autres immeubles avec leurs façades décorées ou vitrées et des espaces parfois même chauffés.

La crise économique de 1929 et la seconde guerre mondiale entraînent une baisse du prix des terrains et incitent au développement des parkings de plein pied. Mais c'est surtout dans les années 50 avec la baisse du prix des automobiles que les infrastructures routières prolifèrent sur tout le territoire aux Etats-Unis comme en Europe. La voiture devient dès lors un outil indispensable. Les projets « tout automobile » se succédèrent comme ce fut le cas notamment des centres commerciaux en périphérie des villes, aussi appelés « centre de parking » aux Etats-Unis, déroulant d'immenses nappes de goudron pour le stationnement. Ces projets eurent un impact énorme sur les modes de vies et la configuration des villes.



Figure I.8 : le parking dans les années 50

C'est à cette même époque que les parkings à étages acquièrent leurs propres identités avec leurs façades ouvertes et leurs rampes en béton. Citons par exemple le « parking à étage en plein air » de Richard Neutra (1940 USA). Ce développement intense fut rapidement freiné par l'augmentation du prix du baril de pétrole provoqué par la crise économique de 1973. Ajoutons à cela la manifestation des premières catastrophes écologiques liées au pétrole qui remirent en cause l'utilisation de l'automobile.

Les années 90 sont marquées par le retour au parking à étage pour des raisons d'économie d'espace ainsi que la mise en place de « parkings relais » permettant aux populations périphériques de déposer leur véhicule hors des centres ville et utiliser les transports en commun. Cette idée avait été développée dans les années 50 par Louis Kahn qui proposait pour le centre-ville de Philadelphie, un projet de piétons, protégé de l'automobile par un collier de tours cylindriques réservées aux parkings.

Aujourd'hui, les parcs de stationnement sont construits majoritairement sous les édifices dans un souci d'économie d'espace mais aussi parce que l'automobile a désormais changé de statut [4].

I.5 Les différents types de stationnement :

a) Stationnement public

Il est convenu d'appeler stationnement public, tout stationnement, souvent hors-rue, accessible à tous sans réserve. Les stationnements loués à des individus sont considérés publics lorsqu'ils peuvent être loués à quiconque au moment de refaire le contrat de location

b) Stationnement privé :

Il est convenu de catégoriser comme stationnement privé, les espaces de stationnement qui sont réservés (ou interdits) à une catégorie spécifique d'usagers (employés, résidents, propriétaires, etc.).

c) Stationnement payant :

Un stationnement est dit payant, lorsque l'acte de stationnement est autorisé moyennant une charge monétaire. Le montant exigé peut être fixe ou peut varier selon : le temps de stationnement, le moment de la journée ou la saison.

d) Stationnement gratuit :

Un stationnement est dit gratuit, s'il n'exige pas un paiement à la suite d'un acte de stationnement

e) Stationnement pour personnes handicapées :

Un espace de stationnement est dit pour personnes handicapées, s'il est exclusivement réservé pour des personnes à mobilité réduite. Généralement, il respecte les géométries exigées par les normes en vigueur.

f) Stationnement incitatif :

Un stationnement incitatif est un espace de stationnement pour automobiles, généralement située périphérie d'une agglomération et qui a pour but d'inciter les automobilistes à accéder à leurs centres d'intérêt en transport en commun. Il peut être intérieur ou extérieur, payant ou gratuit.

g) Stationnement de longue durée :

Un stationnement est dit de longue durée, lorsqu'il est conçu pour un roulement plus lent (périodes de stationnement plus longues). Il est généralement présent dans des endroits où l'on trouve des voyageurs (aéroports, gares, ports...).

h) Stationnement de courte durée :

Un stationnement de courte durée est prévu pour un roulement rapide (période de stationnement plus court). Il est généralement présent sur rue dans les centres d'affaires.

i) Stationnement temporaire :

Un espace de stationnement est dit temporaire, lorsqu'il est créé afin de corriger ou d'accommoder, pour une durée déterminée, une contrainte temporaire au stationnement (chantier, inondation...).

j) Stationnement hors sur rue :

Un espace de stationnement hors rue est un espace de stationnement qui n'est pas situé le long d'une voie de circulation automobile.

k) Stationnement en parallèle (ou longitudinal) :

Le stationnement en parallèle est un mode de stationnement qui consiste à garer une automobile parallèlement à un trottoir. Ce type de stationnement est essentiellement réservé aux stationnements sur rue. Il est considéré comme étant le type de stationnement sur rue le plus sûr et donc le plus recommandé.

l) Stationnement à angle :

Le stationnement à angle est un type de stationnement qui consiste à garer une automobile de biais, de sorte à former un angle de 90°, 75°, 60°, ou 45° avec la voie de circulation connexe.

I.6 Les différents types de parking :**1.6.1 Les parkings en surface :**

Très courant dans les communes, ce type de parking comprend le stationnement en voirie (le long d'une rue, d'un quai...), les espaces dédiés au stationnement entre des bâtiments, qu'ils soient publics ou privés, des anciens terrains vagues... Si ces parkings, situés de plain-pied et en extérieur, sont appréciés pour leur facilité d'accès, ils font en revanche l'objet d'une emprise au sol importante [5].

1.6.2 Les parkings fermés ou souterrains :

Situés sous des bâtiments d'habitation, des équipements (aéroports, gare...), les parkings souterrains sont construits sur plusieurs niveaux pour permettre le stationnement d'un grand nombre de voitures. Impliquant un coût de construction plus élevé, ces parkings nécessitent par ailleurs des contraintes réglementaires plus fortes en termes de sécurité et d'environnement [5].

1.6.3 Parking à étages :

Un parking à étages, aussi nommé parking silo, est un bâtiment construit en extérieur. Son mode de fonctionnement ressemble beaucoup à celui du parking souterrain à la différence qu'il ne demande pas de lourds travaux de creusement. Là aussi des rampes permettent de monter ou de descendre les étages en voiture, et il y a des ascenseurs et des escaliers pour les piétons. Les parkings silo se prêtent particulièrement bien à un fonctionnement mutualisé [6].

1.6.4 Parking ouvert :

Un parking ouvert est un parking dont la ventilation est assurée par des baies latérales de ventilation naturelle. Cette dernière permet les échanges d'air extérieur et intérieur sans participation mécanique. On parle également de parking largement ventilé pour les désigner.

L'appellation parking ouvert est dépendante du pays. Elle est déterminée par les normes et les arrêtés de ce dernier. Un des grands avantages de ce type de parking est l'économie d'énergie électrique en lumières car on profite de la clarté extérieure via les ouvertures en façade. Ce type de parking permet donc un meilleur tarif horaire [6].

1.6.5 Parkings automatiques :

Les parkings automatiques sont généralement des ouvrages souterrains ou en élévation dont les rampes intérieures sont remplacées par des systèmes de levage et de translation des véhicules. Deux grandes catégories se distinguent sur le marché :

1. les « norias », sortes de grandes roues verticales où les plateaux se présentent face à l'entrée du véhicule. Ce dernier, une fois laissé sur le plateau est déplacé par rotation de la noria ;
2. les « transbordeurs », combinant un ascenseur et une palette qui descendent le véhicule jusqu'à un alvéole puis le remettent par déplacement de la palette le supportant [6].

1.6.6 Parcs Relais :

Ce type de parcs se trouve dans la périphérie des villes. Ces parkings sont subventionnés par les collectivités, car ils favorisent la mobilité générale dans la mesure où leur emplacement est choisi stratégiquement. Les parcs relais sont en effet positionnés à proximité immédiate de gares routières ou ferroviaires, des stations de métro ou de tramway, des arrêts de bus... La logique de cette implantation est de favoriser l'inter modularité (la mixité des moyens de transports) afin de désengorger la circulation dans les centres urbains. Au Canada on les nomme « stationnements incitatifs » ou « Parc-O-Bus » [6].

1.6.7 Parking privé contre parking public :

Un parking public est un espace public qui a été équipé pour pouvoir accueillir des véhicules qui y stationneront. Les conducteurs devront généralement s'acquitter d'une contribution financière pour occuper une place de stationnement. La plupart du temps, les parkings publics, du fait de leurs tarifs et de leur réglementation concernant la durée, sont adaptés à un usage de moyenne voire courte durée.

Les parkings privés se trouvent, comme leur nom l'indique, dans des lieux privés comme les immeubles collectifs ou les résidences. Ils peuvent être utilisés par les résidents ou par des propriétaires ou locataires extérieurs [6].

I.7. Les points forts du stationnement intelligent :

Le stationnement intelligent contribue largement à améliorer les services pour les conducteurs et les usagers de la route et de la ville. Il présente beaucoup d'avantages car il permet de :

- Obtenir des informations précises sur les emplacements occupés ou non occupés en temps réel.
- la commodité et facilité d'utilisation des places de stationnement.
- Augmenter l'activité et se déplacer plus librement dans la ville en utilisant les technologies modernes.
- Assurer la sécurité du trafic pour les conducteurs et les utilisateurs.
- Profiter du temps de la recherche et de l'espace libre pour stationnement
- Réduire la pollution et l'utilisation de l'essence et l'émission de gaz toxiques.
- Une surveillance et une gestion améliorées et en temps réel de l'espace de stationnement disponible, ce qui entraîne une génération de revenus significative.
- Fournir un service excellent aux touristes,
- Etablir des décisions intelligentes à l'aide de données, notamment des applications d'état en temps réel et des rapports d'analyse historique.
- Travailler sur la communication d'informations aux usagers avant, pendant et après de stationnement.
- Assurer plus de sécurité par surveillance en temps réel.

I.8 Conclusion

Le stationnement intelligent joue un rôle majeur car il participe à diminuer la congestion et les nombreux problèmes relatifs à la mobilité en ville. Dans ce chapitre nous avons donné une idée sur le parking intelligent, sa définition, son fonctionnement et ses avantages.

Chapitre II

Matériels et méthodes

II.1. Introduction :

L'objectif de ce chapitre est de mettre l'accent sur la description des différents matériels logiciels et méthodes qu'on a utilisées pour la réalisation de notre projet de fin d'études.

Nous nous concentrons entre autres sur la carte et l'environnement Arduino et Fritzing, les capteurs, les actionneurs et d'autres éléments.

Nous introduisons aussi quelques notions sur la vision artificielle, qui va être un élément constituant le système du parking intelligent que nous allons proposer.

II.2. Vision et Intelligence artificielles :

Tout au long de l'histoire, les humains ont créé des machines pour travailler pour eux. Récemment, ce flux a évolué pour inclure des machines qui imitent nos sens, telles que notre vision. La technologie de reconnaissance visuelle est une technologie qui permet de voir et de reconnaître des choses. Ces technologies permettent aux machines, aux robots et aux applications de voir et de comprendre le monde tel que nous le voyons. La vision par ordinateur (également appelée vision artificielle ou vision numérique) est un type d'ingénierie informatique. Il s'agit d'apprendre aux ordinateurs à « voir » des images numériques, telles que des photos et des vidéos. Les ingénieurs qui travaillent dans ce domaine ont des tâches diverses. L'un d'eux est de trouver des moyens d'utiliser des appareils photo numériques sur des appareils et des ordinateurs. Ils trouvent également des moyens d'apprendre à l'ordinateur à reconnaître les images et les vidéos. Cela se fait par programmation ou apprentissage automatique. Il existe différents types de vision par ordinateur. Ils varient en fonction de ce que l'ordinateur essaie de reconnaître. L'ordinateur peut rechercher du texte, des images ou des visages. Nous présenterons ces trois catégories plus en détail [24].

- **Reconnaissance optique de caractères**

Les technologies de ROC sont maintenant présentes dans certaines applications pour téléphone intelligent. Ces applications prennent des photos de tes notes écrites à la main. Elles les convertissent ensuite en texte électronique. Pouvoir prendre des notes à la main pour ensuite les convertir en texte est beaucoup plus simple que de les taper sur un petit appareil. Transformer de l'information visuelle, comme tes notes écrites à la main, en données texte présente plusieurs avantages. On peut effectuer des recherches dans des données texte, elles peuvent être classées en catégories et elles occupent beaucoup moins de mémoire sur ton téléphone ou ton ordinateur, la figure II.1 illustrant la conversion de l'écriture manuscrite en texte dactylographié par la ROC[24].



Figure II.1 : La conversion de l'écriture manuscrite en texte dactylographié par la ROC

- **Reconnaissance visuelle**

Les systèmes de reconnaissance d'images déterminent ce que sont les objets. Il s'agit d'un des plus importants systèmes dans les voitures autonomes. Comme les autres voitures, les voitures autonomes doivent pouvoir détecter les objets. Mais elles doivent aussi pouvoir décider de l'action à effectuer en fonction de l'objet et de la situation. Par exemple, si la voiture reconnaît un panneau d'arrêt, elle doit s'arrêter. Mais si une voiture détecte une personne, elle doit analyser où la personne se trouve et ce qu'elle est en train de faire. La personne se trouve-t-elle en sécurité sur le trottoir? La personne traverse-t-elle la rue? Tu peux t'imaginer que ce système se doit d'être très bon à faire son travail, la figure II.2 illustrant Voiture autonome détectant les véhicules autour d'elle. Les rectangles représentent les autres voitures détectées. [24].



Figure II.2 : La Voiture autonome détectant les véhicules autour d'elle

- **Technologies de reconnaissance faciale**

La technologie de reconnaissance faciale (TRF) est une technologie qui identifie les visages humains. Le processus utilisé est semblable à la manière dont les humains se reconnaissent entre eux. Le système de reconnaissance faciale d'un ordinateur est semblable à ton propre système de reconnaissance faciale. Tu vois le visage de quelqu'un avec tes yeux. Un téléphone intelligent prend la photo du visage de quelqu'un avec sa caméra. Ton cerveau remarque les caractéristiques faciales et les emmagasine dans ta mémoire. C'est ce qui te permet de te souvenir des gens. Un ordinateur fait la même chose, mais à l'aide d'algorithmes, La figure II.3 montre la conception de reconnaissance faciale.

De nombreux domaines utilisent maintenant les TRF. Leur domaine d'utilisation principal est la sécurité. Certains téléphones intelligents et verrous utilisent l'empreinte faciale ou le balayage de l'iris au lieu de mots de passe. L'avantage d'utiliser ainsi sa propre personne est que tu n'as pas à te souvenir de ton mot de passe [24].



Figure II.3 : Concept de reconnaissance faciale

II.3. Partie matérielle :

II.3.1.La carte Arduino UNO :

II.3.1.1. Présentation de la carte :

Arduino est une plate-forme de prototypage d'objets interactifs à usage créatif constituée d'une carte électronique et d'un environnement de programmation. Sans tout connaître ni tout comprendre de l'électronique, cet environnement matériel et logiciel permet à l'utilisateur de formuler ses projets par l'expérimentation directe avec l'aide de nombreuses ressources disponibles en ligne.

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, le pilotage d'un robot, la commande des moteurs, les jeux de lumières, la communication avec l'ordinateur et la commande des appareils mobiles [7] [8].

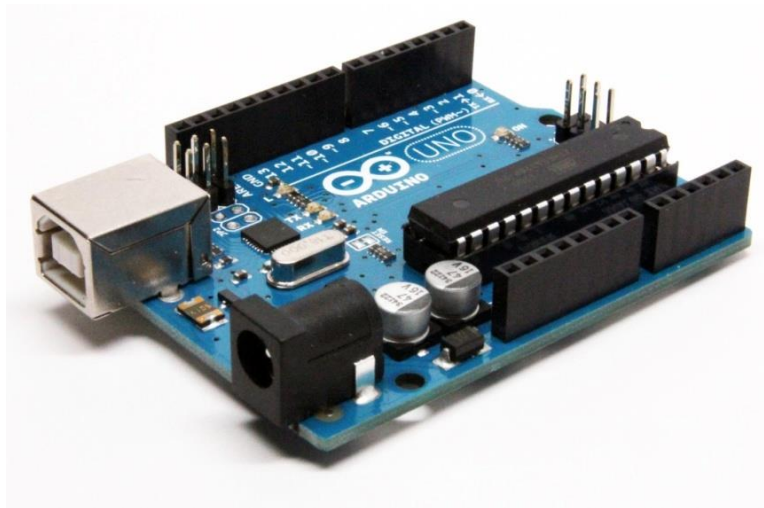


Figure II.4: La carte Arduino UNO

II.3.1.2. Caractéristiques de la carte Arduino UNO :

Nous résumons les caractéristiques de la carte Arduino UNO dans le tableau II.1 ci-dessous :

Tableau II.1 : Caractéristiques de la carte Arduino Uno

Micro contrôleur	ATmega328
Tension d'alimentation interne	5v
Tension d'alimentation (recommandée)	7 à 12V, limites = 6 à 20 V
Entrées/sorties numériques	14 dont 6 sorties PWM
Entrées analogiques	6
Courant max par broches E/S	40 Ma
Courant max sur sortie 3,3V	50Ma
Mémoire Flash	32 KB dont 0.5 KB utilisée par le bootloader
Mémoire SRAM	2 KB
Mémoire EEPROM	1 KB
Fréquence horloge	16 MHz

II.3.1.3. Les différents composants de la carte Arduino UNO :

La figure II.5 englobe les différents composants de la carte Arduino, qui sont détaillés dans ce qui suit:

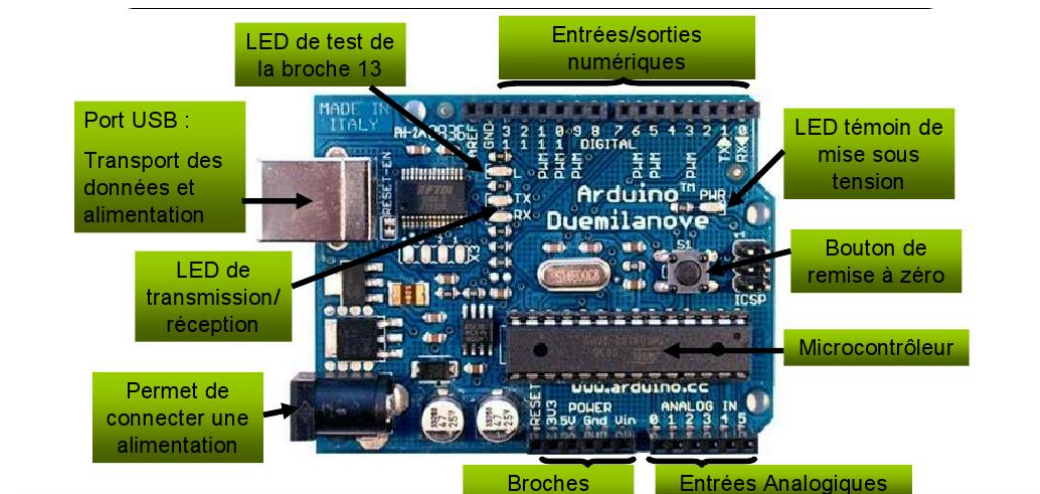


Figure II.5: Les composants de la carte Arduino UNO

a- Le Microcontrôleur ATmega328

a-1 Description

Un microcontrôleur ATmega328 est un circuit intégré qui rassemble sur une puce plusieurs éléments complexes dans un espace réduit au temps des pionniers de l'électronique. Aujourd'hui, en soudant un grand nombre de composants encombrants ; tels que les transistors ; les résistances et les condensateurs tout peut être logé 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 II.6 montre un microcontrôleur ATmega 328, qu'on trouve sur la carte Arduino [9].

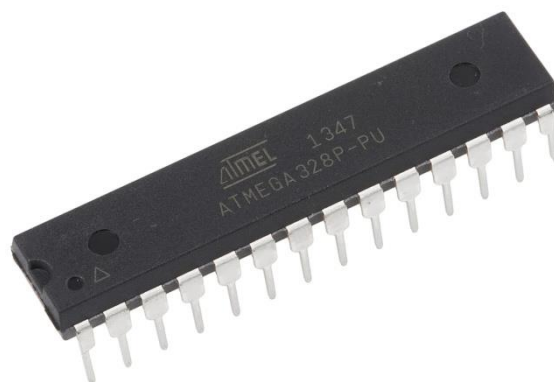


Figure II.6 : Microcontrôleur ATmega328

Le microcontrôleur ATmega328 est constitué d'un ensemble d'éléments qui ont chacun une fonction bien déterminée. Il est en fait constitué des mêmes éléments que sur la carte mère d'un ordinateur. Globalement, l'architecture interne de ce circuit programmable se compose essentiellement de :

- **La mémoire Flash:** C'est Celle qui contiendra le programme à exécuter. Cette mémoire est effaçable et réinscriptible mémoire programme de 32Ko (dont bootloader de 0.5 ko).
- **RAM:** c'est la mémoire dite "vive", elle va contenir les variables du programme. Elle est dite "volatile" car elle s'efface si on coupe l'alimentation du microcontrôleur. Sa capacité est 2 ko.
- **EEPROM:** C'est le disque dur du microcontrôleur. On y enregistre des infos qui ont besoin de survivre dans le temps, même si la carte doit être arrêtée. Cette mémoire ne s'efface pas lorsque l'on éteint le microcontrôleur ou lorsqu'on le reprogramme [10].

b-2 Caractéristiques

Nous résumons les caractéristiques du Microcontrôleur ATmega328 [10] dans le tableau II.2 ci-dessous :

Tableau II.2 : Les Caractéristiques d'un Microcontrôleur ATmega328

Vitesse de processeur	16MHz
Taille de mémoire de programme	32KB
Taille de mémoire, RAM	2KB
Taille d'EEPROM	1KB
Nombre de broches	28
Type de boîtier MCU	DIP
Nombre d'E/S	20
Type d'interface embarquée	I2C, SPI, UART
Tension d'alimentation min	1.8V
Tension d'alimentation max	5.5 V
Type de packaging	Pièce

c-3 Principe de fonctionnement

L'ATMega328, schématisé dans la figure II.7, se compose de 28 pattes qui sont [11] :

- 1 : Reset : C'est pour le bouton poussoir de réinitialisation.
- 2 : E/S numérique réservé pour RX (la réception).
- 3 : E/S numérique réservé pour TX (la transmission).
- 7-20 : VCC : c'est l'alimentation avec 5 V.
- 8-22 : GND : c'est la masse.
- 9-10 : Crystal (quartz)
- 21 : AREF (Analog référence).
- 23-24-25-26-27-28 : Des entrées analogiques.
- 4-5-6-11-14-15-16-17-18-19 : Entrées /sorties numérique .

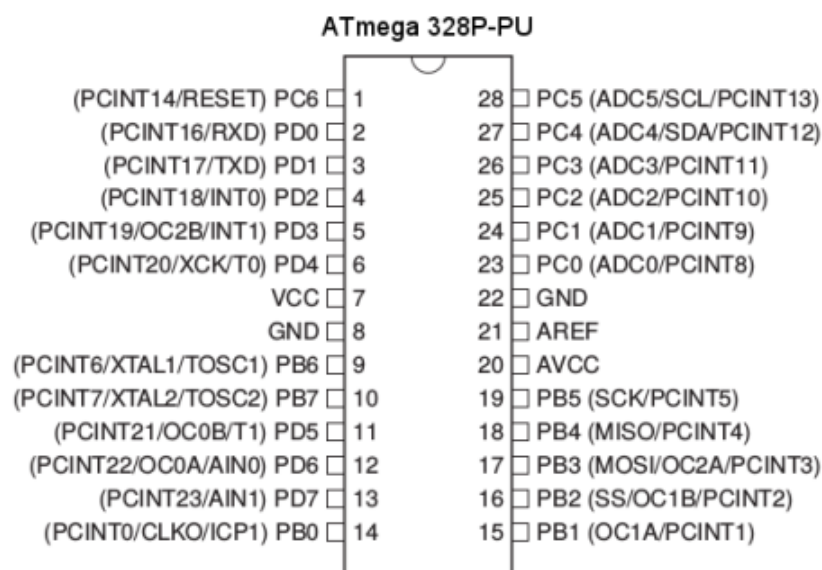


Figure II.7 : Datasheet d'ATMega328

b- Les sources d'alimentation de la carte Arduino UNO

On peut distinguer deux sources d'alimentation (Entrée Sortie) et cela comme suit :

- **VIN** La tension d'entrée positive lorsque la carte Arduino est utilisée avec une source de tension externe (à distinguer du 5V de la connexion USB ou autre source 5V régulée). On peut alimenter la carte à l'aide de cette broche, ou, si l'alimentation est fournie par le jack d'alimentation, accéder à la tension d'alimentation sur cette broche.
- **5V** La tension régulée utilisée pour faire fonctionner le microcontrôleur et les autres composants de la carte (pour info : les circuits électroniques numériques nécessitent une tension d'alimentation parfaitement stable dite "tension régulée" obtenue à l'aide d'un composant appelé un régulateur et qui est intégré à la carte Arduino). Le 5V régulé fourni par cette broche peut donc provenir soit de la

tension d'alimentation VIN via le régulateur de la carte, ou bien de la connexion USB (qui fournit du 5V régulé) ou de toute autre source d'alimentation régulée.

- **3.3V** Une alimentation de 3.3V fournie par le circuit intégré FTDI (circuit intégré faisant l'adaptation du signal entre le port USB de votre ordinateur et le port série de l'ATmega) de la carte est disponible: ceci est intéressant pour certains circuits externes nécessitant cette tension au lieu du 5V. L'intensité maximale disponible sur cette broche est de 50mA [10].

c- Les entrées /sorties

Cette carte possède 14 broches numériques (numérotées de 0 à 13). Elle peut être utilisée soit comme une entrée numérique, soit comme une sortie numérique, en utilisant les instructions `pinMode`, `digitalWrite` et `digitalRead` du langage Arduino. Ces broches fonctionnent en 5V. Chaque broche peut fournir ou recevoir un maximum de 40mA d'intensité et dispose d'une résistance interne de "rappel au plus" (pull-up) (déconnectée par défaut) de 20-50 KOhms. Cette résistance interne s'active sur une broche en entrée à l'aide de l'instruction `digitalWrite` (broche, HIGH).

En plus, certaines broches ont des fonctions spécialisées :

• Interruptions Externes :

- Broches 2 et 3. Elles peuvent être configurées pour déclencher une interruption sur une valeur basse, sur un front montant ou descendant, ou sur un changement de valeur.
- Impulsion PWM (largeur d'impulsion modulée) : Broches 3, 5, 6, 9, 10, et 11. Fournissent une impulsion PWM 8-bits à l'aide de l'instruction `analogWrite`.

- **SPI (Interface Série Périphérique):** Broches 10 (SS), 11 (MOSI), 12 (MISO), 13 (SCK). Ces broches supportent la communication SPI (Interface Série Périphérique) disponible avec la librairie pour communication SPI. Les broches SPI sont également connectées sur le connecteur ICSP qui est mécaniquement compatible avec les cartes Méga.

- **I2C :** Broches 4 (SDA) et 5 (SCL). Supportent les communications de protocole I2C (ou interface TWI (TwoWire Interface - Interface "2 fils"), disponible en utilisant la librairie `Wire/I2C` (ou TWI - Two-Wire interface - interface "2 fils").

- **LED :** Broche 13. Il y a une LED incluse dans la carte connectée à la broche 13. Lorsque la broche est au niveau HAUT, la LED est allumée, lorsque la broche est au niveau BAS, la LED est éteinte. La carte UNO dispose 6 entrées analogiques (numérotées de 0 à 5), 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.

La carte Arduino UNO intègre un fusible qui protège le port USB de l'ordinateur contre les surcharges en intensité (le port USB est généralement limité à 500mA en intensité). Bien que la plupart des ordinateurs aient leur propre protection interne, le fusible de la carte fournit une couche supplémentaire de protection. Si plus de 500mA sont appliqués au port USB, le fusible de la carte coupera automatiquement la connexion jusqu'à ce que le court-circuit ou la surcharge soit stoppé [7].

d- ports de communications

La carte Arduino UNO a de nombreuses possibilités de communications avec l'extérieur. L'Atmega328 possède une communication série UART TTL (5V), grâce aux broches numériques 0 (RX) et 1 (TX). On utilise (RX) pour recevoir et (TX) transmettre (les données séries de niveau TTL). Ces broches sont connectées aux broches correspondantes du circuit intégré ATmega328 programmé en convertisseur USB – vers – série de la carte, composant qui assure l'interface entre les niveaux TTL et le port USB de l'ordinateur.

Comme un port de communication virtuel pour le logiciel sur l'ordinateur, La connexion série de l'Arduino est très pratique pour communiquer avec un PC, mais son inconvénient est le câble USB, pour éviter cela, il existe différentes méthodes pour utiliser ce dernier sans fil :

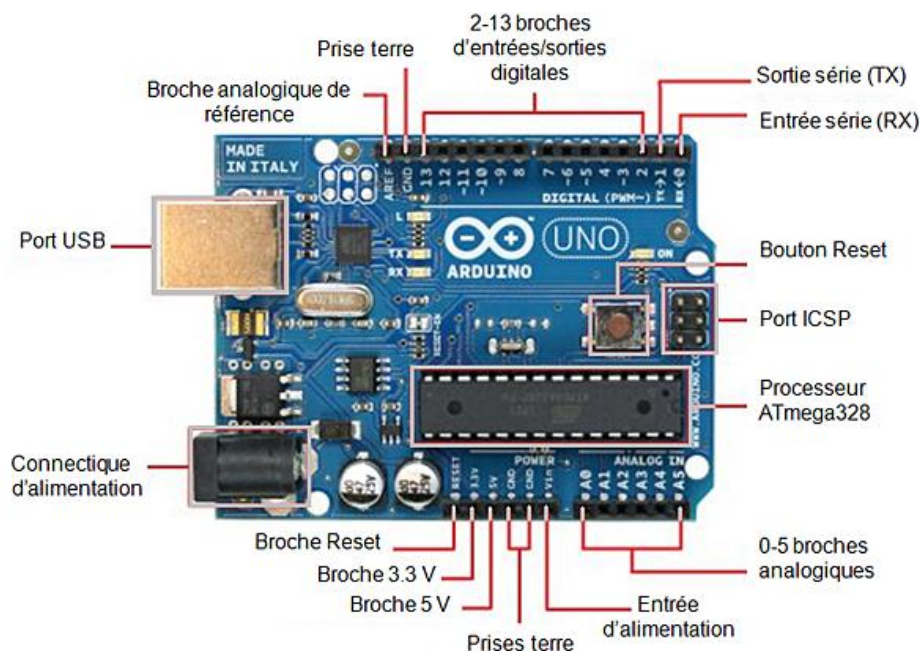


Figure II.8: Constitution de la carte Arduino UNO

II.3.2. Module RFID RC522

➤ Lecteur RFID :

a- Description

Le module RC522, décrit dans la figure II.9, est une interface qui permet l'identification sans contact à partir d'un badge ou une clé RFID. Lancé par la société NXP, Il est basé sur le

circuit intégré Philips RC522 et il communique avec Arduino via l'interface SPI. Il utilise la bande ISM 13.56MHz, la distance de communication peut aller jusqu'à 6cm mais la plupart des modules NFC marchent très bien avec 1cm de distance, il s'agit d'une puce sans contact à basse tension, à faible cout et de petite taille, un choix idéal pour les instruments intelligents et portables.

Le mécanisme RFID/NFC se base sur une radio communication de courte distance, ils utilisent la norme ECM-A340 et ISO/IEC 18092. Le MFRC522 utilise un concept avancé de modulation et de démodulation qui est entièrement présenté dans tous les types de protocoles et de méthodes de communication sans contact à 13,56 MHz, il prend en charge l'algorithme de cryptage CRYPTO1 pour vérifier les produits MIFARE de communications sans contact à haute vitesse, avec un débit de transmission de données bidirectionnel pouvant atteindre 424 kbits/ S. En tant que nouveau produit de la série de cartes de lecteur hautement intégrées de 13,56 MHz, il communique avec la machine hôte via le mode série, vous pouvez choisir entre I2C, SPI, UART en série. Ce module est idéal pour des projets du parking intelligent pour identifier la personne avec son badge avant d'ouvrir la barrière. Il peut être utilisé dans n'importe quel projet d'identification, la figure II.10 montre l'utilisation d'un lecteur RFID avec une carte Arduino UNO [12].



Figure II.9 : Module RFID RC522

b- Caractéristiques [12]

- Basée sur la puce Philips MFRC522
- Power Voltage : 3.3V
- Current: 13-26mA
- Fréquence d'utilisation : 13.56MHz
- Distance opérationnelle : 0 ~ 60mm

- Interface : SPI
- Dimensions : 40mm × 60mm
- Module Name : MF522-ED
- Working current : 13—26mA/ DC 3.3V
- Standby current : 10-13mA/DC 3.3V
- Sleeping current : <80uA
- Peakcurrent : <30mA
- Data communication speed : Maximum 10Mbit/s
- Card types supported : mifare1 S50、 mifare1 S70、 mifare Ultra Light mifare Pro、 mifare Desfire
- Working temperature : -20—80 degree
- Storage temperature : -40—85 degree
- Humidity : relevant humidity 5%—95%
- Max SPI speed : 10Mbit/s

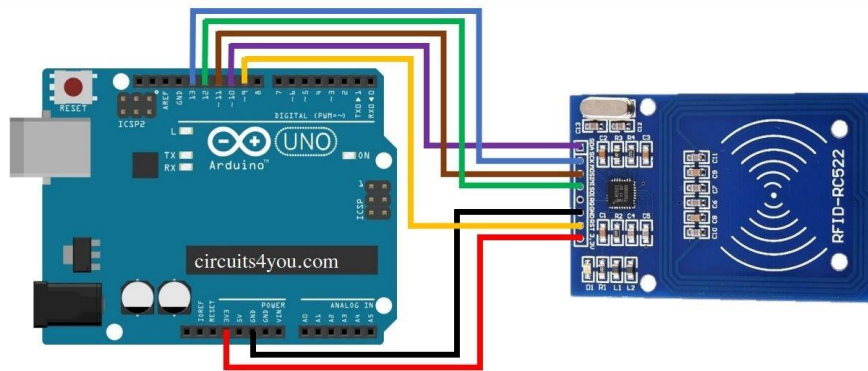


Figure II.10: Utilisation d'un lecteur RFID avec Arduino UNO

➤ **Carte RFID :**

La Mifare Type-A RFID Card fonctionne à une fréquence sans fil de 13,56 MHz, conformément à la norme ISO14443A, et contient 1 000 octets de mémoire de lecture / écriture.

Cette carte RFID peut être lue par presque tous les lecteurs RFID / NFC 13,56 MHz ou appareils mobiles compatibles NFC. Ces puces peuvent être écrites et stocker jusqu'à 1 Ko de données dans une EEPROM inscriptible divisée en banques, et peuvent gérer plus de 100 000 réécritures. Le numéro d'identification ne peut pas être changé [18].



Figure II.11 : Carte RFID

II.3.3. Module de capteur infrarouge TCRT5000:

a- Description

Le module suivant utilise des capteurs de proximité infrarouges TCRT5000 (émetteur et récepteur), décrit dans la figure II.13. Il existe un maximum de 4 broches, dont deux sont les broches VCC et GND, les deux autres étant une sortie analogique et une sortie numérique. Une fois que vous avez le module alimenté, soyez invité à le placer afin que les DEL infrarouges soient tournées vers la direction que vous souhaitez détecter. Il existe même un trou de montage sur le PCB si nécessaire. En ce qui concerne les sorties, ce module offre des options analogiques et numériques. La broche numérique aura une valeur de 0 ou 1 indiquant si un objet est ou non détecté (ou si la lumière est réfléchié ou non, cela est utile lors de la détection de lignes en noir et blanc). La sortie analogique fournira des données relatives à la distance de l'objet détecté. Il existe même un potentiomètre pour permettre des ajustements de sensibilité (Notez que ceci est un capteur de proximité, dans ce cas, ce qui signifie une portée maximale n'exécède pas 25mm), la figure II.12 montre un module de capteur infrarouge TCRT5000 [13].

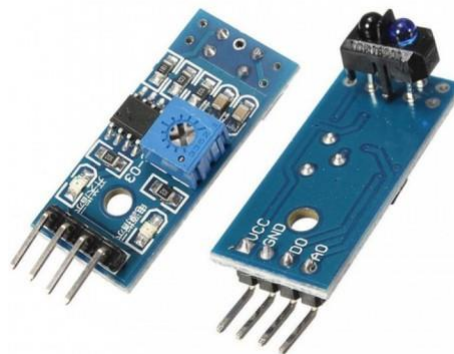


Figure II.12 :TCRT5000 Module de capteur infrarouge

b- Caractéristiques [13]

- e- Tension de service: 3.3VDC à 5VDC
- f- IC: LM393
- g- Capteur: TCRT5000
- h- Plage de fonctionnement: 1mm à 25mm
- i- Dimensions: 31mm x 14mm / 1.22 po x 0.55 po

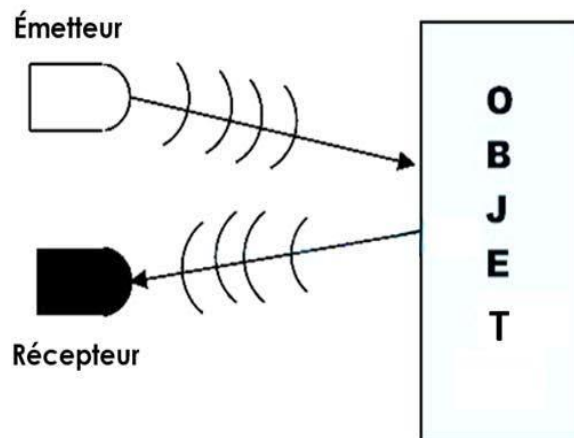


Figure II.13: TCRT5000 Module de capteur infrarouge de proximité pour détection d'obstacle

II.3.4. Servomoteur :**a- Description**

Un servomoteur, mentionné dans la figure II.14, est un actionneur (moteur) capable de maintenir une opposition à un effort statique et dont la position est vérifiée en continue et corrigée en fonction de la mesure. Le servomoteur est composé de plusieurs éléments visibles :

- Les files.
- L'axe de rotation sur lequel est monté un accessoire en plastique ou en métal.
- Le boîtier qui le protège.

aussi de plusieurs éléments que l'on ne voit pas et qui se trouvent à l'intérieur du boîtier.

- Un moteur à courant continu.
- Des engrenages pour former un réducteur.
- Un capteur de position de l'angle d'orientation de l'axe (un potentiomètre bien souvent).

- Une carte électronique pour le contrôle de la position de l'axe et le pilotage du moteur à courant continu.

La position est définie avec une limite de débattement d'angle de 180 degrés. Souvent abrégé en « servo » provenant du latin *servus* qui signifie « esclave », la figure II.15 montre un servomoteur avec une carte Arduino UNO.

Il existe plusieurs servomoteurs, dans notre application on a utilisé un servomoteur de faible puissance (9 Gramme) SG90.

Il contient trois fils, un pour l'alimentation, l'autre pour la masse(GND) et la dernière pour le signal d'entrée (la commande) [14].

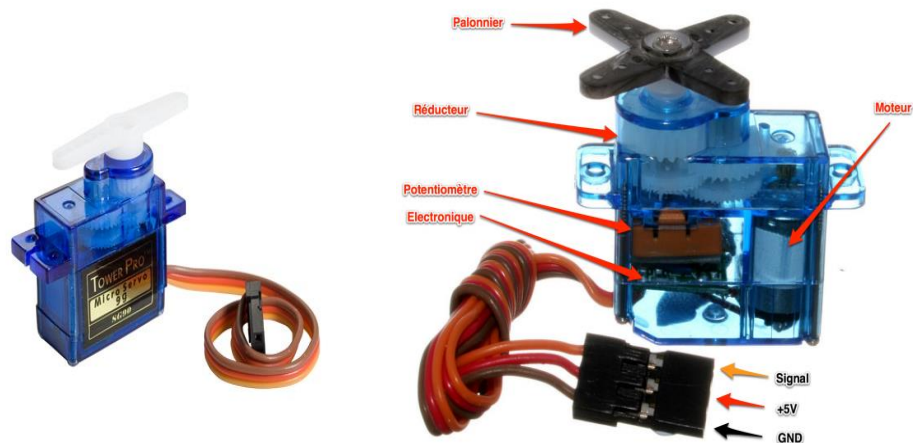


Figure II.14 : Vue d'un servomoteur

b- Caractéristiques [15]

- j-** Nombres de broches : 3 fils (Marron : Masse) (Rouge : Vcc) (Orange : Commande)
- k-** Dimension : 22mm x 11.5mm x 27mm
- l-** Poids : 9 grammes
- m-** Tension d'alimentation Min : 4.8V
- n-** Tension d'alimentation Max : 6V
- o-** Vitesse 0.12 s/60° sous 4.8V
- p-** Couple 1.2 kg/cm sous 4.8V
- q-** Consommation 125

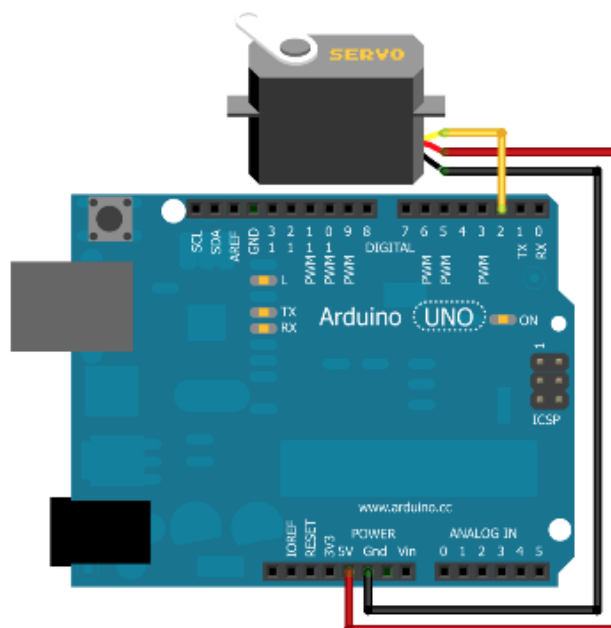


Figure II.15: Un servomoteur avec une carte Arduino Uno

c- Principe de fonctionnement d'un servomoteur

Les servomoteurs sont commandés par l'intermédiaire d'un câble électrique qui lui transmet des consignes de position sous forme d'un signal codé en largeur d'impulsion communément appelé PWM. Cela signifie que c'est la durée des impulsions qui détermine l'angle absolu de l'axe de sortie et donc la position de bras de commande du servomoteur, comme le montre la figure II.16 ci-dessous :

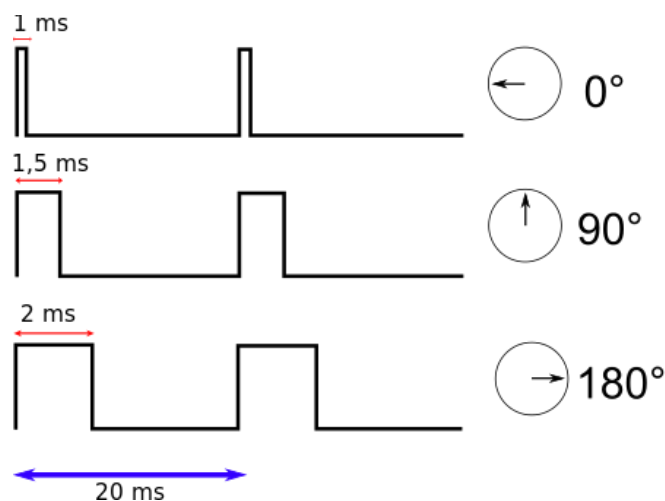


Figure II.16 : l'angle de l'axe du servomoteur en fonction de la largeur de l'impulsion

Le signal est répété périodiquement, en général toutes les 20 millisecondes, ce qui permet à l'électronique de contrôler et de corriger continuellement la position angulaire de l'axe de sortie, cette dernière étant mesurée par le potentiomètre.

Lorsque le moteur tourne, l'axe du servomoteur change de position, ce qui modifie la résistance du potentiomètre. Le rôle de l'électronique est de commander le moteur pour que la position de l'axe de sortie soit conforme à la consigne reçue.

II.3.5. Module relais 5v :

a- Description

Idéal pour le pilotage de dispositifs externes dont la tension/courant est élevés. Cette platine intègre 1 relais 5V (figure II.17) avec optocoupleurs de protection dont le pilotage s'effectue sur un connecteur mâle au pas de 2,54 mm directement depuis un module Arduino ou compatible [16].



Figure II.17 : Vue d'un relais 5v

b- Caractéristiques

- 1 sortie relais RTC (NO-COM-NF)
- Raccordement sur les relais via bornes à vis
- Pilotable direct depuis Arduino
- Entrées isolées par des optocoupleurs
- Pouvoir de coupure: 30V / 10A max.
- Dimensions: 50,6 x 38,8 x 19,3 mm
- Poids: 30 g

II.3.6. Afficheur LCD 16x2 :

a - Description

L'écran à cristaux liquides, LCD en anglais pour Liquid Crystal Display, permet la création d'écran plat à faible consommation d'électricité. Aujourd'hui ces écrans sont utilisés presque dans tous les affichages électroniques.

Les écrans à cristaux liquides utilisent la polarisation de la lumière par des filtres polarisants et la biréfringence de certains cristaux liquides en phase nématique, dont on peut faire varier l'orientation en fonction du champ électrique. Du point de vue optique, l'écran à cristaux liquides est un dispositif passif : il n'émet pas de lumière, seul sa transparence varie, et il doit donc disposer d'un éclairage.

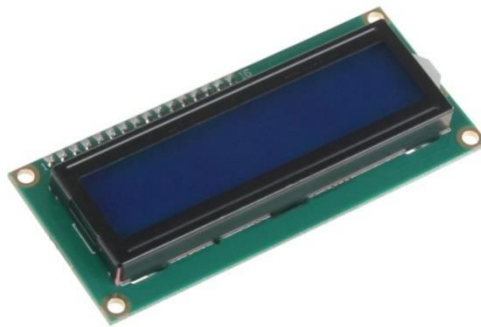


Figure II.18 : Un afficheur LCD 16x2

b-Caractéristiques

- Alimentation: 5 Vcc
- Interface I2C (adresse 0x27)
- Caractères blancs sur fond bleu
- Contraste ajustable via potentiomètre
- Dimensions: 80 x 38 x 18 mm

c- Branchement et utilisation

Dans la figure II.18, en haut de l'afficheur, figurent 16 marques de soudure. C'est une bonne façon de repérer les afficheurs compatibles avec la bibliothèque LCD. Ces afficheurs peuvent communiquer avec l'Arduino via la bibliothèque Liquid Crystal.

En retournant l'écran LCD on trouve au verso au niveau des broches deux numéros 1 et 16 désignant respectivement la première et la dernière broche. Pour pouvoir brancher correctement l'écran il faut respecter le schéma suivant (figure II.19) :

- Les deux premiers pins servent à l'alimentation de l'écran respectivement le GND et le 5V.
- Le troisième pin est connecté à un potentiomètre et sert pour régler l'affichage (le contraste de l'écran).
- Le quatrième, est connecté au pin 12 de l'Arduino par exemple. Il sert à sélectionner la zone mémoire de l'écran LCD dans laquelle nous allons écrire.

- Le cinquième doit toujours être connecté au ground. C'est un sélecteur de mode lecture ou écriture.
- Le sixième, est connecté au pin 11 par exemple de l'Arduino. Il permet de lancer ou non l'écriture dans les zones mémoires.
- Les quatre suivants (reliés au ground) servent pour la communication 8 bits. Pour la communication 4 bits, il est conseillé de les relier au ground (facultatif).
- Les quatre pins qui suivent, se connectent par exemple sur les pins 2, 3, 4, 5 de l'Arduino. Ils servent pour la communication
- Les deux pins tout à droite servent pour alimenter la LED du rétro-éclairage. [19]

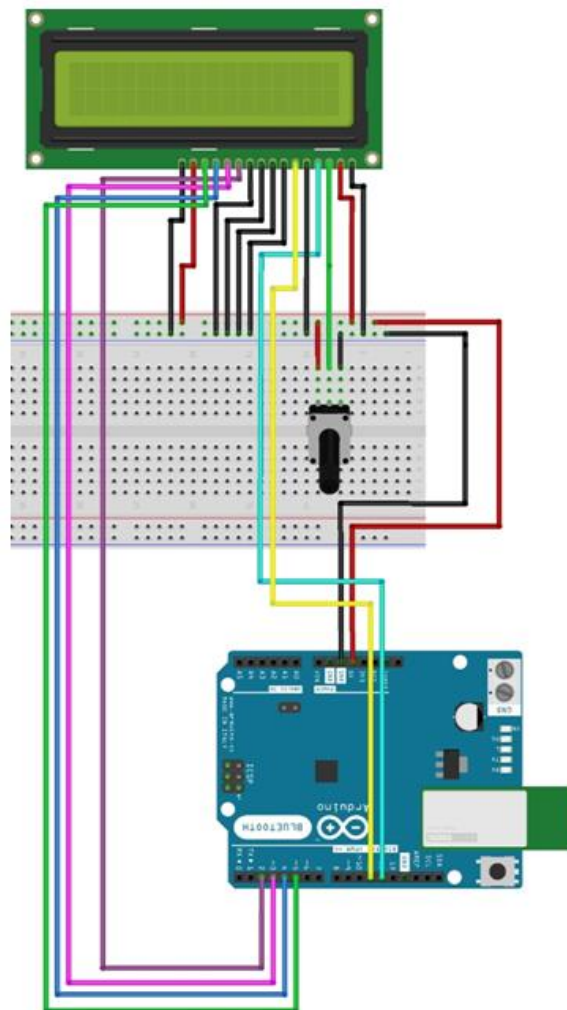


Figure II.19: Schéma d'un afficheur LCD 16x2 avec Aduino Uno

II.3.7. Module I2C LCD, PCF88574 :

a-Description

Le module s'ajoute simplement à un afficheur LCD standard afin de le commander en I2C. Le module est un petit PCB qui peut être soudé à l'arrière d'un écran LCD de sorte que vous puissiez le commander au travers d'un bus I2C. Il est basé sur le composant PCA 8574.

De manière générale, en utilisant ce produit pour réduire le nombre de broches (seulement 2 broches sont nécessaires pour l'I2C), et simplifier la connectivité avec les écrans LCD.

Il est compatible avec plusieurs types d'écrans LCD à cristaux liquides, tels que le LCD16x01, le LCD16x02, le LCD16x04 et le LCD20x04 [20].

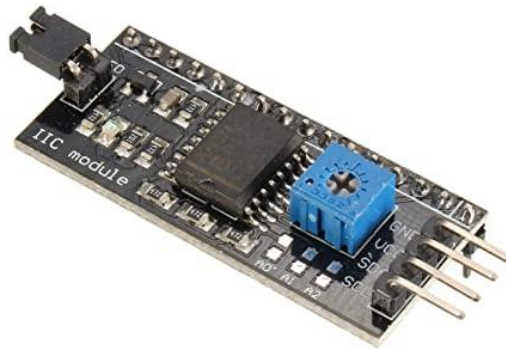


Figure II.20: Module convertisseur I2C LCD

II.4.8 Buzzer :

a- Description

Le buzzer, décrit dans la figure II.21, est un composant principalement conçu de bandes qui répondent aux effets Piézoélectrique. La piézoélectricité est une propriété de certains minéraux déformés sous l'action d'un champ électrique. Ce phénomène est réversible, si l'on déforme ce minéral, il générera de l'électricité [23].



Figure II.21: Vue d'un Buzzer

Dans l'univers Arduino, le buzzer est principalement utilisé pour faire des sons.

b- Caractéristiques

Les caractéristiques du buzzer sont décrites dans le tableau II.3 suivant :

Tableau II.3 : Les Caractéristiques du buzzer

Tension nominale	6V DC
Tension de fonctionnement	4-8V DC
Courant nominal	<30Ma
Type de son	Bip continu
Fréquence de résonance	~ 2300 Hz

c- Branchement et utilisation

Le branchement du buzzer avec la carte Arduino Uno est donné dans le schéma de la figure II.22.

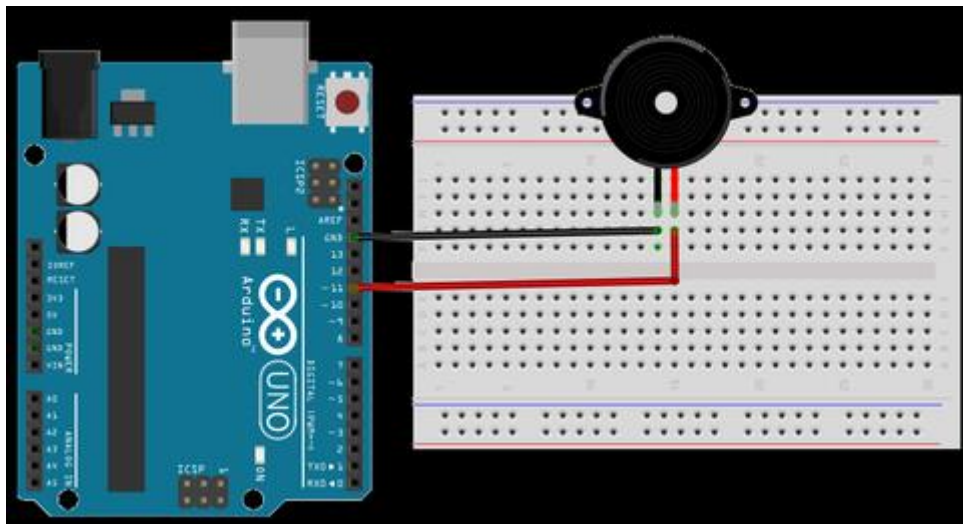


Figure II.22: Buzzer avec une carte Arduino Uno

II.3.9. Capteur ultrason (HC-SR04) : [21]

a- Description

Le capteur à ultrasons HC-SR04 est capable de mesurer la distance des objets situés de 2cm à 400cm du capteur avec une précision de 3mm. Le capteur est composé d'un émetteur d'ultrasons, d'un récepteur et du circuit de commande.

b- Caractéristiques

Le tableau II.4 ci-dessous résume les caractéristiques du capteur ultrason HC-SR04 :

Tableau II.4 : Les Caractéristiques du capteur ultrason HC-SR04

Tension d'alimentation	5V DC
Courant d'alimentation	15mA
Fréquence de travail	40Hz
Distance maximale de détection	4m

Distance minimale de détection	2cm
Angle de détection	15 degrés
Signal d'entrée de l'émetteur	Impulsion à l'état haut de 10µs
Signal de sortie du récepteur	Signal numérique à l'état haut et la distance proportionnellement
Dimension	45*20*15mm

c-Principe de fonctionnement :

- Envoyer un signal numérique à l'état haut sur l'émetteur pendant 10 µs.
- Le capteur envoie automatiquement 8 impulsions d'ultrasons à 40 kHz et détecte les signaux qui reviennent.
- Si le signal revient, la durée de l'état haut du signal reçue correspond au temps entre l'émission des ultrasons et leur réception.
- $Distance = (\text{temps à l'état haut signal reçu} * \text{vitesse du son})/2$, avec vitesse du son dans l'air=340 m/s.



Figure II.23 : Module HC-SR04 capteur ultrason

II.3.10.Les composants :

II.3.10.1.Bread board (platine de montage) :

La carte d'expérimentation ou la carte prototype en figure II.24, appelée aussi planche à pain, est un appareil qui peut produire des prototypes de circuits électroniques et les tester.

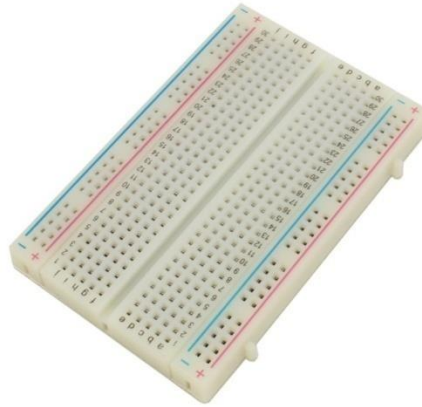


Figure II.24 : Bread board

II.3.10.2. Les fils de liaison :

Ce sont des fils dans une variété de couleurs destinés à établir la connexion entre les composants électroniques et la bread board.



Figure II.25 : Les fils de liaison

II.3.10.3. Résistance (R) :

La résistance est un Composant non polarisée est l'opposé du flux de courant et est proportionnelle à sa "résistance". Elle est exprimée en ohms. Le code couleur permet d'identifier cette valeur.

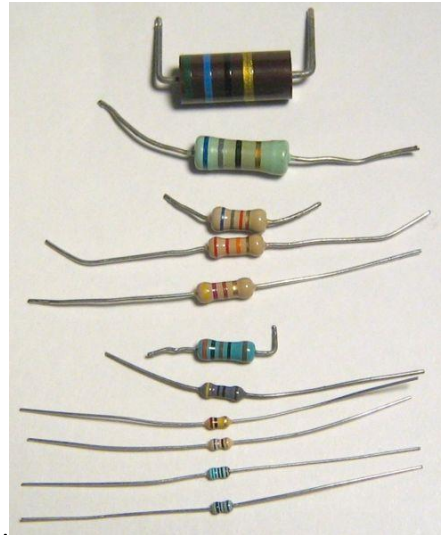


Figure II.26 : Résistances

II.3.10.4. Diode LED :

Une diode électroluminescente, plus connue sous l'appellation Del ou Led (*light-emitting diode*), désigne un composant optoélectronique qui permet l'émission de lumière monochromatique.



Figure II.27 : Diode LED

II.4. Partie logicielle

Le matériel utilisé ne peut fonctionner qu'à travers des programmes qui peuvent le mettre en marche, pour atteindre un objectif bien précis. Nous nous sommes basés sur le logiciel Arduino et Fritzing dont voici quelques détails de base :

II.4.1. Le logiciel Arduino :

Le logiciel de programmation de la carte Arduino utilise un code proche du langage C. Une fois, le programme saisi ou modifié, il sera transféré et mémorisé dans la carte à travers une liaison USB.

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

II.4.1.1. Présentation de l'IDE ARDUINO:

Le logiciel Arduino est un environnement de développement (IDE) open source et gratuit, téléchargeable sur le site officiel d'Arduino [22].

L'IDE Arduino permet :

- D'éditer un programme : des croquis (sketch en Anglais),
- De compiler ce programme dans le langage « machine » de l'Arduino,
- De téléverser le programme dans la mémoire de l'Arduino,
- De communiquer avec la carte Arduino grâce au terminal,

A l'ouverture, l'interface visuelle du logiciel ressemble à la figure II.28

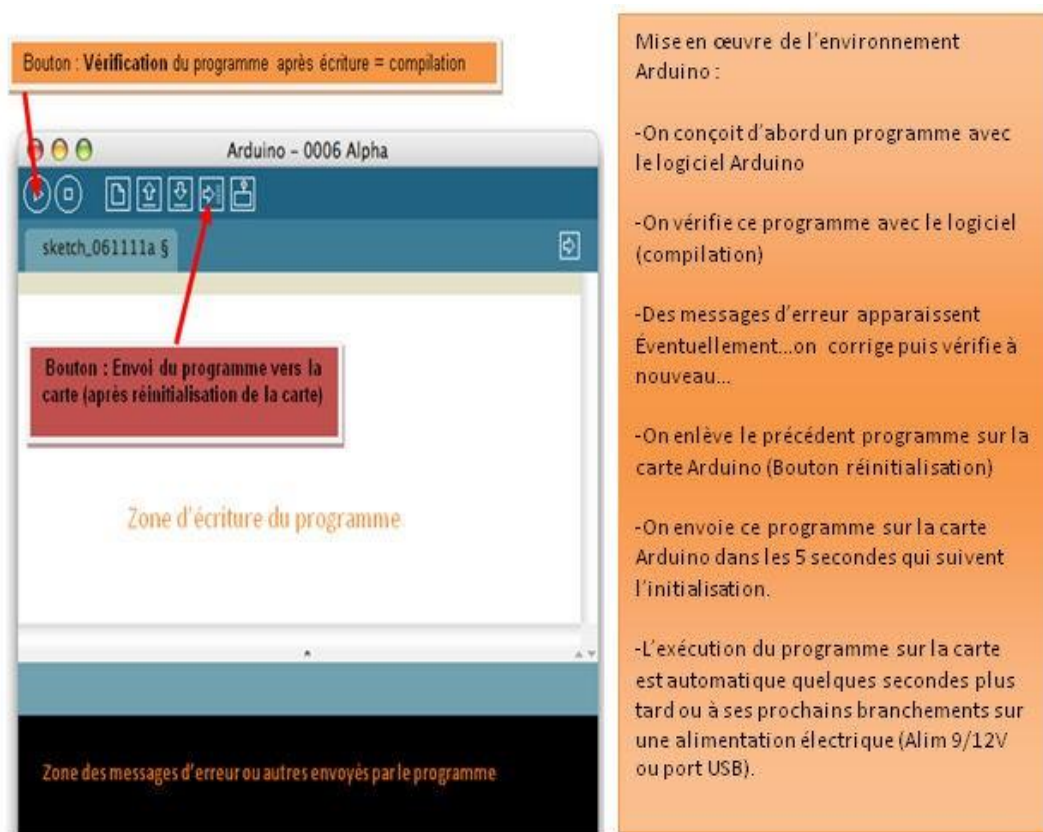


Figure II.28: L'interface visuelle du logiciel Arduino

II.4.1.2. Programmer avec Arduino

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

II.4.1.3. Structure d'un programme

Un exemple de structure de programme Arduino est présenté dans la figure II.29

Il y a trois phases consécutives :

Commentaires multilignes pour se souvenir du patch.

1/La définition des constantes et des variables
int ledPin = 13; // LED connectée à la broche 13

2/La configuration des entrées et sorties
void setup()
{
 pinMode(ledPin, OUTPUT); // configure ledPin comme une sortie
}

3/La programmation des interactions et Comportements
void loop()
{
 digitalWrite(ledPin, HIGH); // met la sortie à l'état haut (led allumée)
 delay(3000); // attente de 3 secondes
 digitalWrite(ledPin, LOW); // met la sortie à l'état bas (led éteinte)
 delay(1000); // attente de 1 seconde
}

Une fois la dernière ligne exécutée, la carte revient au début de la troisième phase et recommence sa lecture et son exécution des instructions successives. Et ainsi de suite. Cette **boucle** se déroule des milliers de fois par seconde et anime la carte.

Commentaire

```

/* Ce programme fait clignoter une LED branchée sur la broche 13
 * et fait également clignoter la diode de test de la carte
 */

int ledPin = 13; // LED connectée à la broche 13

void setup()
{
  pinMode(ledPin, OUTPUT); // configure ledPin comme une sortie
}

void loop()
{
  digitalWrite(ledPin, HIGH); // met la sortie à l'état haut (led allumée)
  delay(3000); // attente de 3 secondes
  digitalWrite(ledPin, LOW); // met la sortie à l'état bas (led éteinte)
  delay(1000); // attente de 1 seconde
}

```

Done compiling.

22

Figure II.29 : L'exemple de structure de programme Arduino

II.4.2. Le logiciel Fritzing :

Fritzing, dont l'interface est présentée dans la figure II.30, est un logiciel open-source multiplateforme permettant de construire des schémas et des circuits électroniques que nous utilisons avec Arduino. Plusieurs vues sont disponibles : platine d'essai, schémas électriques et circuit imprimé. Le projet dispose d'un site web [17], et se veut un outil qui permet aux utilisateurs de documenter leurs prototypes et les partager avec d'autres.

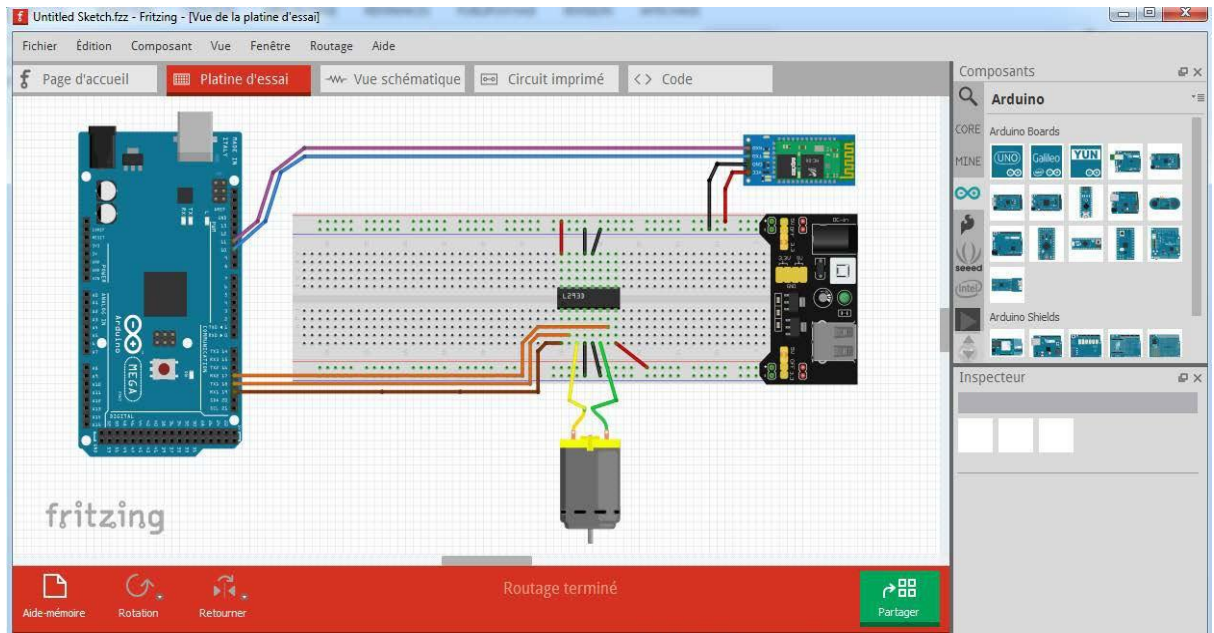


Figure II.30: L'interface de Fritzing

II.6. Conclusion :

Dans ce chapitre nous avons présenté les différents outils matériels et logiciels utilisés dans notre projet. Nous avons survolé les logiciels Arduino et Fritzing et présenté les différents Capteurs utiles pour la finalisation de notre travail.

Le chapitre suivant sera consacré à la conception et la réalisation pratique de notre projet et la simulation de système finale.

Chapitre III

Conception et réalisation

III. Conception et réalisation

III.1. Introduction

Au cours de ce chapitre nous allons détailler les différentes étapes de la réalisation de notre système, ainsi que les branchements des différents capteurs et actionneurs et leurs différents programmes de configuration associés.

Nous présentons aussi la démarche algorithmique ainsi que le déroulement des différentes étapes du programme pour la conception et la réalisation du notre parking intelligent. Dans les parkings normaux, les personnes handicapées sont marginalisées, nous leur consacrons donc une partie de notre travail.

III.2. Problématique :

Le problème du stationnement urbain est plus compliqué qu'il n'y paraît. Dans un environnement urbain à espace restreint, le stationnement est en concurrence avec les autres usages et équipements urbains. Ces solutions sont contre-intuitives et nécessitent une gestion intégrée du stationnement.

En effet, se garer, notamment dans les grandes villes, devient de plus en plus compliqué. Beaucoup de personnes quittent la ville à la recherche d'un bien immobilier moins cher et se rendent au travail en voiture. Par conséquent, les difficultés de stationnement entraîneront une plus grande densité de trafic, un grand nombre de coûts d'utilisation (carburant) et une pollution sonore dans les grandes zones urbaines, ce qui rend le stationnement un véritable défi pour tout espace urbain. De plus les personnes handicapées sont souvent marginalisées et le fait de prendre leur place dans les parkings constitue un obstacle important les empêchant d'atteindre leurs objectifs (domicile, travail, hôpital, crèche, etc ...).

Pour faire face à ces problèmes, nous suggérons une solution pour un parking automatique intelligent, pouvant s'inspirer des principes de la vision artificielle. Nous proposons un prototype de parking intelligent de 4 places pour usagers normaux et 2 places pour usagers handicapés. Cette solution peut bien s'adapter aux grands espaces, résidences, hôpitaux, aires de jeux, hôtels, aéroports, administrations, etc....

III.3. Structure du système du parking proposé :

Le système que nous visons à concevoir comporte un bloc principal constitué d'une carte à microcontrôleur de type Arduino (figure III.1). Il est commandé principalement par une carte d'accès par badge RFID, qui est connectée à un ensemble de périphériques constitués de capteurs et d'actionneurs ainsi :

a) Capteurs :

- Capteur infrarouge et/ou caméra : pour détecter la présence ou l'absence de véhicule.
- Capteur ultrason : pour détecter l'entrée et la sortie du véhicule ainsi que le comptage.

b) Actionneurs :

- Servomoteur : pour faire monter et descendre les barrières.
- Un relai : pour allumer et éteindre des lampes.

c) Le module d'acquisition des données du parking :

C'est la partie hardware du système à base de la carte Arduino UNO. Elle prend en charge la réception et l'envoi des différentes informations entre les périphériques.

Ce module gère et maintient le fonctionnement du circuit et tous les capteurs où les actionneurs qui sont connectés.

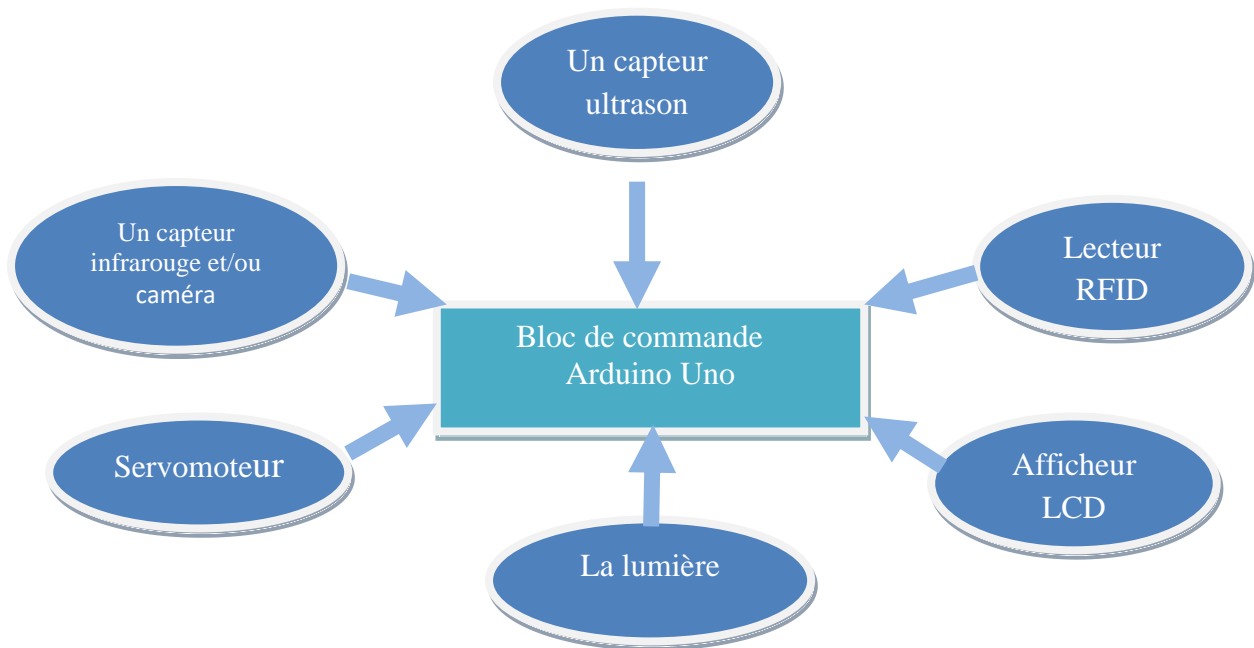


Figure III.1 : Schéma bloc de notre système

Voici, dans la figure III.2, la structure schématique globale de notre parking intelligent :

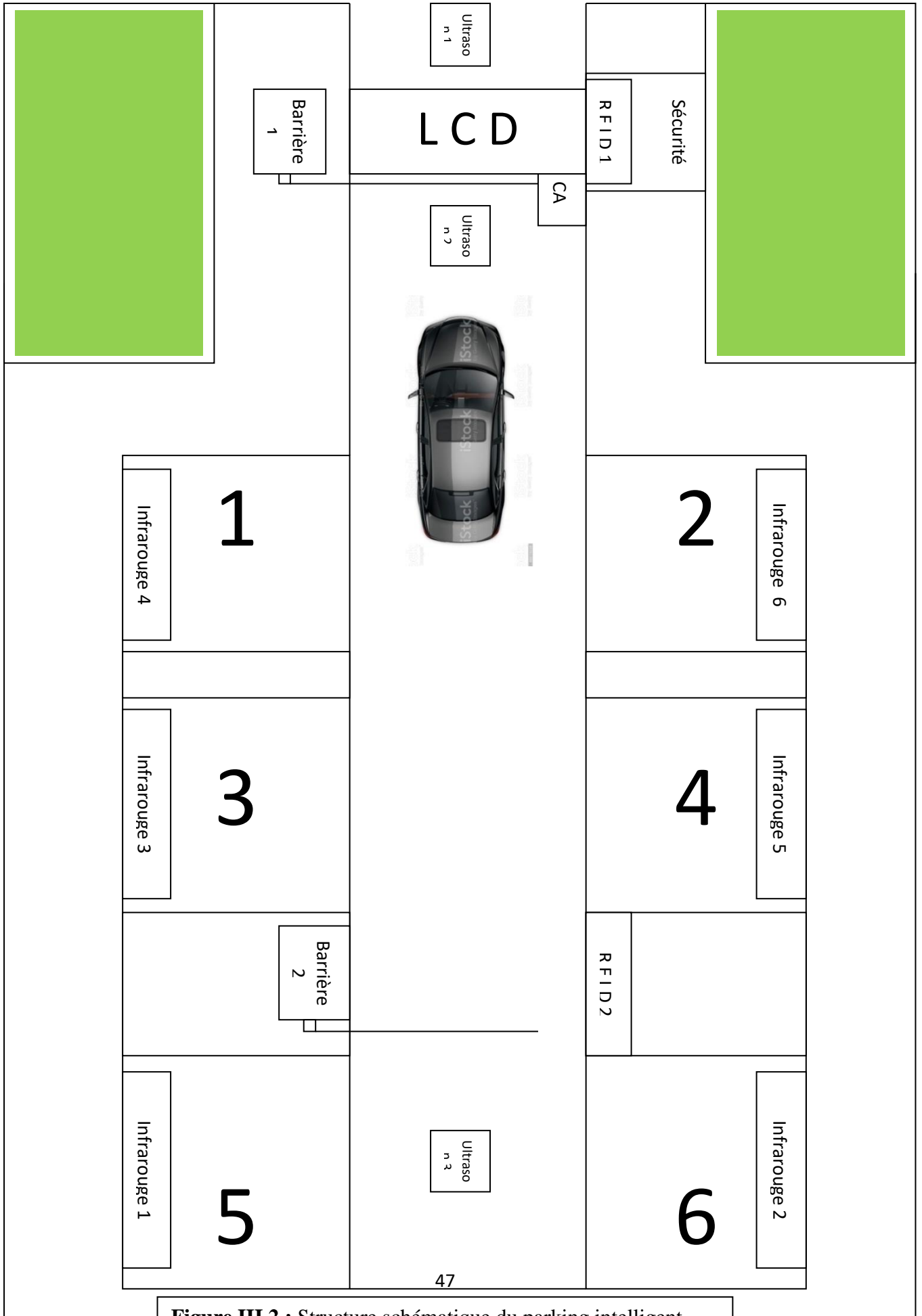


Figure III.2 : Structure schématique du parking intelligent

III.4. Conception Algorithmique :

Algorithme d'entrée

Nous considérons N pour les personnes normaux / H pour les personnes handicapés / A pour Administration
Et compteur total une variable qui mémorise le nombre de voiture total dans le parking (initialisé à zéro)

Début

Présentation du véhicule

Détection du véhicule par carte d'accès

Si l'accès n'est pas autorisé (validation négative par lecteur RFID1)

La barrière reste fermée

Sinon

Si le conducteur à une carte d'accès N

Démarrer le system de détection

Si parking N plein alors maintenir fermer la barrière 1

Sinon

Si le parking N contient au moins une place alors autoriser le stationnement

Finsi ;

Finsi ;

Finsi ;

Finsi ;

Si le conducteur à une carte d'accès H

Démarrer le système de détection

Si parking H plein alors maintenir fermer la barrière 1

Sinon

Si le parking H contient au moins une place alors

- Ouvrir la barrière 1

- Revalider la carte d'accès par le lecteur RFID2

Si le conducteur est H alors autoriser la barrière 2

Sinon laisser fermer la barrière 2

Finsi ;

Finsi ;

Finsi ;

Finsi ;

Si le conducteur à une carte d'accès A alors accès direct à la barrière 1 et 2

Finsi ;

Démarrer le système de comptage

Compteur total ← compteur total +1

Afficher le message (le nombre de voiture dans le parking est égal à compteur total)

Fin.

Algorithme sortie

Début

Démarrer le système de détection (sortie)

Si le véhicule s'éloigne du capteur infrarouge et s'approche du capteur ultrason (attaché à la barrière 1 ou barrière 2 et 1

Alors

- Ouvrir la barrière

- compteur total ← compteur total -1

Affichage du message le nombre de voiture dans le parking est égal à compteur total

Finsi ;

Fin.

Figure III.3 : Algorithmes d'entrée/sortie du véhicule dans notre parking

III.5. Principe de fonctionnement du système :

III.5.1. Détection des véhicules à la barrière (RFID/Caméra) :

Pour détecter la présence d'un véhicule à la barrière, nous avons pensé premièrement à système de vision artificielle qui doit être équipée d'une caméra (exemple en figure III.4). Pour relever la première barrière du parking, le système de vision doit détecter la présence d'une plaque d'immatriculation qui fait signe de la présence d'un véhicule. Cependant, avec la difficulté d'avoir une base d'images concrètes et une caméra, nous avons laissé tomber cet outil de détection et opter pour une alternative qui est le lecteur RFID

En effet, à la première barrière de notre parking intelligent, le lecteur RFID va permettre l'entrée des véhicules pour les personnes normaux et handicapées (barrière principale). À la deuxième barrière, on propose un autre lecteur RFID qui va autoriser l'entrée des véhicules pour les personnes handicapées uniquement (barrière secondaire).



Figure III.4 : Exemple de caméra de détection de plaques d'immatriculation

III.5.2. Détection des véhicules à l'intérieur du parking :

Le principe consiste à équiper chaque place de stationnement d'un capteur infrarouge *TCRT5000* pour détecter la présence ou l'absence d'un véhicule et d'informer en temps réel que la place est occupée ou libre.

Une fois la détection faite, le système va autoriser le véhicule à entrer (si la place est libre) et va refuser son accès si il n'y a plus de places, et ce par un message sur l'afficheur LCD.

III.5.3. Système de détection avec la technologie RFID :

Pour accepter l'entrée du véhicule, le système de détection et le système de RFID fonctionnent ensemble. Le système RFID n'ouvre pas la barrière même si le véhicule est autorisable. Le système de détection reconnaît que le parking est plein et empêche la barrière de s'ouvrir (même si le véhicule est autorisé par le RFID).

III.5.4. Système de compteur :

C'est un système qui permet de calculer le nombre de véhicules à l'intérieur du parking à l'aide de capteurs ultrasons *HC-SR04*. Nous avons utilisé deux capteurs, l'un à côté de RFID et l'autre après la barrière. Leurs buts est de savoir si le véhicule entre ou sort pour incrémenter ou décrémenter le paramètre correspondant au nombre de véhicules à l'intérieur du parking ainsi :

- *Augmentation du nombre de véhicules* : Si le véhicule est entré, il passe par le capteur 1 puis le capteur 2 (capteur de sécurité).Le système déduira alors que le véhicule est entré, donc le nombre augmentera et l’affichera sur l’écran LCD.
- *Diminution du nombre de véhicules* : Si le véhicule est sorti, il passe par le capteur 2 puis le capteur 1 (capteur de sécurité).Le système comprendra alors que le véhicule est sorti et donc le nombre diminuera sera affiché aussi sur le même écran.

III.5. Matériel :

Dans le tableau III.1 et les figures (III.5 ...II.17) suivants nous citons tous les composants intervenant dans la réalisation de notre système et dans la conception de la maquette :

Tableau III.1 : Ensemble des composants de notre parking intelligent

Composant	Nombre
Carte Arduino Uno	2
Lecteur RFID	2
Carte RFID	8
Capteur infrarouge TCRT5000	6
Servo-moteur	2
Relais 5v	1
Afficheur LCD 16x2	1
Convertisseur I2C LCD	1
Buzzer	1
LED	4
Résistance	4
Breadboard	1
Les fils de liaison	152
Capteur ultrason HC-SR04	3
Maquette	1



Figure III.5 : Les deux cartes Arduino UNO



Figure III.6 : Lecteur RFID RC522



Figure III.7 : Les cartes RFID

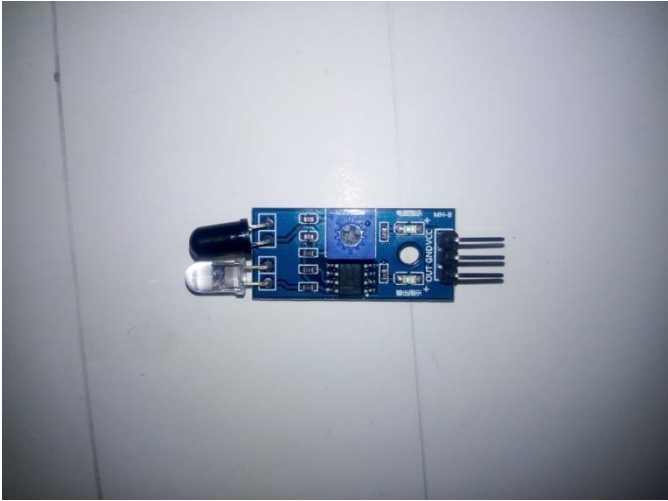


Figure III.8: Capteur infrarouge TCRT5000



Figure III.9 :Servo-moteur



Figure III.10 : Relais 5v



Figure III.11 : Afficheur LCD 16x2 avec I2C



Figure III.12 : Capteur ultrason HC-SR04



Figure III.13 : Buzzer



Figure III.14 : Les LED

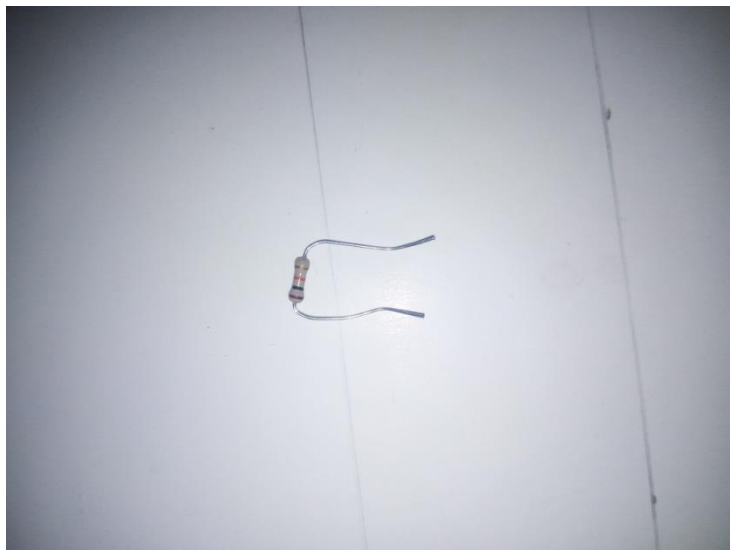


Figure III.15 : Résistance

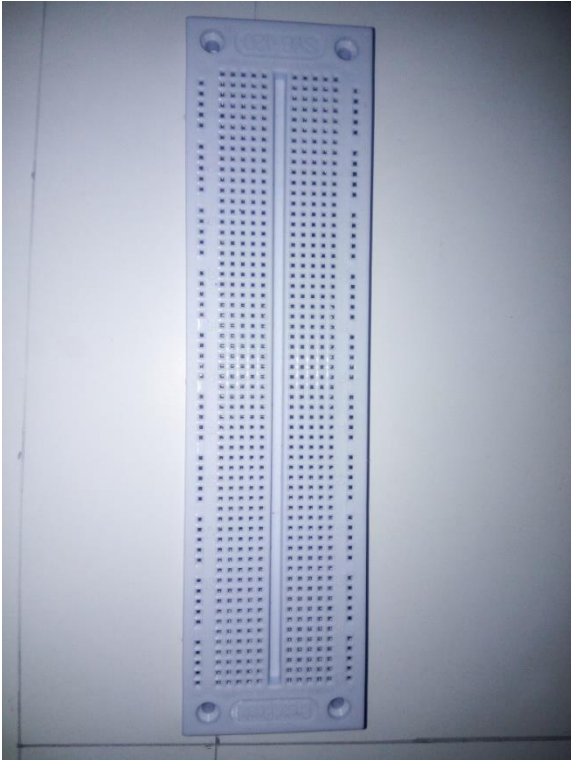


Figure III.16 : Un Breadboard



Figure III.17 : fils de liaison

III.6. Quelques simulations virtuelles des composants sous Fritzing :

Avant de passer à la réalisation de notre parking intelligent, nous avons simulé le montage de nos composants sous le logiciel Fritzing (voir chapitre 2). Les résultats sont détaillés dans les figures allant d'III.18 à III.26.

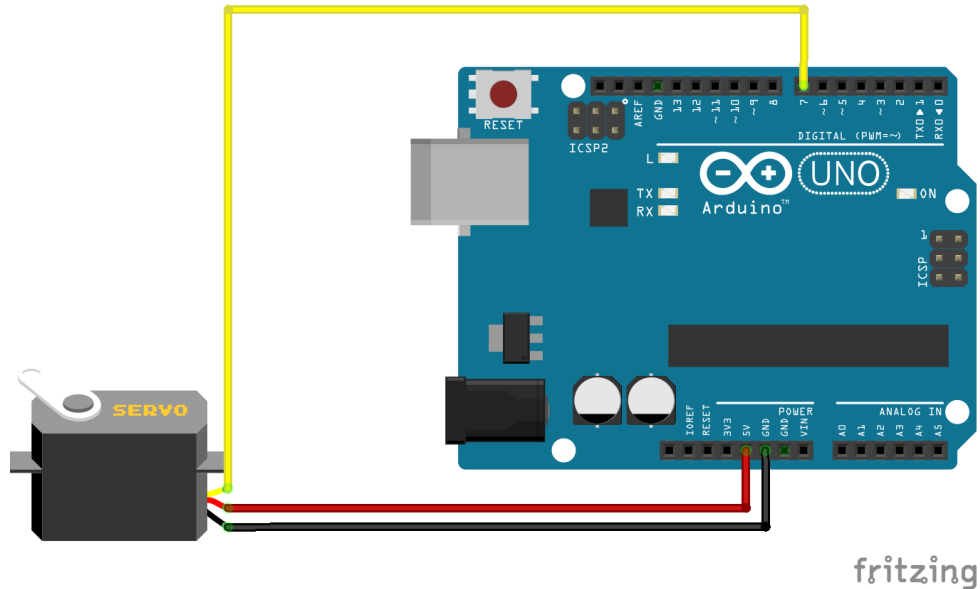


Figure III.18 : Simulation d'un Servo Moteur

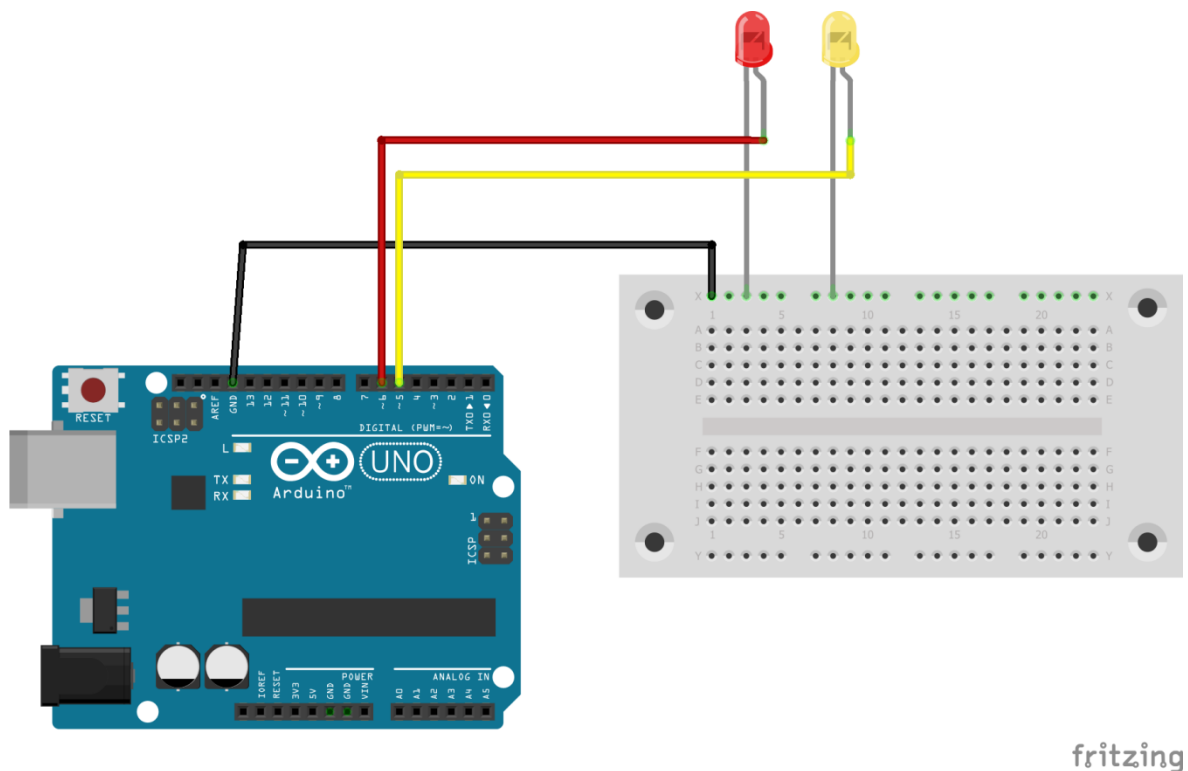


Figure III.19 : Simulation d'une Diode LED

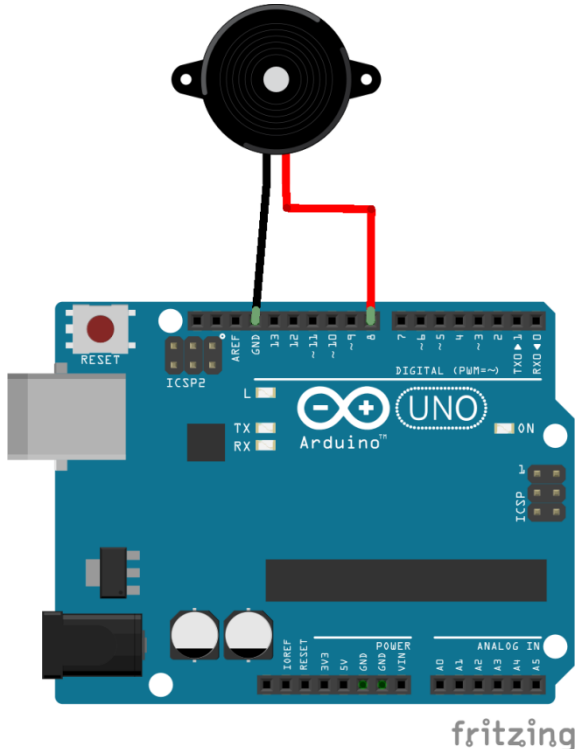


Figure III.20 : Simulation d'un Buzzer

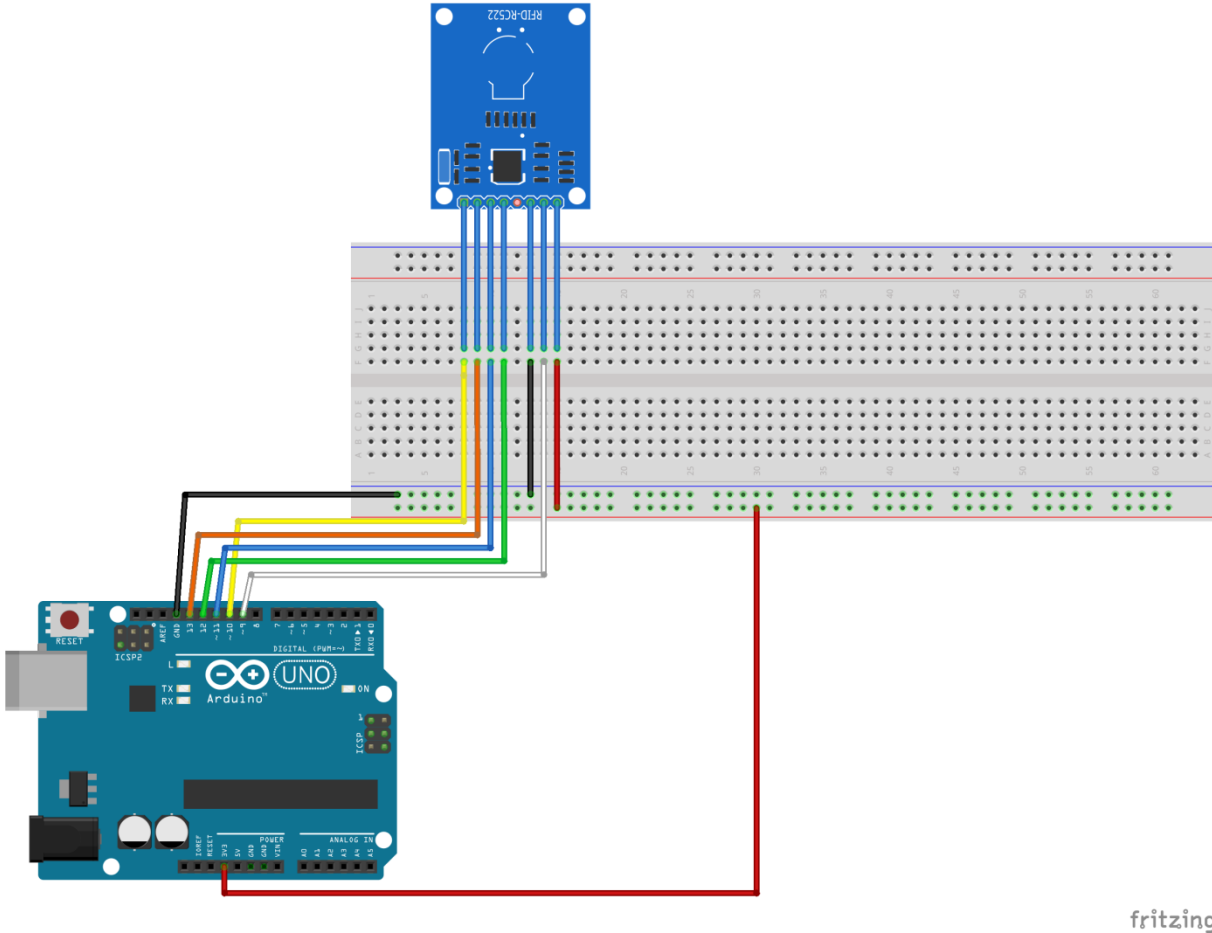


Figure III.21 : Simulation d'un Lecteur RFID

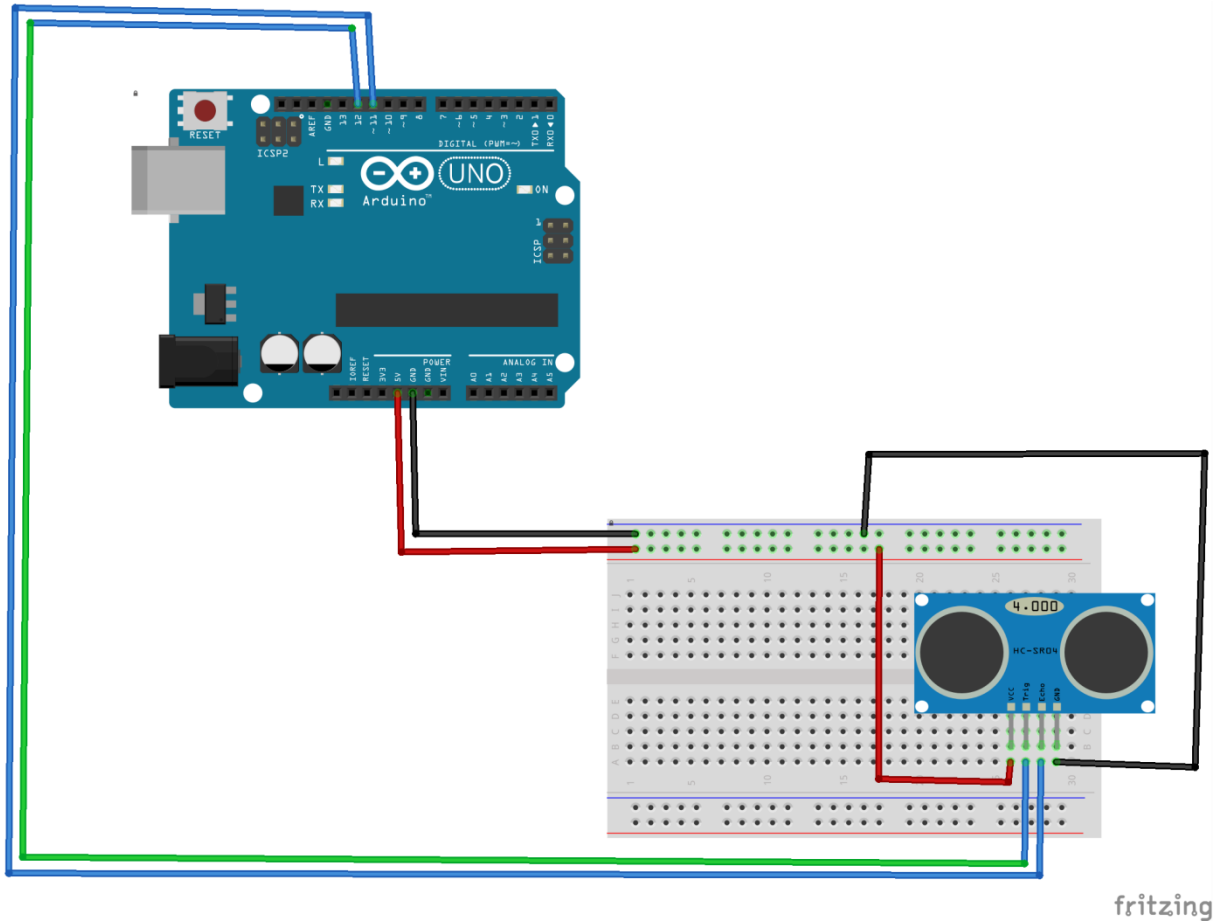


Figure III.22 : Simulation d'un capteur ultrason HC-SR04

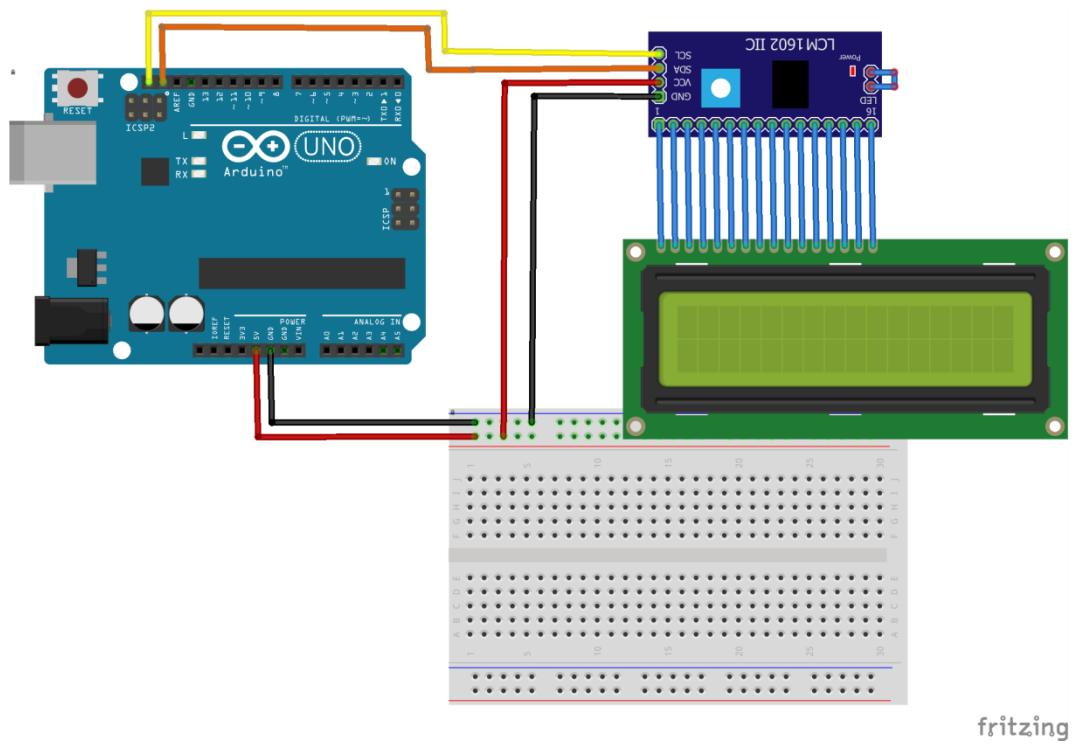


Figure III.23 : Simulation d'un afficheur LCD avec I2C

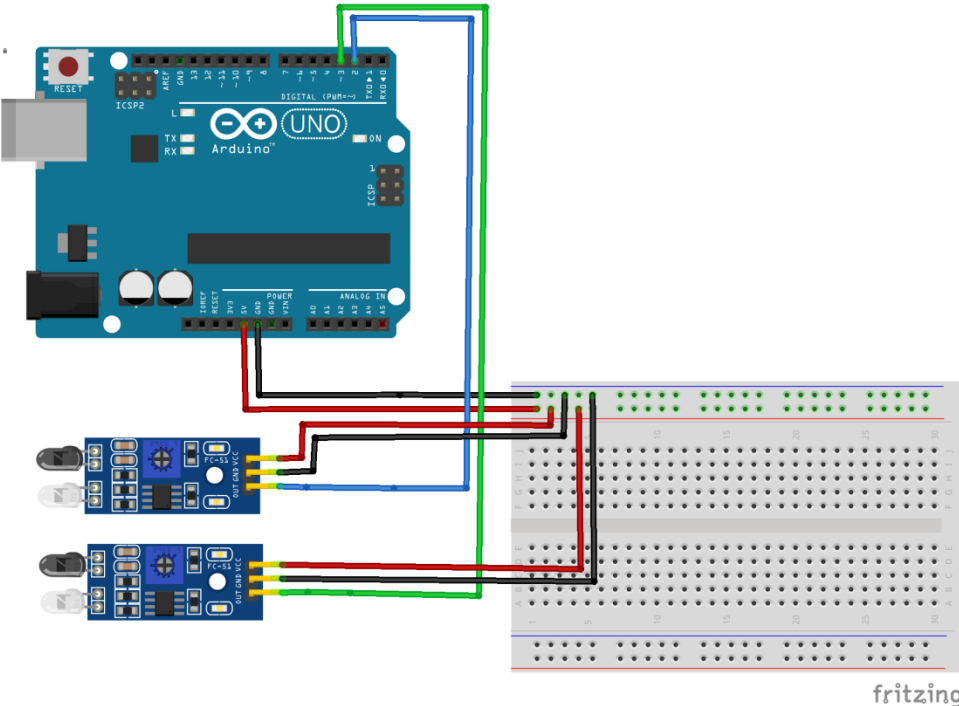


Figure III.24 : Simulation d'un capteur infrarouge TCRC5000

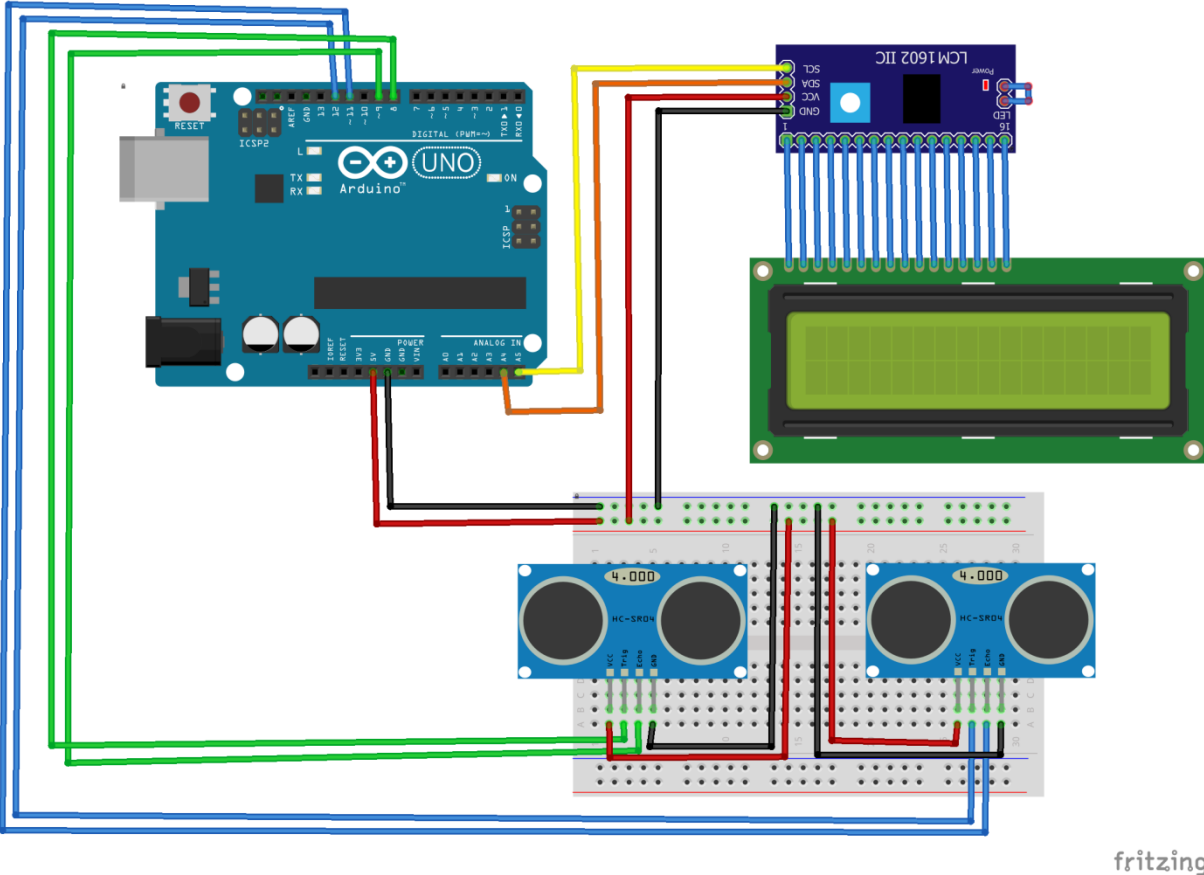


Figure III.25 : Simulation d'un système compteur

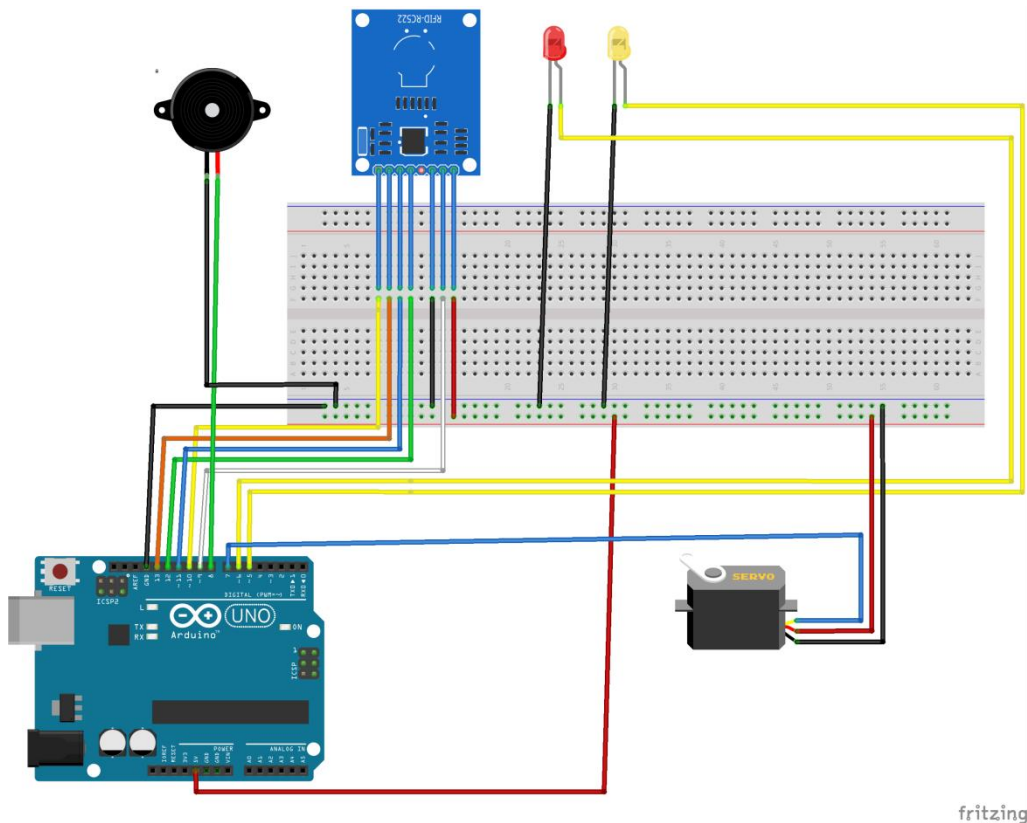


Figure III.26 : Simulation d'un système Accès RFID

III.7. Présentation des programmes sur IDE :

Le matériel simulé avec le Fritzing, prévu pour la réalisation du parking, ne peut fonctionner que si nous le relient à un ensemble d'instructions, appelé *Programme*. Nous avons travaillé avec l'Environnement de Développement Intégré, IDE ARDUINO, dont les principales fonctions sont détaillées dans les figures III.27 jusqu'à III.31

```
#include <Wire.h>
#include <LiquidCrystal_I2C.h>
LiquidCrystal_I2C lcd(0x27,16,2);
void setup()
{
  lcd.begin();
  lcd.backlight();
}
void loop()
{
  lcd.clear();
  lcd.print("Cars in:");
  lcd.setCursor(10,0);
  lcd.print(count);
  lcd.setCursor(0,1);
  lcd.print("N:");
  lcd.setCursor(3,1);
  lcd.print(acces);
  lcd.setCursor(8,1);
  lcd.print("H:");
  lcd.setCursor(12,1);
  lcd.print(val)
}
```

Figure III.27 : Sous-programme de l'afficheur LCD 16x2 avec I2C

```

#include "SPI.h"
#include "MFRC522.h"
#define SS_PIN 10
#define RST_PIN 9
int pos=0;
int acces=0;
MFRC522 rfid(SS_PIN, RST_PIN);
MFRC522::MIFARE_Key key;
void setup()
{
  Serial.begin(9600);
  SPI.begin();
  rfid.PCD_Init();
}
void loop()
{
  if (!rfid.PICC_IsNewCardPresent() || !rfid.PICC_ReadCardSerial())
    return;
  MFRC522::PICC_Type piccType = rfid.PICC_GetType(rfid.uid.sak);
  String strID = "";
  for (byte i = 0; i < 4; i++) {
    strID +=
      (rfid.uid.uidByte[i] < 0x10 ? "0" : "") +
      String(rfid.uid.uidByte[i], HEX) +
      (i != 3 ? ":" : "");
  }strID.toUpperCase();
  delay(500);
  if (strID.indexOf("F9:2F:1B:B4") >= 0 || strID.indexOf("83:95:41:16") >= 0)
  { if(acces==0){
    for(pos=0;pos<=70;pos+=1)
    {
      servo.write(pos);
      delay(20);
    }
    delay(2500);
    for(pos=70;pos>=0;pos-=1)
    {
      servo.write(pos);
      delay(20);
    }
  }
}
}
}
}

```

Figure III.28 : Sous-programme du lecteur RFID

```
#include <Servo.h>
Servo servo;
void setup() {
  servo.attach(8);
  servo.write(angle);
}
void loop()
{
  // scan from 0 to 70 degrees
  for(angle = 0; angle < 70; angle++)
  {
    servo.write(angle);
    delay(20);
  }
  // now scan back from 70 to 0 degrees
  for(angle = 70; angle > 0; angle--)
  {
    servo.write(angle);
    delay(20);
  }
}
```

Figure III.29 : Sous-programme du servo-moteur

Et voici dans les figures III.29 et II.30 les sous-programmes de la communication entre deux cartes Arduino Uno

```
void setup()
{
  Serial.begin(9600);
}
void loop()
{
  if(digitalRead(in1) == LOW)
  {
    Serial.write('3');
    delay(20);
  }
}
```

Figure III.30 : Sous-programme de l'Arduino Uno master

```
char message;
val=0;
void setup()
{
  Serial.begin(9600);
}
void loop()
{
  if (Serial.available() )
  {
    message = Serial.read();
    if(message == '1')
    {
      val=1;
    }
  }
}
```

Figure III.31 : Sous-programme de l'Arduino Uno slave

III.8. Système sur plaque d'essai :

Après des simulations sur Fritzing, nous avons procédé à des expérimentations sur une plaque d'essai (circuit complet) pour tester le fonctionnement de notre système (figure III.32). Nous avons constaté que le système de comptage des visiteurs et celui du lecteur RFID fonctionnent.

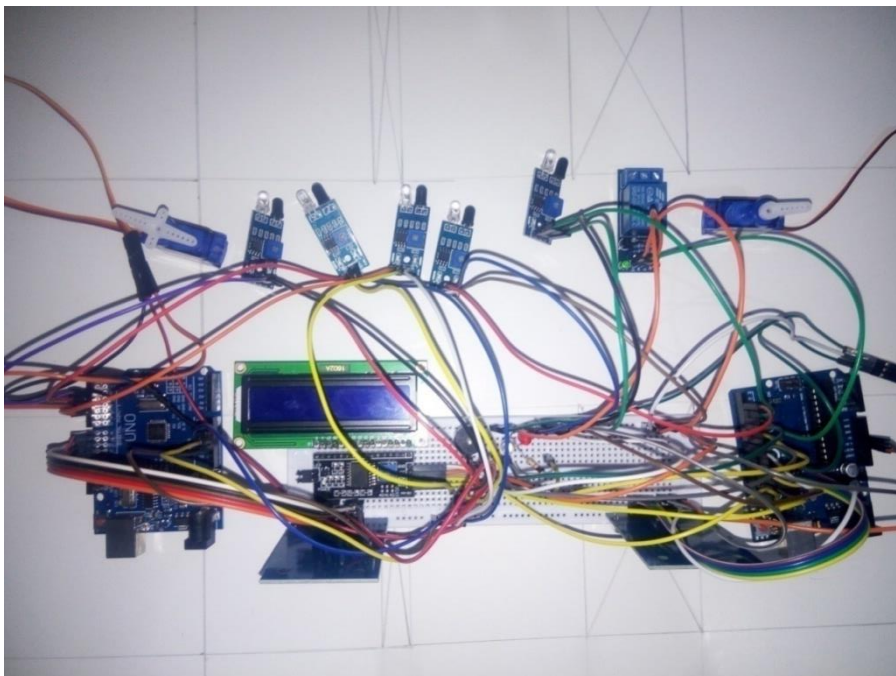


Figure III.32 : Circuit sur plaque d'essai

III.9. Implémentation sur maquette finale :

Après les essais sur Fritzing et sur la plaque d'essai, nous passons à l'implémentation de notre système sur la maquette finale, qui est une représentation miniaturisée d'un parking intelligent privé.

Cette maquette est réalisée sur un plancher de carton de dimension 65x49. Les images en figure III.33, III.34 et III.35 suivantes représentent les différentes parties de maquette.



Figure III.33 : Plancher de carton

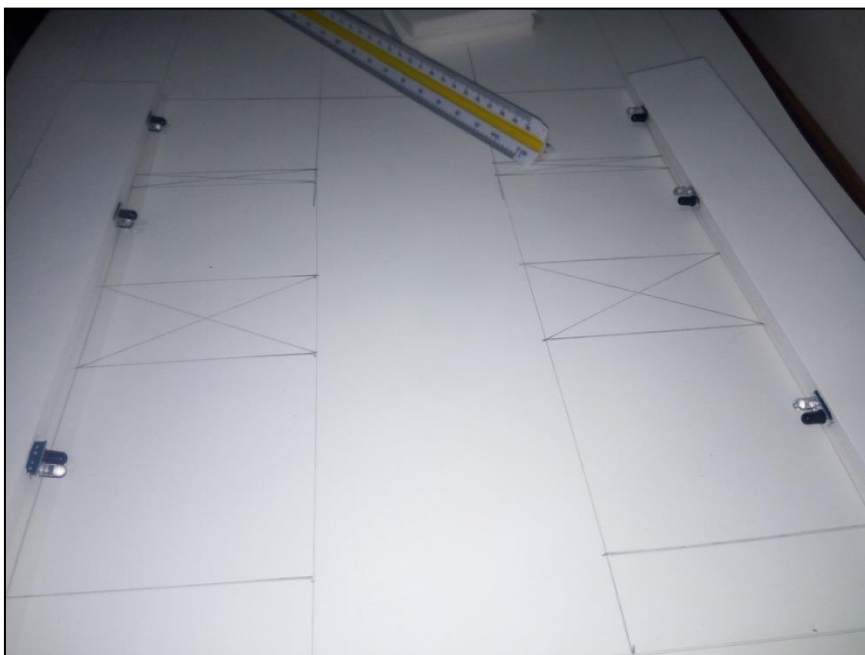


Figure III.34 : Semi plan

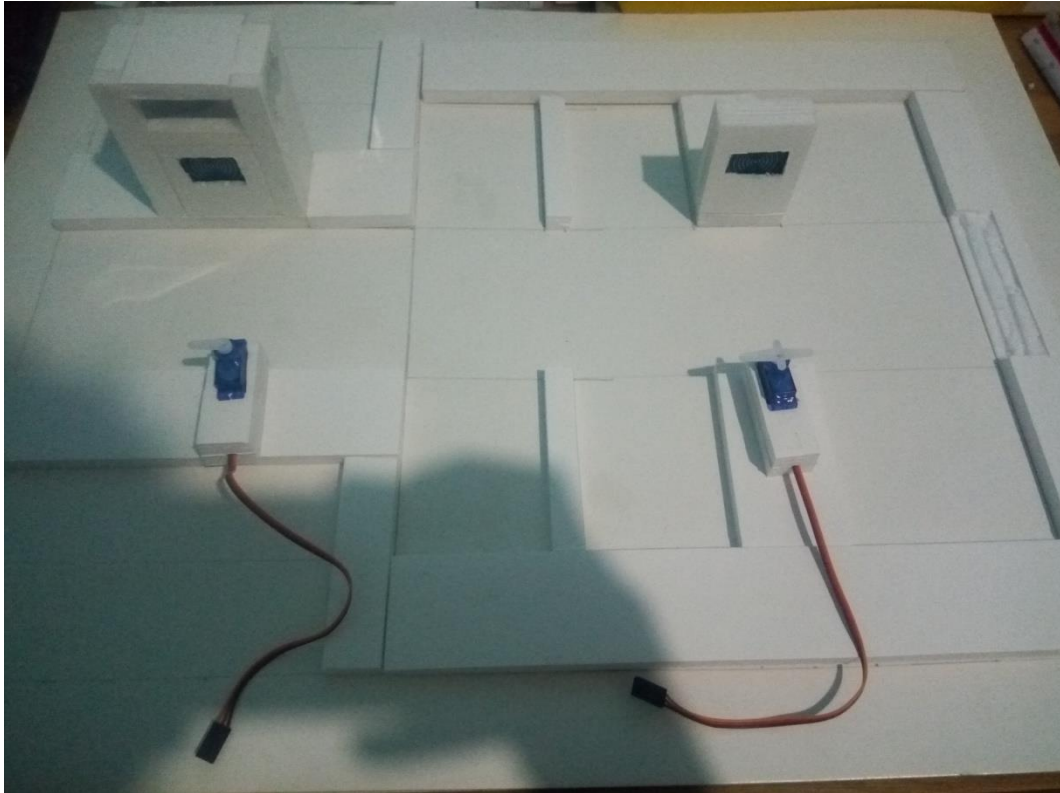


Figure III.35 : Parking semi fini

Et voici en figure III.36 le prototype final de notre parking intelligent :



Figure III.36 : Le prototype final de notre parking intelligent

III.10. Jeu d'essai :

Pour valider notre système, nous avons procédé à plusieurs jeux de test correspondant premièrement à un parking vide, puis à un parking plein. Les résultats sont très satisfaisants dans les deux cas.

✓ **L'entrée**

- **Présentation de véhicule**

Le véhicule se présente devant la barrière l'entrée le véhicule devant la barrière 1 principale de l'entrée (figure II.37).



Figure III.37 : Le véhicule devant la barrière 1 principale

- **Vérification de la présence d'une carte RFID 1**

Après la présentation de la carte RFID 1 qui se valide, la barrière principale remonte (figure III.38)



Figure III.38 : La validation de la carte RFID 1 et l'entrée du véhicule

Après l'entrée du véhicule la barrière principal descend, voici en figure III.39 la barrière principale 1 descendre.



Figure III.39 : La barrière principale 1 descendre

- **Stationnement de véhicule normal**

Si le conducteur est une personne sans limitation fonctionnelle, le véhicule stationne dans lieu de stationnement de personnes normales. Voici en figure III.40 le stationnement du véhicule normal.

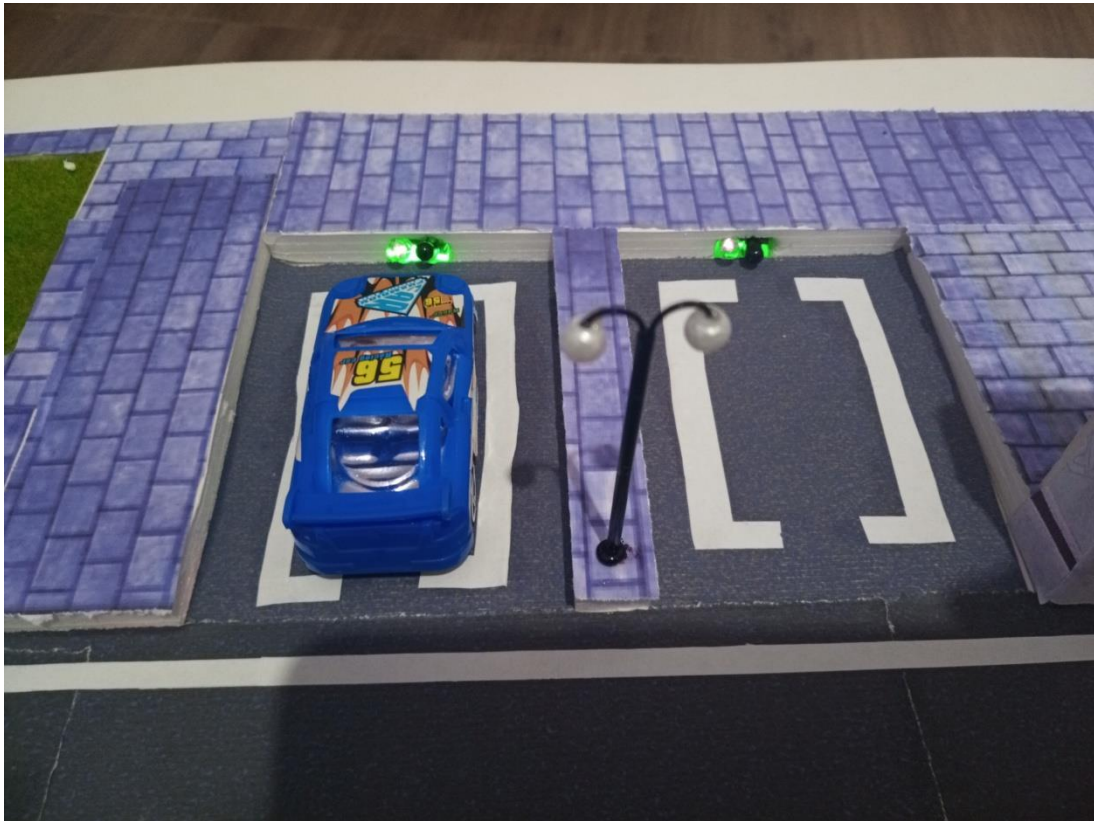


Figure III.40 : Le stationnement du véhicule normal

- **Vérification de la présence d'une carte RFID 2**

Si le conducteur est une personne handicapée, après la présentation de la carte RFID 2, elle se valide, la barrière secondaire remonte. La figure III.41 illustre la validation de la carte RFID 2 et l'entrée du véhicule.

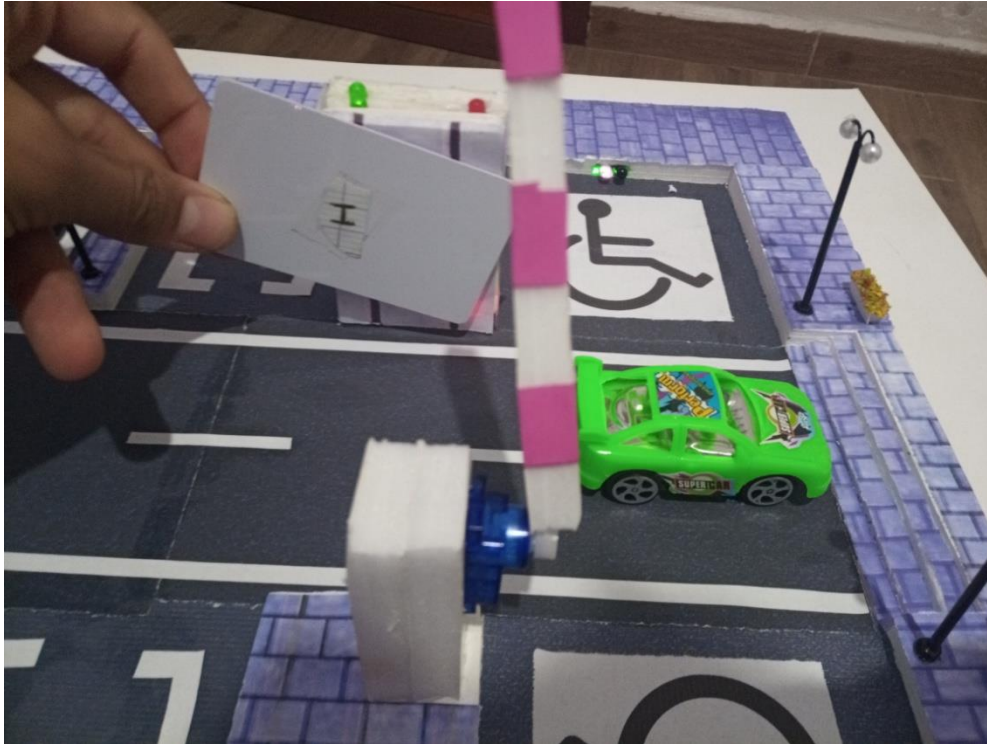


Figure III.41 : La validation de la carte RFID 2 et l'entrée du véhicule

- **Stationnement de véhicule handicapé**

Le véhicule stationne dans le lieu de stationnement des personnes en situation de handicap, comme mentionné dans la figure III.42.



Figure III.42 : Le stationnement véhicule handicapé

✓ La sortie

• La sortie de véhicule

Le véhicule se présente devant la sortie, la barrière principale ou secondaire remonte (voir figure III.43).



Figure III.43 : Le véhicule devant la barrière principale ou secondaire

III.11. Conclusion :

Dans ce chapitre, nous avons détaillé les différentes étapes de la mise en œuvre réelle du système : les organigrammes et les procédures des différents capteurs et actionneurs, et la mise en œuvre des applications d'automatisation du stationnement. Sur le plan pratique, la bonne prise en main du logiciel "Arduino" nous permet d'utiliser un compilateur ; c'est l'"IDE", qui a la capacité de créer du code HEX, qui peut être injecté dans son microcontrôleur. Compte tenu des limitations de matériels et des moyens dont nous disposons pour la conception du parking intelligent, nous pouvons conclure que les résultats obtenus sont très satisfaisants. L'architecture et le système que nous proposons ont été simulés et mis en œuvre avec succès et peuvent participer à résoudre le problème de stationnement urbain surtout pour les handicapés.

Conclusion générale

Conclusion générale

Le nombre de véhicules dans le milieu urbain continue d'avoir une croissance importante et engendre, par conséquent, un énorme problème de pollution et de stationnement. Puisque les parkings conventionnels n'arrivent plus à subvenir aux besoins des usagers et des autorités, les smart parkings ont été conçus à cette fin.

C'est dans ce contexte que nous avons inscrit l'objectif de notre projet de fin d'études pour la conception et la réalisation d'un parking intelligent. Dans ce sens, nous avons pu développer un système utilisant des outils de guidage de stationnement basés sur différents types de capteurs. En se basant sur ce dernier, les utilisateurs peuvent trouver le meilleur espace libre, tout en économisant ainsi du temps et de l'argent et en minimisant de la pollution.

On retrouve ce type de parking dans les hôpitaux, les grandes surfaces, les résidences...etc. La circulation devient plus fluide car moins de voitures chercheront des places de stationnement ouvertes. La création de parkings intelligents continue d'évoluer avec le développement de la ville. Outre le développement continu de la technologie des capteurs qui est au cœur du développement du stationnement intelligent, il existe diverses autres créations technologiques qui rendent possibles des systèmes plus flexibles, tels que les caméras, les communications sans fil, l'analyse de données et les algorithmes avancés.

Dans nos recherches futures, nous examinerons les aspects de sécurité de notre système et la mise en œuvre à grande échelle du système proposé dans le monde réel à l'aide de pose d'une puce directement sur la voiture. Ainsi au moment de pénétrer dans le parking, le lecteur placé au niveau de la barrière détecte directement la voiture et ouvre la barrière si la voiture qui tente d'entrer a un abonnement par exemple et ceci afin d'éviter à l'utilisateur de sortir une carte à chaque entrée ou sortie du parking. Cela représente un gain de temps pour l'utilisateur et permet également d'éviter d'éventuels bouchons. Aussi, la reconnaissance des plaques d'immatriculation à travers un système d'intelligence et de vision artificielles peut augmenter aussi la sécurité et la reconnaissance. Un système de comptage et d'archivage sera aussi la bienvenue pour stocker toute trace de passage dans le parking dont les informations personnelles, date d'entrée et de sortie, durée de stationnement, etc...

Bibliographie

- [1] <https://www.afnic.fr/fr/expertises/labs/projets-realises/l-internet-des-objets-projets-wingset-proxi-produit.html>
- [2] Jean-Pierre Hauet « L'Internet des objets deux technologies clés les réseaux de communication et les protocoles », Deuxième partie, REE N°5/2016
- [3] <http://e-biblio.univ-mosta.dz/handle/123456789/13120>
- [4] <http://carfree.fr/index.php/2010/08/27/histoire-du-parking/>
- [5] <https://www.groupe-pigeon.com/notre-groupe/notre-actualite/quels-sont-les-differents-types-de-parking>
- [6] <https://fr.wikipedia.org/wiki/Parking>
- [7] <http://www.generationrobots.com/fr/152-arduino>.
- [8] <http://fr.flossmanuals.net/arduino/index>
- [9] C. Tavernier, « Arduino applications avancées ». Version Dunod.
- [10] S.V.D.Reyvanth, G.Shirish, « PID controller using Arduino »
- [11] www.itechnofrance.wordpress.com consulter le : Juillet/201
- [12] <https://www.moussasoft.com/product/module-rfid-rc522-lecteur-rfid>
- [13] https://abra-electronics.com/images/detailed/141/SENS-TCRT5000_1.jpg
- [14] Hippolyte Weisslinger (olyte), Landrault (Eskimon) Arduino : Premiers pas en informatique embarquée, le Blog d'Eskimon Edition du 19 juin 2014
- [15] <http://itechnofrance.wordpress.com/2013/05/05/utilisation-du-servomoteur-sg90-avec-l-arduino/>
- [16] <https://www.lextronic.fr/module-relais-5v-40436.html>
- [17] <https://fritzing.org>
- [18] RFID/NFC 13.56MHz Networking Guide – Libelium
- [19] http://rtmfm.cnrs.fr/IMG/pdf/fiche_6_branchement_et_utilisation_d_un_ecran_LCD.pdf
- [20] <https://www.orbit-dz.com/product/convertisseur-i2c-lcd-module/>
- [21] <http://www.robot-maker.com/shop/img/cms/datasheet-capteur-ultrasons-hc-sr04.pdf>
- [22] <https://www.arduino.cc/en/software/>

[23] <https://pecquery.wixsite.com/arduino-passion/le-buzzer>

[24] <https://parlonssciences.ca/ressources-pedagogiques/documents-dinformation/lintelligence-artificielle-et-la-vision-par>