REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE
MINISTERE DE L'ENSEIGNEMENT SUPERIEUR ET DE LA RECHERCHE SCIENTIFIQUE

UNIVERSITE DR. YAHIA FARES DE MEDEA
FACULTE DE TECHNOLOGIE
DEPARTEMENT DE GENIE MECANIQUE



Mémoire de fin d'étude
En vue de l'obtention de diplôme de Master en Electromécanique
Spécialité : Mécatronique

Thème :

Conception et réalisation d'un smart parking à base d'arduino

Présenté par :

- Betol Sid ahmed
- Reghmit Houcine

Devant le jury composé de :

Dr. ALLICHE Mounir	MCA	Univ. Médéa	Président
Dr. REBHI Redha	MCA	Univ. Médéa	Promoteur
Dr. BOUMAHDI Mouloud	MCA	Univ. Médéa	Examinateur

Année universitaire : 2021/2022

Remerciement

Avant tout, nous remercions ***Allah*** *de nous avoir gardé en bonne santé et donné le courage pour mener à bien ce projet de fin d'études. Nous remercions également nos familles pour les sacrifices qu'elles ont consentis afin que nous puissions terminer nos études.*

Nous remercions cordialement Monsieur ***Dr. REBHI Redha*** *maitre assistant de classe « A » à l'université* ***Dr. Yahia fares Medea*** *pour nous avoir encadrés.*

De même, nous remercions Monsieur ***Dr. ALLICHE Mounir*** *qui nous a fait l'honneur de présider le jury pour la soutenance de notre projet de fin d'étude.*

Nous remercions également Monsieur ***Dr. BOUMAHDI Mouloud*** *l'examinateur, qui a eu l'honneur d'accepter nos sympathies pour siéger dans notre jury.*

Nos remerciements vont également à tous nos professeurs, pour nous avoir formés durant nos études.

Nous tenons à remercier en particulier tous ceux qui ont contribué à finaliser ce modeste travail qui, nous l'espérons, sera à la hauteur de leur engagement.

Dédicace

A nos chers parents

Pour leur soutien, leur patience, leur sacrifice

Et leur amour, vous méritez tout éloge,

Vous qui avez fait de nous ce que nous sommes maintenant.

Nous espérons être comme vous l'avez souhaitez.

Que dieu vous garde et vous bénisse.

Nous dédions aussi ce travail à nos chers frères et sœurs,

Pour leurs affections et leur encouragement

Qui ont toujours été pour nous des Plus précieux.

A ceux qui ont sacrifié, patienté, donné tout pour continuer leurs études.

Pour être là entre nous, apprendre plus et plus pour éclairer le monde par leur savoir,

À tous mes amis à tous ce qui nous a aidés.

A tous ceux que nous aimons nous dédions ce travail.

Sidahmed et Houcine

Sommaire

Introduction générale 1

Chapitre II : Généralités sur les parkings

I.1 Introduction 2
I.2 Historique 2
I.3 Problématique de stationnement 3
I.4 Définition du Parking 4
I.5 Les différents types de parking 4
I.5.1 Les parkings en surface 4
I.5.2 Les parkings fermé ou souterrains 5
I.5.3 Les parkings aériens à étages 6
I.5.4 Les parkings en superstructure largement ventilé (PSLV) 6
I.5.5 Les parcs relais 7
I.6 Système Parking automatisé des voitures 8
I.7 Classification par mode de fonctionnement 8
I.7.1 Systèmes de stationnement semi-automatisés 8
I.7.2 Systèmes de stationnement entièrement automatisés 9
I.8 Estimation d'évolution de l'utilisation des SPA 9
I.9 Pourquoi louer une place de parking ? 9
I.10 Gestion des parkings 9
I.11 Les avantages et les inconvénients de parking automatisés 10
I.11.1 Les avantages 10
I.11.2 Les inconvénients 11
I.12 Conclusion 11

Chapitre II : Description du matériels et logiciels utilisés

II.1 Introduction 12
II.2 Partie matérielle 12
II.2.1 Les composants électroniques utilisés 12
II.2.2 La carte Arduino 12
II.2.2.1 Les différentes cartes Arduino 12
II.2.2.2 Aperçu sur Arduino Uno 14

II.2.2.3 Caractéristiques de l'Arduino Uno 14
II.2.2.4 Configuration du brochage de l'Arduino Uno 14
II.2.3 Microcontrôleur ATmega328 17
II.2.4 Servomoteur 17
II.2.4.1 Description de servomoteur 18
II.2.4.2 Programmation de servomoteur 18
II.2.4.3 Schéma de câblage avec Arduino 19
II.2.5 Capteur IR (FC-51) 19
II.2.5.1 Description de capteur IR 20
II.2.5.2 Principe de fonctionnement 21
II.2.5.3 Schéma bloc d'un capteur IR 22
II.2.5.4 Schéma de câblage avec arduino 23
II.2.6 Capteur LDR 23
II.2.6.1 Principe de fonctionnement 24
II.2.6.2 Caractéristiques d'un capteur LDR 24
II.2.6.3 Les différents types de capteur LDR 24
II.2.6.4 Schéma de câblage avec Arduino 25
II.2.7 Capteur DHT 11 25
II.2.7.1 Description sur DHT 11 25
II.2.7.2 Principe de fonctionnement 26
II.2.7.3 Caractéristiques techniques 26
II.2.7.4 Schéma de câblage avec Arduino 27
II.2.7.5 Applications 27
II.2.8 Écran LCD I2C 16X2 avec rétro-éclairage bleu 27
II.2.8.1 Présentation du matériel de l'écran LCD I2C 28
II.2.8.2 Adresse I2C du LCD 28
II.2.8.3 Schéma de câblage avec Arduino 29
II.2.9 Système RFID 29
II.2.9.1 Description de la carte RFID RC522 30
II.2.9.2 Principe de fonctionnement d'un système RFID 31
II.2.9.3 Schéma de câblage avec Arduino 31
II.2.9.4 Les Applications 32
II.2.10 Buzzer 33
II.2.10.1 Principe de fonctionnement 33
II.2.10.2 Schéma de câblage avec Arduino 33

II.2.11 Breadboard 34
II.3 Partie logicielle 35
II.3.1 Arduino IDE 35
II.3.1.1 L'environnement de développement (IDE) de l'Arduino 35
II.3.1.2 Structure générale du programme (Arduino IDE) 36
II.3.1.3 Description d'un programme Arduino 37
II.3.1.4 Injection du programme 37
II.3.1.5 Les étapes de téléchargement du programme 39
II.3.2 logiciel PROTEUS 39
II.3.2.1 ISIS (Intelligent Schématique Input System) 40
II.3.2.2 ARES 41
II.3.3 Fritzing 41
II.3.4 SolidWorks 42
II.4 Conclusion 43

Chapitre III : Conception et Réalisation

III.1 Introduction 44
III.2 Conception du parking 44
III.2.1 Schémas d'interconnexion des composants 45
III.2.2 Schéma global du système sur le logiciel ISIS 46
III.2.3 Contrôle de système 46
III.2.3.1 Contrôle de température et d'humidité 46
III.2.3.2 Contrôle d'accès au parking 47
III.3 Réalisation du parking 48
III.3.1 Description du prototype 48
III.3.1.1 Bloc d'accès au parking 48
III.3.1.2 Bloc de la gestion du parking 49
III.3.1.3 Bloc d'affichage des informations 49
III.3.2 Comment fonctionne ce système ? 49
III.4 Conclusion 56
Conclusion générale 57
Bibliographie
Annexe

Liste des figures

Chapitre I : Généralité sur le parking		
Figure	**Titre**	**Page**
Figure I.1	Parking Ponthieu.	**2**
Figure I.2	Parking Paternoster.	**2**
Figure I.3	La circulation dans une ville avec et sans le stationnent intelligent.	**3**
Figure I.4	Parc de stationnement.	**4**
Figure I.5	Parking en surface.	**5**
Figure I.6	Parking fermé où Souterrains.	**5**
Figure I.7	Parking aériens à étage.	**6**
Figure I.8	Parking PSLV.	**7**
Figure I.9	Parking relais.	**7**

Chapitre II : Description du matériel et logiciels utilisés		
Figure	**Titre**	**Page**
Figure II.1	Les différentes cartes Arduino.	**13**
Figure II.2	Une Carte Arduino Uno328.	**14**
Figure II.3	Alimentation de l'Arduino Uno328.	**15**
Figure II.4	Broches d'E/S numériques de l'Arduino Uno328.	**15**
Figure II.5	Broches d'E/S analogiques de l'Arduino Uno328.	**16**
Figure II.6	Broches RESET / AREF de l'Arduino Uno328.	**16**
Figure II.7	Microcontrôleur ATmega328.	**17**
Figure II.8	Servomoteur "9 grammes".	**17**
Figure II.9	Structure de Servomoteur "9 grammes".	**18**
Figure II.10	Connexion servomoteur avec la carte Arduino Uno.	**19**
Figure II.11	Capteur IR (FC-51).	**19**
Figure II.12	Structure d'un Capteur IR.	**20**
Figure II.13	Principe de fonctionnement d'un capteur IR.	**21**
Figure II.14	Émetteur IR.	**21**
Figure II.15	Récepteur IR.	**22**
Figure II.16	Schéma bloc d'un capteur IR.	**22**
Figure II.17	Connexion de capteur IR avec la carte Arduino Uno.	**23**
Figure II.18	Capteur LDR.	**23**
Figure II.19	Connexion LDR avec la carte Arduino Uno.	**25**
Figure II.20	Capteur DHT 11.	**26**
Figure II.21	Connexion DHT11 avec la carte Arduino Uno.	**27**
Figure II.22	Structure d'un écran LCD-I2C (16*2).	**28**
Figure II.23	Connexion un écran LCD I2C avec la carte Arduino Uno.	**29**
Figure II.24	La carte RFID RC522 et les tags.	**30**

Figure II.25	Schéma de principe de système RFID.	**31**
Figure II.26	Connexion du module RFID RC522 ave la carte Arduino Uno.	**32**
Figure II.27	Buzzer.	**33**
Figure II.28	Connexion buzzer avec Arduino Uno.	**33**
Figure II.29	Breadboard.	**34**
Figure II.30	Fonctionnement d'un programme Arduino.	**35**
Figure II.31	L'interface de l'IDE de l'Arduino.	**36**
Figure II.32	Barre de bouton.	**36**
Figure II.33	Paramétrage de la carte étape 1.	**38**
Figure II.34	Paramétrage de la carte étape 2.	**38**
Figure II.35	Les étapes de téléchargement du code.	**39**
Figure II.36	L'interface de logiciel ISIS Proteus.	**40**
Figure II.37	L'interface de logiciel Fritzing.	**41**
Figure II.38	L'interface de logiciel Solidworks.	**42**

Chapitre III : Conception et réalisation

Figure	**Titre**	**Page**
Figure III.1	Structure du plan de parking.	**44**
Figure III.2	Schéma synoptique du système.	**45**
Figure III.3	Schéma global du système sur le logiciel ISIS.	**46**
Figure III.4	Schéma interconnexions entre les composants du système d'acquisition de température et d'humidité.	**47**
Figure III.5	Schéma interconnexions entre les composants du système du control d'accès.	**47**
Figure III.6	Représentation schématique du prototype.	**48**
Figure III.7	Affiche un message d'accueil « Bienvenu Smart parking ».	**49**
Figure III.8	Nombre de places libres et la température.	**50**
Figure III.9	Présenter sa carte RFID valide à l'entrée.	**51**
Figure III.10	Détection d'une voiture en cas d'entrée.	**51**
Figure III.11	L'afficheur LCD en cas d'entrée d'une voiture.	**52**
Figure III.12	Détection d'une voiture en cas de sortie.	**53**
Figure III.13	L'afficheur LCD en cas de sortie deux voiture.	**53**
Figure III.14	La détection d'incendie.	**54**
Figure III.15	Affichage du message d'alerte en cas d'incendie.	**55**
Figure III.16	représentation globale du smart parking.	**55**
Figure III.17	représentation globale de circuit du smart parking.	**56**

Liste des tableaux

Tableau II.1	Les composants électroniques utilisés.	**12**
Tableau II.2	Les caractéristiques de la carte Arduino Uno.	**Annexe**
Tableau II.3	Sortie et entrée d'un Capteur IR.	**20**
Tableau II.4	Les caractéristiques de capteur LDR.	**24**
Tableau II.5	Les spécifications de la carte RFID RC522.	**30**
Tableau III.1	Le brochage du module RC522 avec Arduino Uno.	**50**
Tableau III.2	Branchement des capteurs LDR avec la carte Arduino Uno.	**52**

Liste des abréviations

Abréviation	Signification
PSLV	Parking en superstructure largement ventilé.
SPA	Système parking automatisé.
GPS	Global Positioning System.
IR	Infrarouge.
LED	Light Emmiting Diode.
LDR	Light Dependent Resistor.
DHT	Détecteur Humidité et Température.
LCD	Liquid Crystal Display.
RFID	Radio-Fréquence Identification.
IDE	Integrated Development Environnement.
CAO	Conception assisté par ordinateur.
ISIS	Intelligent Shématique Input System.
PCB	Printed Circuit board.
3D	Tradimensionnel.

Abstract

In recent years, the urbain population is high all over the world which has led to many problems for cities, with car parking being one of the major problems that cause congestion. To find solutions and eliminate parking problems, parking must be provided, organized and controlled to ensure good traffic.

An automatic car park is an alternative to a conventional car park, giving a good overview of the available space so this system is the ideal solution to automate the entry and exit process and good parking without problems.

In this dissertation, we realized a smart parking (smart parking) using an Arduino Uno board that controls the différent peripherals such as the RFID system, servomotor, the LCD screen and the alarm system (Buzzer).

If a véhicule is detected using an infrared sensor the barrier is opened and closed, with a view of the number of available spaces and the température on the LCD screen and a ban on access if the parking is full.

Only subscribed customers who are equipped with a specific magnetic card (RFID) with a special serial number and code can be passed ...

Keywords : Smart parking, Parking, Arduino Uno, RFID system, Servomotor, Barrier, LCD screen, IR sensor, Buzzer.

Résumé

Ces dernières années, la population urbaine est élevée dans tout le monde ce qui a conduit de nombreux problèmes pour les villes, le stationnement des voitures étant l'un des problèmes majeurs qui causent de congestion. Pour trouver des solutions et éliminer les problèmes de stationnement, un parking doit être fourni, organisé et contrôlé pour assurer une bonne circulation.

Un parking automatique est une alternative au un parking classique, donnant un bon aperçu de l'espace disponible donc ce système est la solution idéale pour automatiser le processus d'entrée et de sortie et un bon stationnement sans problème.

Dans ce mémoire, nous avons réalisé un smart parking (parking intelligent) en utilisant une carte Arduino Uno qui contrôle les différents périphériques tels que le système RFID, servomoteur, l'écran LCD et le système d'alarme (Buzzer).

Si un véhicule est détecté à l'aide d'un capteur infrarouge la barrière est ouverte et fermée, Avec une vue sur le nombre de places disponibles et la température sur l'écran LCD et une interdiction d'accès si le parking est complet.

Seulement les clients abonnés qui sont équipés d'une carte magnétique spécifique (RFID) avec un numéro de série spécial et un code peuvent être passées …

Mots clés : Smart parking, Le stationnement, Arduino Uno, Système RFID, Servomoteur, Barrière, L'écran LCD, capteur IR, Buzzer.

ملخص

في السنوات الأخيرة، تزايد سكان المدن بسرعة في جميع أنحاء العالم مما أدى الى خلق العديد من المشاكل للمدن ،حيث تعد مواقف السيارات واحدة من المشاكل الرئيسية التي تسبب الازدحام. وللاستجابة ولإيجاد حلول للقضاء على مشاكل وقوف وركن السيارات، يجب توفير موقف لوقوف السيارات وتنظيمها والتحكم فيها ولضمان السير الحسن لها.

موقف السيارات الاوتوماتيكي يمثل بديلا لموقف السيارات الكلاسيكي، والذي يعطي نظرة عامة جيدة للمساحة المتاحة لذلك، هذا النظام هو الحل المثالي لإتمام عملية الدخول والخروج وركن السيارات السليم بدون أي فوضى.

في هذه الأطروحة، لتحقيق موقف سيارات ذكي قمنا باستخدام لوحة Arduino Uno التي تتحكم في الأجهزة الطرفية المختلفة مثل نظام RFIDومحرك المؤازرة وشاشة LCD ونظام الإنذار(الجرس).

إذا تم الكشف عن وجود سيارة باستخدام جهاز استشعار الأشعة تحت الحمراء يتم فتح الحاجز وإغلاقه، مع عرض عدد الأماكن المتاحة ودرجة الحرارة على شاشة LCD وحظر الوصول إذا كان موقف السيارات ممتلئا.

يمكن فقط تمرير العملاء المشتركين المزودين ببطاقة ممغنطة (RFID)والتي تكون مرفقة برقم تسلسلي خاص ورمز...

الكلمات المفتاحية: موقف سيارات ذكي، ركن السيارات، لوحة Arduino Uno، نظام RFID، محرك المؤازرة، حاجز، شاشة LCD ، الجرس.

Introduction générale

Le nombre de véhicules augmente chaque jour dans le monde entier. Cela cause le problème de la congestion et notre smart parking est parmi les facteurs les plus importants pour régler cette situation.

Le parking est un sujet qui a toujours été très présent parce qu'il touche la mobilité de la plupart des personnes dont Les conducteurs doivent toujours se stationner pour se rendre à leurs lieux de destination.

Les conducteurs avaient de réels problèmes pour trouver facilement une place de stationnement et la demande de cette dernière est généralement beaucoup plus importante que l'offre , un problème de sécurité , la perte de temps et un problème de pollution se pose également quand les conducteurs cherchent une place de stationnement.

Pour surmonter les problèmes ci-dessus, il faut un système avancé qui assure Une bonne gestion de parking (Temps, Sécurité, Paiement automatisé, etc…), Aménagement et amélioration de la surface de parking. C'est dans cette optique que s'inscrit ce projet.

Ce mémoire comporte trois chapitres, à travers lesquels nous décrivons le travail effectué pour la conception et la réalisation de notre système.

Dans le premier chapitre, nous présentons les généralités sur le smart parking, qui consiste à donner un aperçu sur le parking (la définition, les déférentes types, les avantages et les inconvénients).

Le deuxième chapitre présente une description des matériels et des logiciels utilisé pour la mise en œuvre de notre système.

Le troisième chapitre est consacré à la conception de notre système, ainsi qu'a sa réalisation.

CHAPITRE I

Généralités sur les parkings

I.1 Introduction

Depuis la naissance de la voiture jusqu'à aujourd'hui, il y a une augmentation importante de production vu qu'elle rend beaucoup de service, et facilite le déplacement dans plusieurs domaine (commerce, Transport, etc…).

Le fait que la production de voiture ne cesse d'augmenter le besoin d'espace pour leur stationnement dans les parkings publics a augmenté aussi, pour cela on va étudier dans la première partie les différents types de parking dont l'objectif est de minimiser l'espace et assuré des bonnes conditions de stationnement (la sécurité, le temps, etc…).

I.2 Historique

Bien que Les systèmes de stationnement automatisés puisse sembler être une invention récente, l'histoire du **stationnement automatisé** remonte à plus de 100 ans. Nés en France au début du siècle dernier, leur histoire peut être marquée par trois moments importants :

– En **1905**, à l'occasion de l'Exposition universelle, le Garage **Rue de Ponthieu,** premier parking automatisé**,** est conçu à Paris. La force motrice était le besoin d'augmenter de façon drastique la capacité d'une petite zone, répondant ainsi aux demandes du marché de l'époque.

– L'année **1920** voit l'émergence du design de la grande roue, également connu sous le nom de « **Paternoster** ». La structure permettait de garer jusqu'à 8 véhicules dans un seul compartiment, optimisant ainsi grandement l'espace.

– En **1950** à Washington DC, l'avancée technologique donne naissance au premier parking automatisé sans chauffeur, un modèle qui s'imposera rapidement dans le monde entier.

Aujourd'hui, les systèmes de stationnement automatisés répondent principalement aux besoins des zones urbaines et suburbaines, où la demande de zones de stationnement est considérablement plus élevée que l'espace disponible. **[1]**

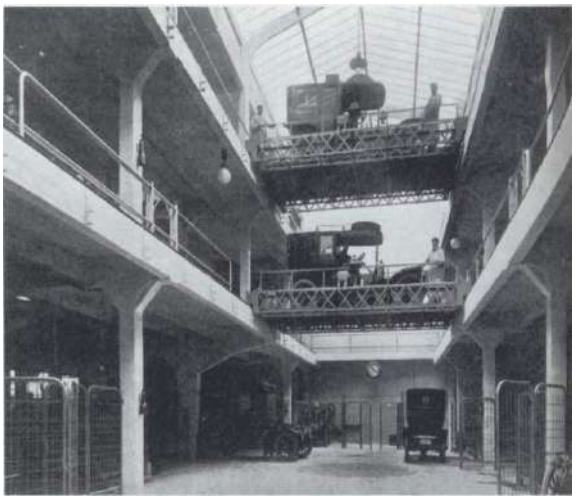


Figure I.1 : parking Ponthieu.



Figure I.2 : parking Paternoster.

I.3 Problématique de stationnement

La propagation du chaos et de la pollution dans le monde en raison du grand nombre de voitures, de l'expansion des villes et du manque de places de stationnement nous avons conduit à la spéculation et la mise en place d'autres terrains de stationnement.

Le stationnement n'est pas seulement pour les propriétaires du quartier ou de la ville, mais pour le grand public et la durée de l'arrêt n'est pas spécifique, et c'est ce qui provoque les problèmes.

Malgré la présence de parkings il y a beaucoup d'accidents et de problèmes qui se produisent entre les conducteurs. Pour réduire ces problèmes le système de stationnement intelligent Réduit ces problèmes car il aide les touristes et les employés ou toute personne qui ne connaît pas la ville a réservé une place pour sa voiture avec facilité et fiabilité. **[2]**



Sans le smart parking



Avec le smart parking

Figure I.3 : La circulation dans une ville avec et sans le stationnent intelligent.

I.4 Définition du Parking

Un parking, ou parc de stationnement est un espace ou un bâtiment spécifiquement aménagé pour le stationnement des véhicules. 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.



Figure I.4 : parc de stationnement.

I.5 Les différents types de parking

Un parc de stationnement est un lieu spécialement aménagé pour le garage des automobiles. On distingue :

I.5.1 Les parkings en surface

Le parking de surface se situe de plain-pied, à l'extérieur, sur l'espace public ou privé. Ce type de parking comprend le stationnement en voirie (places le long d'une rue, d'un quai, etc.) et les espaces dégagés à cette fin entre des bâtiments, ou établis sur des anciens champs, des anciens terrains vagues, etc.



Figure I.5 : parking en surface.

III.5.2 Les parkings fermé ou souterrains

En ville ou sous les aéroports, sous les bâtiments de certaines zones d'activité, souvent sur plusieurs niveaux, le parking souterrain permet d'économiser le foncier. Chaque niveau de stationnement s'apparente à un parking classique, à la différence que l'air y est plus confiné et pollué, que le sol n'y est pas lessivé par les pluies, qu'il peut être recouvert d'un revêtement particulier et que l'on y retrouve des piliers à intervalles réguliers pour soutenir la structure. Des rampes permettent de passer d'un niveau à l'autre. Des ascenseurs ou des escaliers permettent aux occupants des véhicules, une fois ceux-ci garés, de remonter à la surface.

Dans plusieurs pays (dont la France), les parkings souterrains sont désormais obligatoires pour toutes les constructions d'immeubles d'habitation dans certains zonages urbains, avec des prescriptions en matière d'aération, lutte contre l'incendie, sorties de secours, etc.



Figure I.6 : parking fermé où Souterrains.

I.5.3 Les parkings aériens à étages

Un parking aérien à étages est un bâtiment construit en élévation à l'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.
Il existe différents types de parkings à étages selon le mode de construction (en béton, à structure métallique, ou mixte béton/métal), la configuration des rampes d'accès et de circulation entre les différents niveaux.

Du fait de leur forme, les parkings à étages sont également appelés parkings silo même lors de construction semi-enterrées.



Figure I.7 : parking aériens à étage.

I.5.4 Les parkings en superstructure largement ventilé (PSLV)

Un parking en superstructure largement ventilé est un parking dont la ventilation est assurée par des baies latérales de ventilation. La ventilation y est naturelle, elle permet les échanges d'air extérieur et intérieur sans participation mécanique. On parle également de parking ouvert. Un des grands avantages du parking PSLV est l'économie d'énergie électrique en lumières car on profite de la clarté extérieure via les ouvertures en façade.

A noter que, depuis l'arrêté du 9 mai 2006 relatif aux parcs de stationnement ouverts, la loi autorise la construction en ossature métallique des Parcs de Stationnement en Superstructure Largement Ventilés (PSLV) avec un cahier des charges précis imposés par la législation (Parking obligatoirement aérien, 50 % d'ouverture sur les façades, 5 % sur le plancher, Distance entre façades opposées maximum : 75 m).

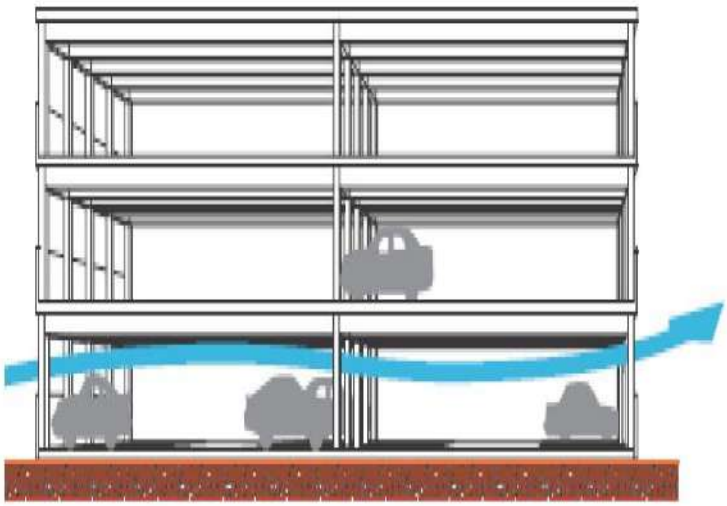


Figure I.8 : parking PSLV.

I.5.5 Les parcs relais

Les parcs relais (P+R) se trouvent en périphérie des villes. Ces parkings sont subventionnés par les collectivités car ils favorisent la mobilité générale et la mutualisation des moyens de transports publics ou privés notamment avec la pratique du co-voiturage. Les parcs relais sont généralement positionnés aux abords immédiats des villes et à proximité des gares routières ou ferroviaires, des stations de métro ou de tramway, des arrêts de bus.

La logique de cette implantation est de favoriser la mixité des moyens de transports (dont le co-voiturage) afin de désengorger la circulation dans les centres urbains et de réduire l'impact écologique et économique de l'automobile (réduction de la pollution de l'air, réduction des dépenses en essence). **[3]**



Figure I.9 : parking relais.

I.6 Système Parking automatisé des voitures

Parking automatisé est une méthode de garer et récupérer les voitures automatiquement qui utilisent généralement un système de palettes et des ascenseurs. L'intention est de voitures compactes plus dans le même espace de stationnement. Parkings automatisés peuvent être situés au-dessus ou au-dessous du sol ou une combinaison des deux. Le système automatisé de Parking alimentés sur le courant électrique pour automatiquement garer et récupérer les voitures, la distance parcourue par chaque véhicule est réduite. Selon le cabinet de consultants environnementaux Waterman Group, les parcs de stationnement utilisant un système robotisé pour le rangement des véhicules génèrent moins de CO2 que les parcs à étages classiques.

Le Parking automatisé doit aussi varier les moyens de paiement. Par exemple, tout le monde n'a pas forcément de monnaie au moment de payer le parking. Il faut développer les paiements par carte ou à l'aide des Smartphones. L'identification des véhicules entrant dans le parking, ainsi aucun paiement n'est demandé, un capteur détecte la voiture et son identifiant à l'entrée et à la sortie du parking et calcule donc automatiquement le prix à payer par le conducteur. Une facture mensuelle ou trimestrielle peut donc être mise en place.

Au fil des années, le monde automobile évolue et, on voit apparaître de plus en plus de voitures électriques. Ces voitures électriques sont bénéfiques contre la pollution mais il faut qu'elles soient rechargées. C'est pourquoi des zones de rechargement doivent être prévues, et plus spécialement des places de parking où l'on peut les recharger. En effet, il serait très avantageux que l'utilisateur puisse recharger sa voiture pendant son stationnement. **[4]**

I.7 Classification par mode de fonctionnement

Les systèmes intelligents de stationnement peuvent être semi-automatisés ou entièrement automatisés. Le choix entre l'SPA entièrement et semi-automatisé est souvent une question d'espace et de coût, mais les grandes capacités (plus de 100 voitures) ont tendance à être entièrement automatisées.

I.7.1 Systèmes de stationnement semi-automatisés

Un système de stationnement semi-automatisé utilise un système mécanique pour déplacer les voitures vers leur place de stationnement, mais il a besoin d'une action humaine pour fonctionner, soit par le conducteur, soit par un préposé. Cette action peut être aussi simple que d'appuyer sur un bouton.

I.7.2 Systèmes de stationnement entièrement automatisés

Un système de stationnement entièrement automatisé ne nécessite aucun personnel, il est entièrement automatique. Du point de vue du conducteur, ils garent simplement leurs véhicules dans un module de stationnement et sont guidés vers la bonne position de stationnement par des capteurs via un panneau d'affichage. Les conducteurs éteignent leurs moteurs et la porte du module de stationnement est fermée pour sécuriser le module. Une fois le module sécurisé, le véhicule est retiré du module de stationnement et stocké. Lorsque les conducteurs reviennent et demandent leur véhicule, leurs véhicules sont retournés à un module de stationnement, généralement orienté dans la bonne direction, prêt à être conduit. Comme il n'y a pas d'exigence de rampes, d'allées et d'accès du personnel aux aires de stationnement, le stationnement automatisé peut généralement garer deux fois plus de véhicules dans le même volume que le stationnement conventionnel. **[5]**

I.8 Estimation d'évolution de l'utilisation des SPA

Une recherche estime que les revenus des SPA atteindront 356,5 millions de dollars par année d'ici 2020 dans le monde. Ainsi, il a beaucoup de potentiel et connaîtra une croissance énorme dans les années à venir. Avec les demandes de stationnement croissantes, les entreprises proposent des solutions innovantes et intelligentes pour répondre à ces exigences. **[4]**

I.9 Pourquoi louer une place de parking ?

- **Gain des temps :** Fini les pertes de temps à chercher une place de parking, vous êtes à proximité de l'endroit où vous voulez vous rendre. Vous êtes détendus et moins stressés.
- **Economie:** Fini l'angoisse de trouver son véhicule couvert de PV ou pire encore, à la fourrière Louez un parking privé est plus économique que d'utiliser des solutions de stationnement classiques.
- **Voiture en sécurité:** Votre voiture est mieux protégée dans un parking privé qu'à l'extérieur, et elle a moins de risques de se faire abimer (griffes, rétros cassés, vol..).
- **Un geste pour la planète:** Réduisez vos émissions de gaz à effets de serre et de polluants.
- **Une circulation décongestionnée. [6]**

I.10 Gestion des parkings

Les automobilistes ont besoin d'informations en continu concernant les disponibilités, les lieux et tarifs. Un guide d'accès de la ville fournira toutes ces informations. Ce genre de guide est en général développé par des agences publiques spécialisées.

Les informations sur ces parkings fournies aux automobilistes comprennent des cartes, signalisations, kiosque, brochures, GPS et bien d'autres systèmes de communication électroniques.

Ces outils peuvent améliorer la commodité, la sécurité, ainsi que proposer d'autres solutions de parking à moindre coût. Les zones industrielles ou résidentielles mettent en place des stratégies adaptées afin d'améliorer la circulation pendant les heures de pointe. **[6]**

I.11 Les avantages et les inconvénients de parking automatisés

I.11.1 Les avantages

- *Efficacité*

SPA nécessite environ 70 % d'espace plus petit et une taille moyenne de 50 % plus petite pour garer un nombre équivalent de voitures. Cela donne au promoteur de garage de stationnement beaucoup d'options, telles que l'ajout de plus de places de stationnement, l'utilisation de plus d'espace à des fins autres que le stationnement ou les espaces verts, etc.

- *Faisabilité et rentabilité*

Le stationnement et le développement de garages de stationnement ont souvent un effet direct sur la faisabilité d'un projet immobilier. La flexibilité de conception des SPA peut aider en termes de faisabilité et de rentabilité en permettant au stationnement d'être situé dans des zones où le stationnement conventionnel ne conviendra pas. Les emplacements possibles pour SPA sont pratiquement illimités : au-dessus du sol, sous terre, autoportants, sous ou intégrés dans des bâtiments existants et même des zones très étroites ou de forme irrégulière.

- *Respect de l'environnement et durabilité*

SPA réduit les émissions de CO2 de 85% ou plus car elles éliminent la nécessité pour les voitures de circuler au ralenti de la recherche des espaces de stationnement. De plus, les avantages environnementaux de SPA nécessitent généralement moins de matériaux de construction, un volume d'excavation beaucoup plus petit, des temps de construction plus courts et des perturbations de construction réduites.

- *Sécurité et sûreté*

Les véhicules parking 85% moins de temps que le stationnement conventionnel, car il n'y a pas de manœuvre ou de recherche de places de parking nécessaires. Les voitures et leurs contenus sont protégés contre les dommages, le vol et le vandalisme car personne ne conduit ou n'a accès à eux une fois qu'ils entrent dans le parking. Les clients gardent leurs clés et la voiture est verrouillée en tout temps. **[7]**

I.11.2 Les inconvénients

- *Le coût élevé de la construction ou de l'installation*

Le coût d'avoir un système de gestion du stationnement sain et fonctionnel est généralement élevé. Cela est dû aux différents composants qui entrent dans le fonctionnement du système. Des composants tels que la fonctionnalité statistique, la billetterie automatisée et les rapports statistiques, et bien d'autres, rendent tout cela coûteux. Certaines organisations peuvent ne pas être en mesure de se le permettre.

- *Entretien régulier*

Le système est automatisé ; cependant, il nécessite encore plusieurs entretiens réguliers de la part de l'entreprise. Il s'agit de s'assurer que le système fonctionne parfaitement et que rien ne s'est mal passé. L'entretien pourrait être une fois tous les mois.

- *Fonctionnement*

Beaucoup de gens ne sont pas habitués au système de gestion du stationnement. En conséquence, il peut être difficile pour eux de faire usage, provoquant ainsi d'autres complications pendant le stationnement.

- *La consommation d'énergie est élevée pour fonctionner tellement système.* **[8]**

I.12 Conclusion

Dans ce chapitre nous avons défini la notion de parking et détaillé les déférentes types , la classification, les avantages et les inconvénients , la nécessité de louer une place dans ce dernier, avec l'informatisation des systèmes de gestion de parking.

Pour réaliser une smart parking, on aura besoin de quelques spécifiques composants qui seront présentés dans le chapitre suivant.

CHAPITRE II

Description du matériel et logiciels utilisés

II.1 Introduction

Les composants électriques sont aujourd'hui une partie intégrante dans la vie quotidienne. En fait, le but de ces composants est de connecter à d'autres appareils pour réaliser une fonction électronique afin de rendre notre vie plus facile et plus pratique.

Dans ce chapitre nous allons parler les différents composants électroniques nécessaires de ce projet que nous proposons, Ensuite qui traitera la logicielle.

II.2 Partie matérielle

II.2.1 Les composants électroniques utilisés

Voici un tableau qui représenté les différents composants de ce projet avec la quantité :

Articles	quantité
Arduino Uno	**1**
Servomoteur	**2**
Capteur IR	**2**
Capteur LDR	**6**
Capteur DHT 11	**1**
Afficheur LCD 16*2 - I2C	**1**
RFID-RC522	**1**
Buzzer	**1**
Breadboard	**1**

Tableau II.1 : Les composants électroniques utilisés.

II.2.2 La carte Arduino

II.2.2.1 Les différentes cartes Arduino

Il existe plusieurs types de carte Arduino (Méga, UNO, Ethernet, Due, Leonardo, Esplora, Robot, Micro, YUN, NANO, MINI, Lily pad, Zéro, O Pro, 101 et etc…)



Arduino Méga



Arduino Uno



Arduino Ethernet



Arduino Due



Arduino Leonardo



Arduino Zero



Arduino Mo Pro



Arduino Esplora



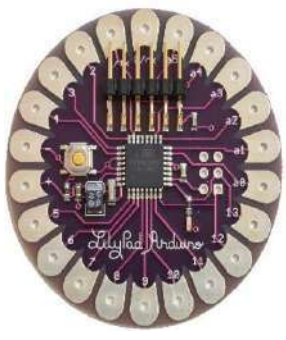
Arduino Nano



Arduino 101



Arduino Yun



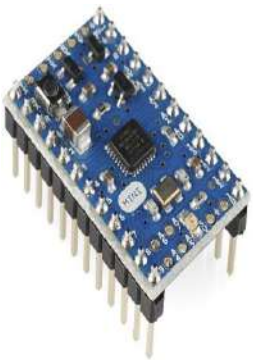
Arduino LilyPad



Arduino Robot



Arduino Micro



Arduino Mini

Figure II.1 : Les différentes cartes Arduino.

II.2.2.2 Aperçu sur Arduino Uno

L'Arduino Uno est une carte électronique programmable basée sur microcontrôleur ATmega 328, qui permettent de contrôler des éléments mécaniques : systèmes, lumières, moteurs, etc. Cette carte électronique permet donc à son utilisateur de programmer facilement des choses et de créer des mécanismes automatisés, sans avoir de connaissances particulières en programmation. **[9]**

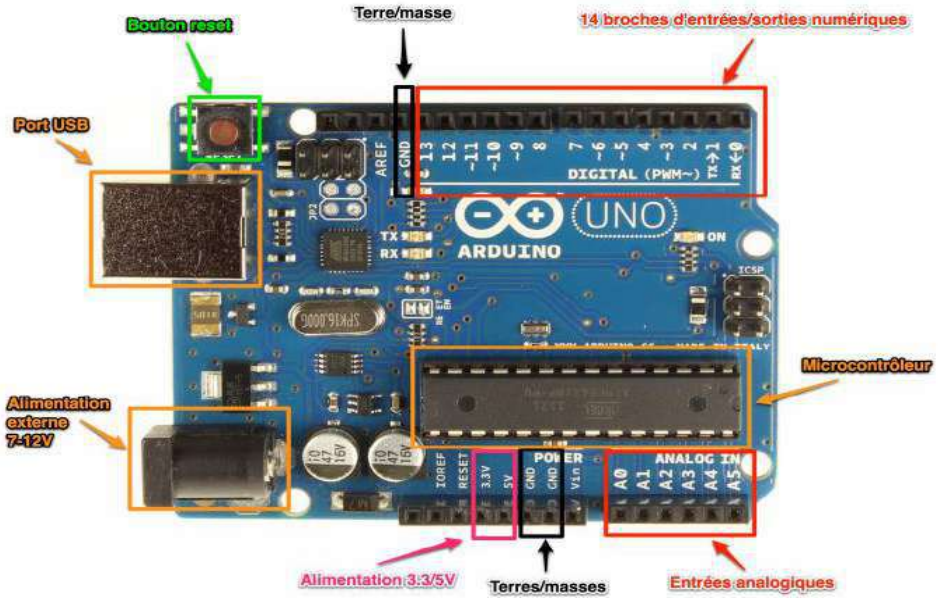


Figure II.2 : Une Carte Arduino Uno328.

II.2.2.3 Caractéristiques de l'Arduino Uno

La carte Arduino Uno est omniprésente dans de nombreux projets. Ceci est dû à ses caractéristiques intéressantes. **[9]**

Les caractéristiques de la carte Arduino Uno sont représentées dans le tableau II.2 (Annexe).

II.2.2.4 Configuration du brochage de l'Arduino Uno

Alimentation

- **Vin :** La tension d'entrée positive lorsque la carte Arduino est utilisée avec une source de tension externe (qui se distingue de la tension 5V connectée à l'USB ou à d'autres alimentations régulées 5V).
- **5V** : L'alimentation régulée utilisée pour alimenter les composants de la carte. Elle peut provenir soit du VIN, soit être fournie par USB.
- **3V3 :** Une tension régulée de 3.3 V générée par le régulateur embarqué. L'appel de courant maximal est de 50 mA.
- **GND :** Broche de masse (ou 0V). **[9]**

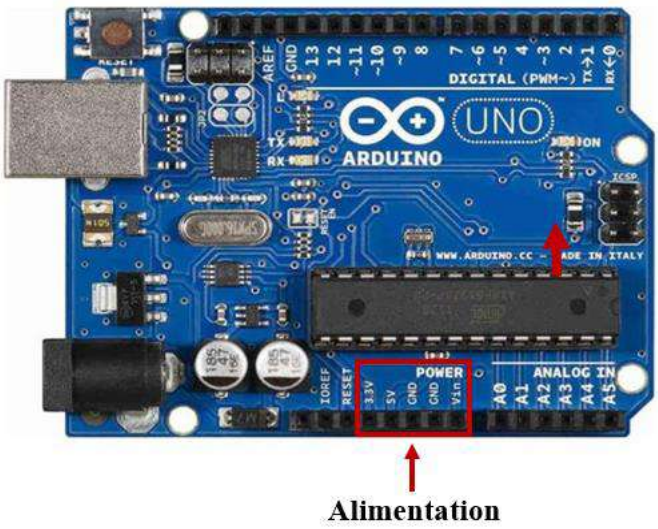


Figure II.3 : Alimentation de l'Arduino Uno328.

Broches d'Entrée et sortie numériques

- **Série :** (TX) et (RX). Utilisé pour transmettre (TX) et recevoir (RX) des données série TTL.
- **Interruptions externes :** Ces broches peuvent être configurées pour déclencher une interruption sur une valeur basse, un front montant ou descendant, ou un changement de valeur.
- **PWM :** 3,5,6,9,10 et 11 Offre une sortie PWM de 8 bits.
- **SPI :** (MISO), (MOSI), (SCK), (SS). Ces broches disponibles pour la communication SPI.
- **LED :** LED de tests connectés à la broche numérique 13.
- **I2C:** (SDA) et (SCL). **[9]**

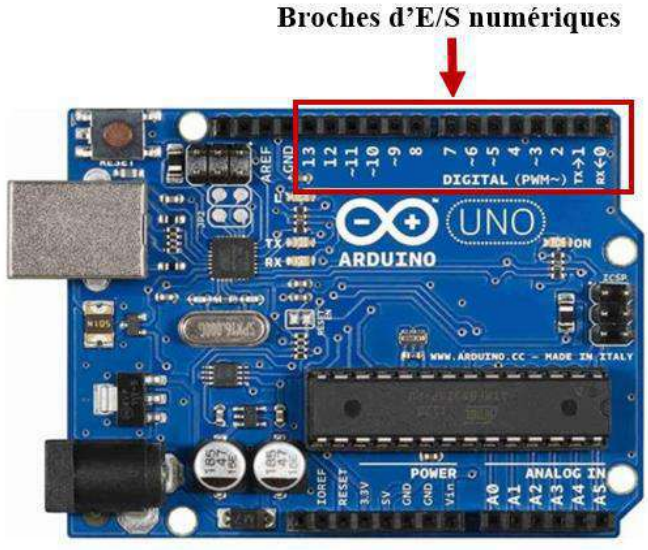


Figure II.4 : Broches d'E/S numériques de l'Arduino Uno328.

Broches d'Entrées et sortie analogiques

La carte dispose de 6 entrées et sortie analogiques pour cela, étiquetées A0 à A5. **[9]**

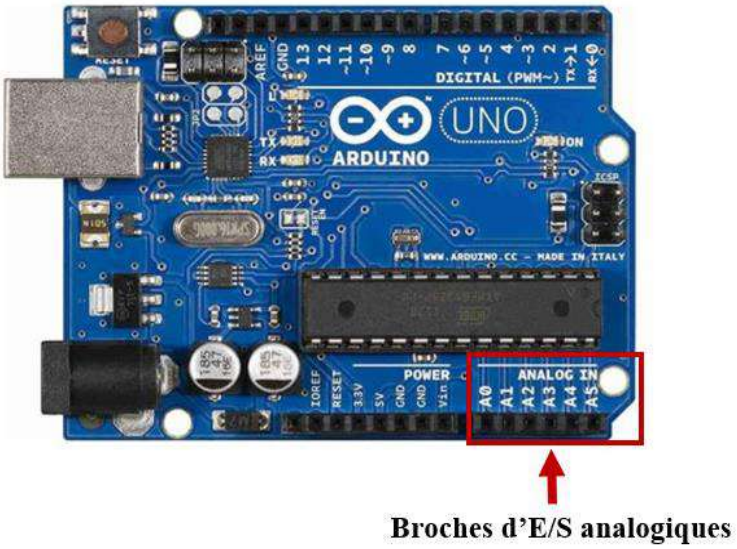


Figure II.5 : Broches d'E/S analogiques de l'Arduino Uno328.

Autres broches

- **AREF :** Tension de référence pour les entrées analogiques (si différente de 5V).
- **Reset :** En plaçant cette broche au niveau BAS, le microcontrôleur est réinitialisé (redémarré). **[9]**

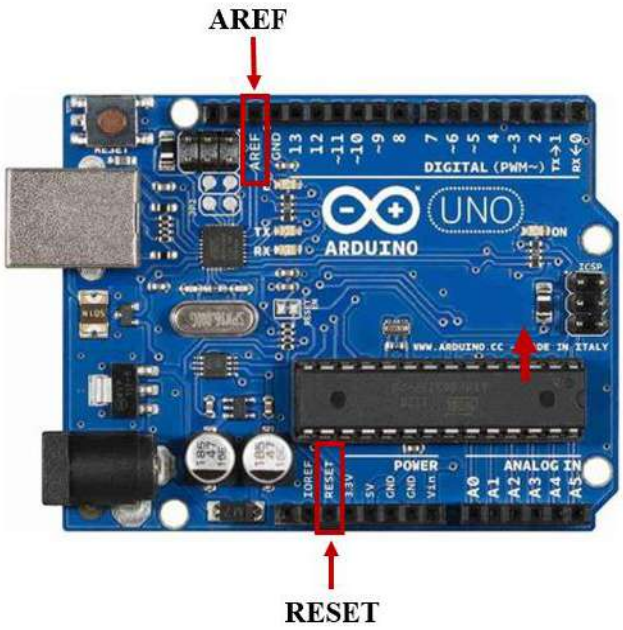


Figure II.6 : Broches RESET / AREF de l'Arduino Uno328.

II.2.3 Microcontrôleur ATmega328

Le microcontrôleur est une composante très importante dans la carte Arduino. La puce qui équipe la carte Arduino Uno est une « microcontrôleur ATmega328 ». C'est un microcontrôleur d'une famille AVR® RISC de Micro chip 8 bits, dont la programmation peut êtreréalisée en langage C/C++.

Le microcontrôleur ATmega328 embarque un processeur central pour l'exécution du programme stocker dans une mémoire Flash ISP de 32 Ko, une SRAM de 2 Ko, une EEPROM de 1 Ko pour enregistrer les informations et conserver les données. Ainsi qu'il possède des 23 lignes d'E/S générales, 32 registres de travail généraux, un Le dispositif fonctionne entre 1,8 et 5,5 volts. Ces broches sont utilisées pour connecter le microcontrôleur avec différents composantsélectroniques. **[10]**



Figure II.7 : Microcontrôleur ATmega328.

II.2.4 Servomoteur

Un servomoteur (vient du latin servus qui signifie « esclave »)

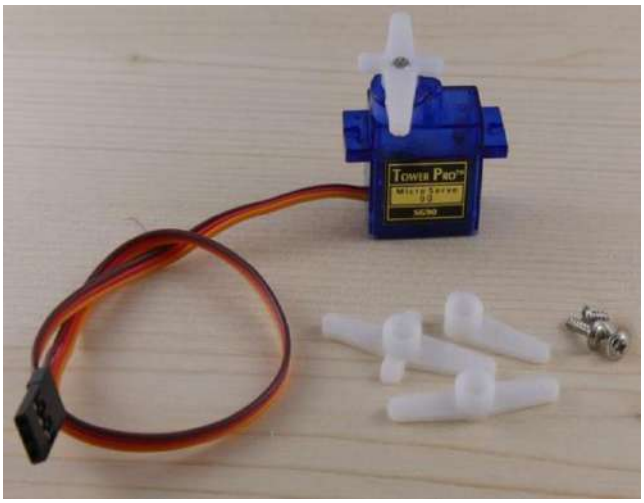


Figure II.8 : Servomoteur "9 grammes".

II.2.4.1 Description de servomoteur

Le servomoteur est un système relativement complexe qui intègre des composants électroniques et électromécaniques.

Ce dispositif est composé d'un moteur à courant continu à deux fils, d'un train d'engrenages, d'un potentiomètre, d'un circuit intégré et d'un arbre de sortie. De plus, ils sont connectés trois fils à ce dispositif : un fil de terre (0V), un fil de tension d'entrée (généralement 4,5V à 6V) et un fil de commande.

À l'aide du fil de commande envoie un signal à ce servomoteur, qui ordonne à l'arbre de tournerdans une position angulaire spécifique (0°-180°). **[11]**



Figure II.9 : Structure de Servomoteur "9 grammes".

II.2.4.2 Programmation de servomoteur

Afin de faire bouger le servomoteur, nous devrons utiliser une bibliothèque de codes appelée "Servo". Il s'agit de la configuration standard de l'environnement de développement Arduino (IDE).

La bibliothèque de servomoteur peut contrôler jusqu'à 12 servomoteurs simultanément avec une carte Arduino UNO, Il suffit d'importer la bibliothèque en ajoutant cette ligne en début de programme pour l'utiliser : « #include <Servo.h> ». En effet, afin de créer un objet Servo, il faut simplement déclarer une variable (globale) de type" Servo ", exemple : " monServo ", Si vous souhaitez utiliser plusieurs servomoteurs en même temps, vous devez créer plusieurs variables, une pour chaque servomoteur. **[12]**

II.2.4.3 Schéma de câblage avec Arduino

Le câblage est relativement simple, le fil rouge (ou orange en fonction des constructeurs) du servomoteur doit être connecté sur la broche 5V de la carte Arduino, le fil noir (ou marron en fonction des constructeurs) sur la broche GND et le fil jaune (ou blanc en fonction des constructeurs) sur une broche numérique de la carte Arduino. la figure II.10 illustre le schéma de câblage de servomoteur avec la carte Arduino Uno.

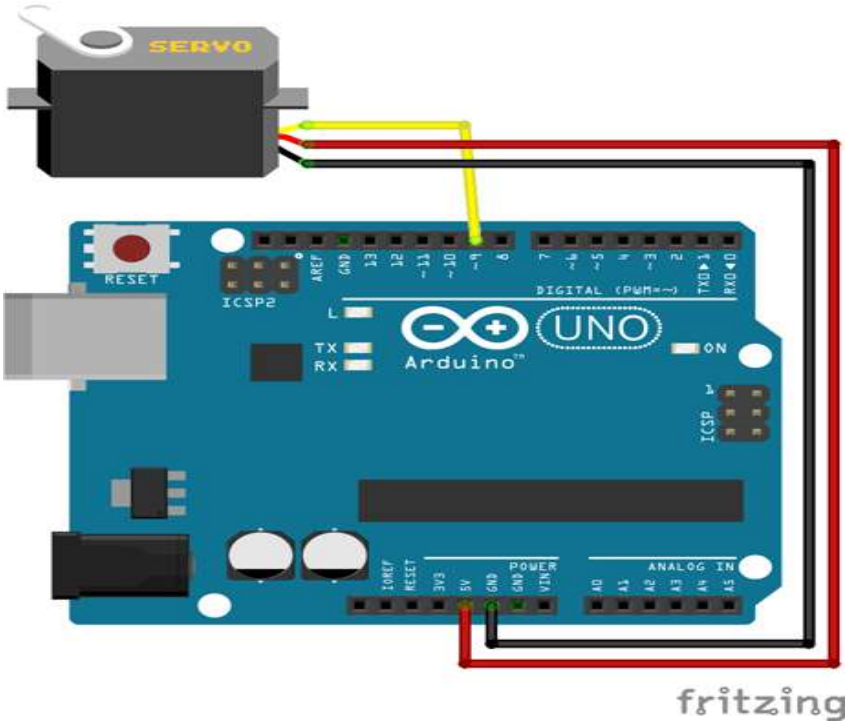


Figure II.10 : Connexion servomoteur avec la carte Arduino Uno.

II.2.5 Capteur IR (FC-51)

Le détecteur infrarouge est un dispositif capable de repérer des mouvements ou une présence physique sur l'ensemble du périmètre couvert par son capteur.



Figure II.11 : Capteur IR (FC-51).

II.2.5.1 Description de capteur IR

Le capteur IR peut être alimenté par une tension entre 3 et 5 V DC, la présence de tension est indiquée par sa LED rouge.

Ce module utilise une paire de diodes infrarouges (un émetteur et un récepteur) qui fonctionnent par réflexion. Il peut détecter un objet à une distance comprise entre 2 et 30 cm avec un angle de détection de 35°, cette distance peut être réglée par la résistance variable.

Ainsi la couleur influe également l'opération de détection. En effet, la distance de détection seraminimale si l'objet est noir, et maximale s'il est blanc.

En fin, La stabilité du signal est garantie par l'utilisation d'un comparateur LM393. **[13]**

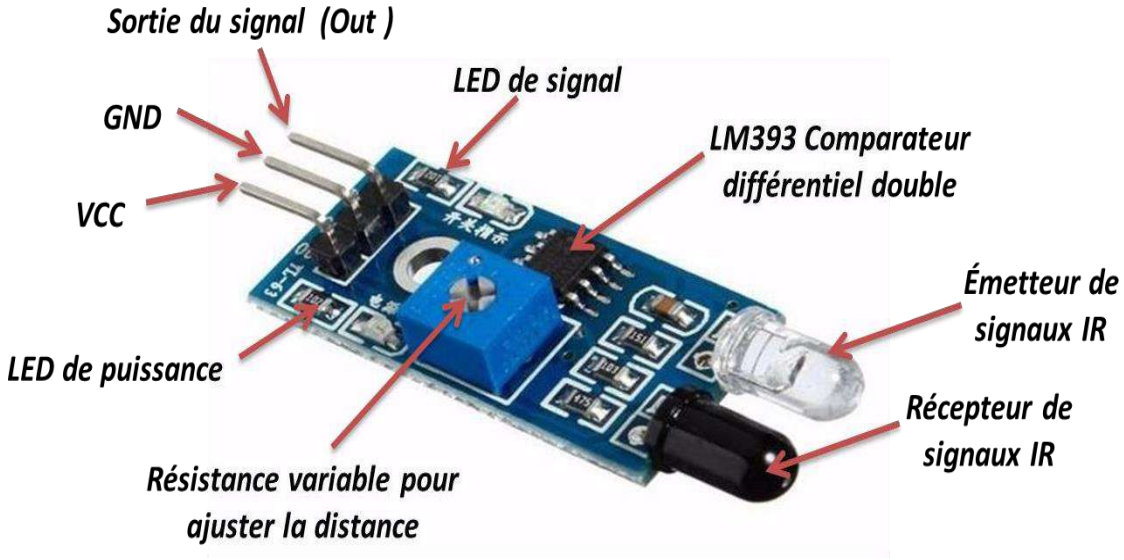


Figure II.12 : Structure d'un Capteur IR.

Les broches de ce capteur sont présentées dans le tableau suivant :

Sortie et entrée du PIN	
Entrée du PIN	**VCC :** Entrée d'alimentation 3.3V-5V DC. **GND :** 0V Broche d'alimentation.
Sortie du PIN	**OUT :** Broche de sortie numérique.

Tableau II.3 : Sortie et entrée d'un Capteur IR.

II.2.5.2 Principe de fonctionnement

Lorsque Le capteur infrarouge envoie un rayonnement qui atteint le corps, il est réfléchi vers le récepteur infrarouge.

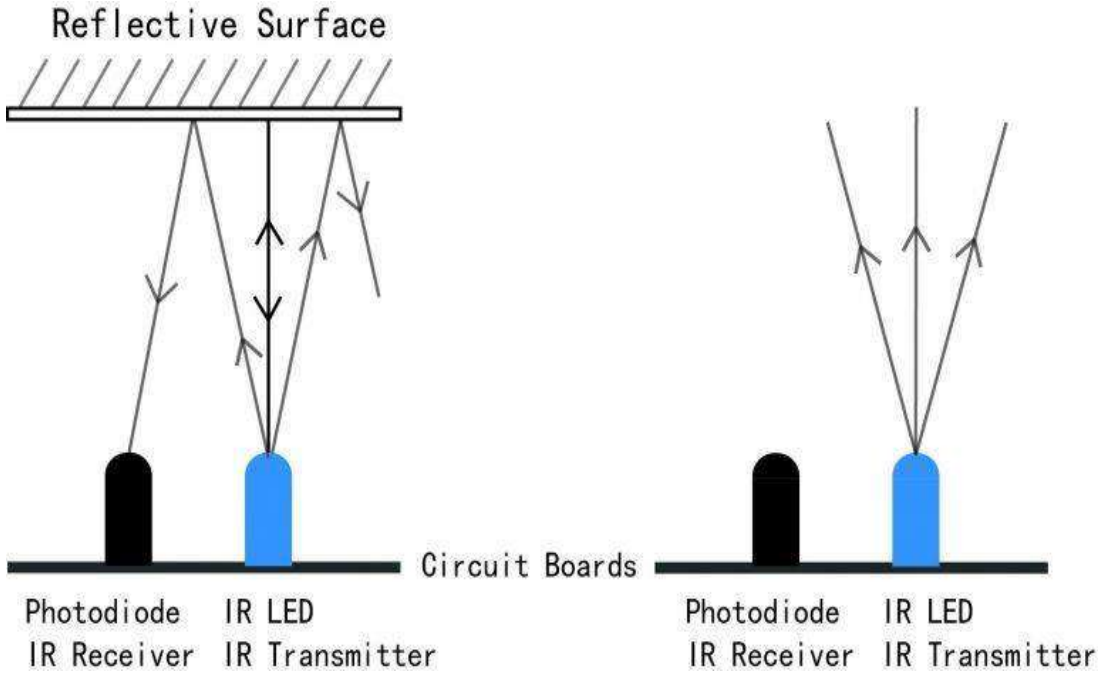


Figure II.13 : Principe de fonctionnement d'un capteur IR.

a. Émetteur IR

Un émetteur infrarouge est une diode électroluminescente (LED) qui émet des radiations infrarouges. C'est pourquoi on les appelle des LED IR. Même si une LED IR ressemble à une LEDnormale, le rayonnement qu'elle émet est invisible pour l'œil humain. **[14]**



Figure II.14 : Émetteur IR.

b. Récepteur IR

Les récepteurs infrarouges sont également appelés capteurs infrarouges car ils détectent le rayonnement d'un émetteur IR. Les récepteurs IR se présentent sous la forme de photodiodes et de phototransistors. Les photodiodes infrarouges sont différentes des photodiodes normales

car elles ne détectent que le rayonnement infrarouge. L'image d'un récepteur IR typique ou d'une photodiode est présentée dans la figure. **[14]**



Figure II.15 : Récepteur IR.

II.2.5.3 Schéma bloc d'un capteur IR

Lorsque l'émetteur IR émet un rayonnement, ce dernier atteint l'objet et une partie du rayonnement est réfléchie vers le récepteur IR. Par conséquent, Le comparateur LM393 détecte la faible valeur de résistance de la photodiode et envoie un signal logique haut (5V) à la sortie de module (out).

Dans le cas contraire, Le comparateur LM393 détecte la valeur de résistance haute de la photodiode et envoie un signal logique bas (0V) à la sortie du capteur (out). **[15]**

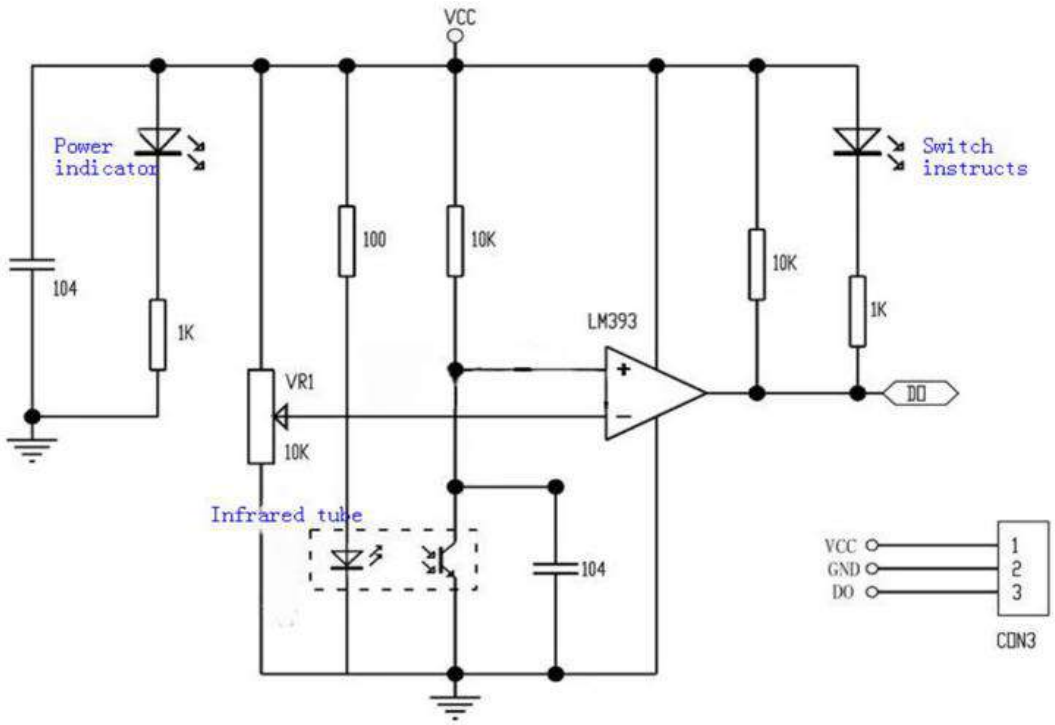


Figure II.16 : Schéma bloc d'un capteur IR.

II.2.5.4 Schéma de câblage avec arduino

Comme on peut le constater sur la figure II.17, nous n'avons besoin que de 3 connexions au capteur. Les connexions sont les suivantes : tension (rouge), masse (noire) et signal (jaune) pouvant être connecté à n'importe quelle broche analogique de notre carte Arduino UNO. La figure illustre le schéma de câblage du capteur IR (FC-51) avec la carte Arduino.

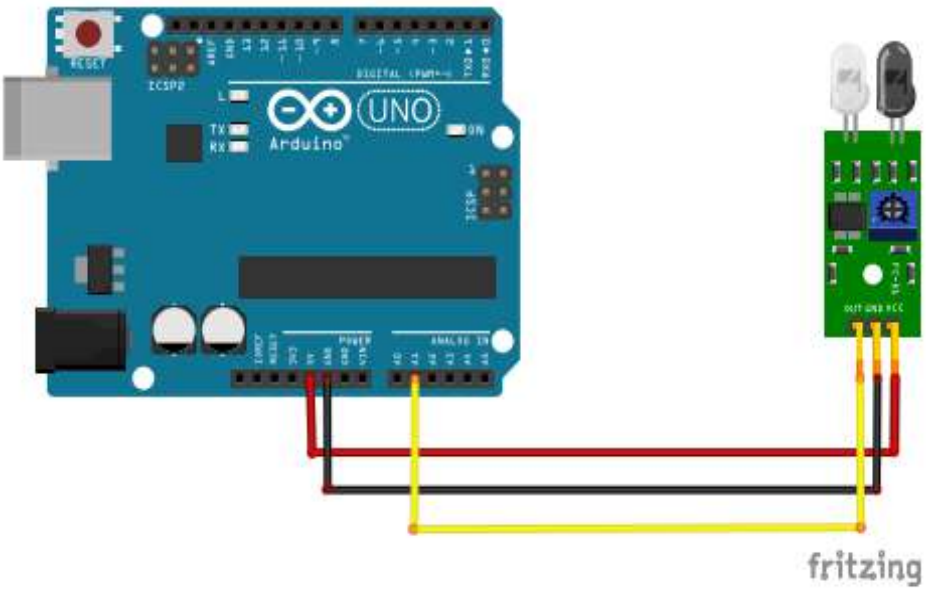


Figure II.17 : Connexion de capteur IR avec la carte Arduino Uno.

II.2.6 Capteur LDR

Les photorésistances (LDR) sont des résistances variables qui sont contrôlées par la lumière. Également appelées résistances photo dépendantes ou cellules photoconductrices, elles sont généralement utilisées dans des circuits de commutation activés par la lumière ou l'obscurité, ou des circuits de détection sensibles à la lumière.



Figure II.18 : Capteur LDR.

II.2.6.1 Principe de fonctionnement

La résistance dépendante de la lumière (LDR) fonctionne sur le principe de **la « photoconductivité ».** La résistance LDR est modifiée en fonction de l'intensité lumineuse qui tombe sur le LDR. Lorsque l'intensité lumineuse augmente sur la surface LDR, la résistance LDR diminue et la conductivité de l'élément augmente. Lorsque l'intensité lumineuse diminue sur la surface LDR, la résistance LDR augmente et la conductivité de l'élément diminue. **[16]**

II.2.6.2 Caractéristiques d'un capteur LDR

Les Caractéristiques d'un capteur LDR sont représentées dans le tableau II.4 :

Spécifications technique	
Tension de fonctionnement	5V ou 3.3V DC
Puce de comparaison	LM393
Type de sortie	Sorties numériques (D0)
Sensibilité	Réglable

Tableau II.4 : Les caractéristiques de capteur LDR.

II.2.6.3 Les différents types de capteur LDR

Les résistances dépendantes de la lumière sont classées en fonction des matériaux utilisés :

a- Résistances photo intrinsèques

Ces résistances sont des dispositifs semi-conducteurs purs comme le silicium ou le germanium. Lorsque la lumière tombe sur le LDR, les électrons sont excités de la bande de valence vers la bande de conduction et le nombre de porteurs de charge augmente.

b- Résistances photo extrinsèques

Ces dispositifs sont dopés avec des impuretés et ces impuretés créent une nouvelle bande d'énergie au-dessus de la bande de valence. Ces bandes sont remplies d'électrons. Par conséquent, cela diminue la bande interdite et une petite quantité d'énergie est nécessaire pour les déplacer. Ces résistances sont principalement utilisées pour les longues longueurs d'onde. **[17]**

II.2.6.4 Schéma de câblage avec Arduino

La photorésistance doit être connectée sur une entrée analogique afin de mesurer la variation de tension en fonction de la lumière captée par celle-ci. Pour exploiter le capteur photorésistance, il suffit d'alimenter les deux pattes comme suit : Une à la broche 5V de la carte Arduino et l'autre à l'entrée analogique de la carte par l'intermédiaire d'une résistance de 10KOm comme le montre le schéma de la figure . **[18]**

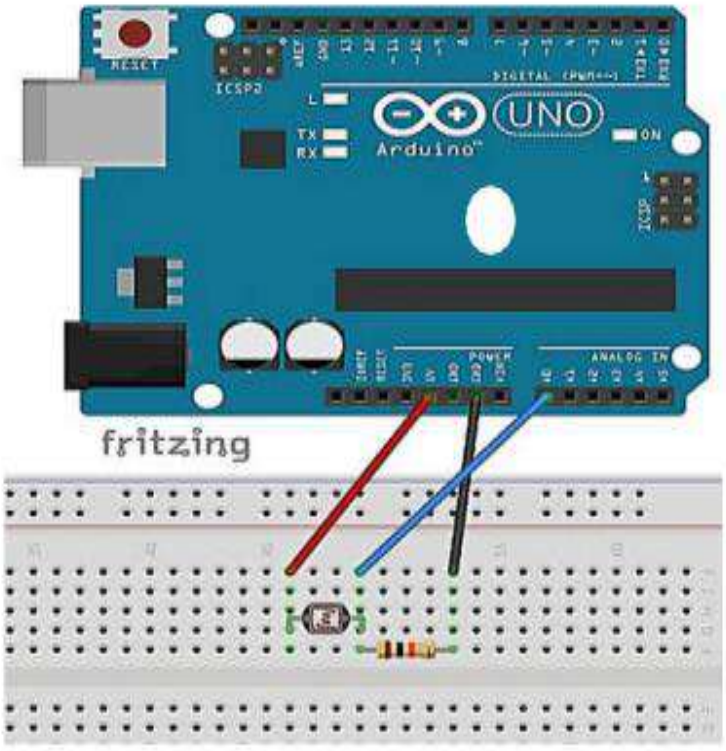


Figure II.19 : Connexion LDR avec la carte Arduino Uno.

II.2.7 Capteur DHT 11

DHT11 est un capteur numérique pour détecter la température et l'humidité.

II.2.7.1 Description sur DHT 11

Le capteur de température et humidité DHT 11 est un module multifonction, il nous renseigne à la fois sur la température et sur l'humidité relative sur une large gamme de mesure.

Il fournit une lecture fiable lorsque le degré d'hygrométrie (humidité) ambiante est compris entre 20 % RH et 90 % RH, et lorsque la température est comprise entre 0°C et 50°C, ce qui couvre les besoins de la plupart des foyers et des activités quotidiennes ne présupposant pas de conditions extrêmes. **[19]**

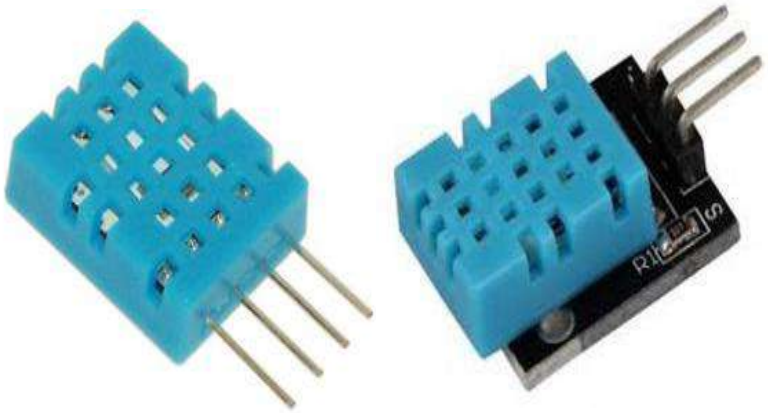


Figure II.20 : Capteur DHT 11.

II.2.7.2 Principe de fonctionnement

Le capteur DHT11 se compose d'un élément capacitif de détection d'humidité et d'une thermistance pour détecter la température. Le condensateur de détection d'humidité a deux électrodes avec un substrat de rétention d'humidité comme diélectrique entre elles.

La modification de la valeur de capacité se produit avec le changement des niveaux d'humidité. La mesure IC, traiter ces valeurs de résistance modifiées et les changer sous forme numérique.

Pour mesurer la température, ce capteur utilise une thermistance à coefficient de température négative, ce qui entraîne une diminution de sa valeur de résistance avec une augmentation de la température. Pour obtenir une plus grande valeur de résistance, même pour le plus petit changement de température, ce capteur est généralement composé de céramiques semi-conductrices ou de polymères. **[19]**

II.2.7.3 Caractéristiques techniques

- Alimentation : 3.3v à 5v.
- Plage de mesure : -température : de 0 à 50°C. Humidité : de 20% à 90% RH.
- Précision : -température : 2°C. Humidité : 5% RH.
- Sensibilité : -température : 1°C. Humidité : 1% RH.
- Période de collecte du signal : 2 Secondes. **[18]**

II.2.7.4 Schéma de câblage avec Arduino

Comme on peut le constater sur la figure II.21 , nous n'avons besoin que de 3 connexions au capteur. Les connexions sont les suivantes : tension (rouge), masse (noire) et signal (jaune) pouvant être connecté à n'importe quelle broche analogique de notre carte Arduino Uno.
La figure II.21 illustre le schéma de câblage du capteur DHT11 avec la carte Arduino. **[18]**

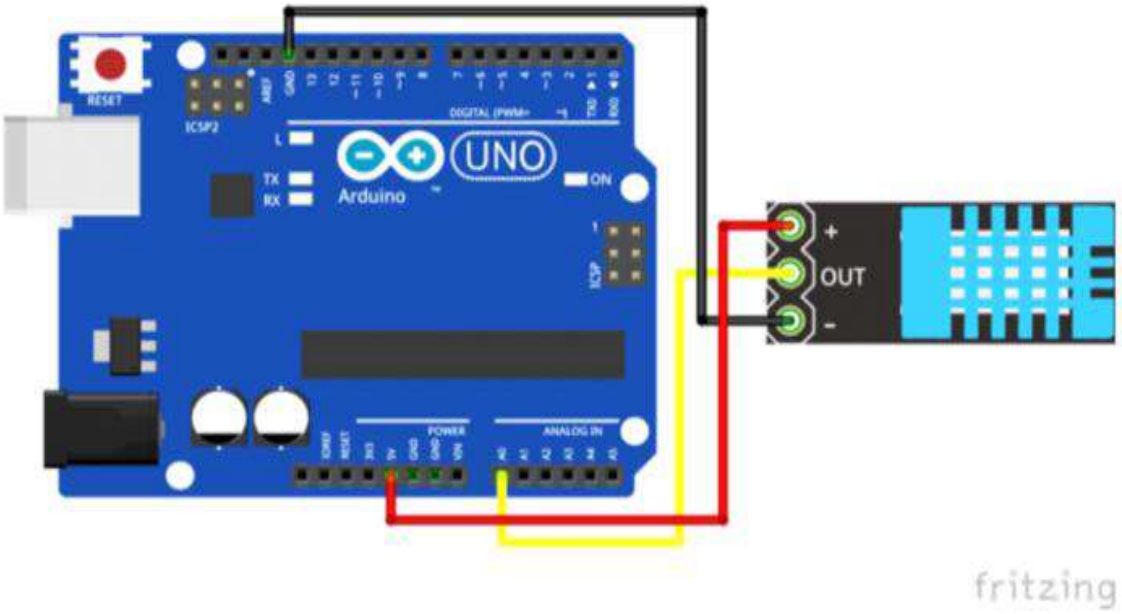


Figure II.21: Connexion DHT11 avec la carte Arduino Uno.

II.2.7.5 Applications

Ce capteur est utilisé dans diverses applications telles que la mesure de l'humidité et des valeurs de température dans les systèmes de chauffage, de ventilation et de climatisation. Les stations météorologiques utilisent également ces capteurs pour prédire les conditions météorologiques. Le capteur d'humidité est utilisé à titre préventif dans les maisons où les gens sont affectés par l'humidité. Les bureaux, les voitures, les musées, les serres et les industries utilisent ce capteur pour mesurer les valeurs d'humidité et comme mesure de sécurité. Sa taille compacte et son taux d'échantillonnage ont rendu ce capteur populaire parmi les amateurs. **[19]**

II.2.8 Écran LCD I2C 16X2 avec rétro-éclairage bleu

Il s'agit d'un écran LCD 16X2 avec rétro éclairage bleu équipé d'un convertisseur I2C qui permetla connexion à Arduino en utilisant seulement deux broches par connexion, plus les deux câbles d'alimentation. **[20]**

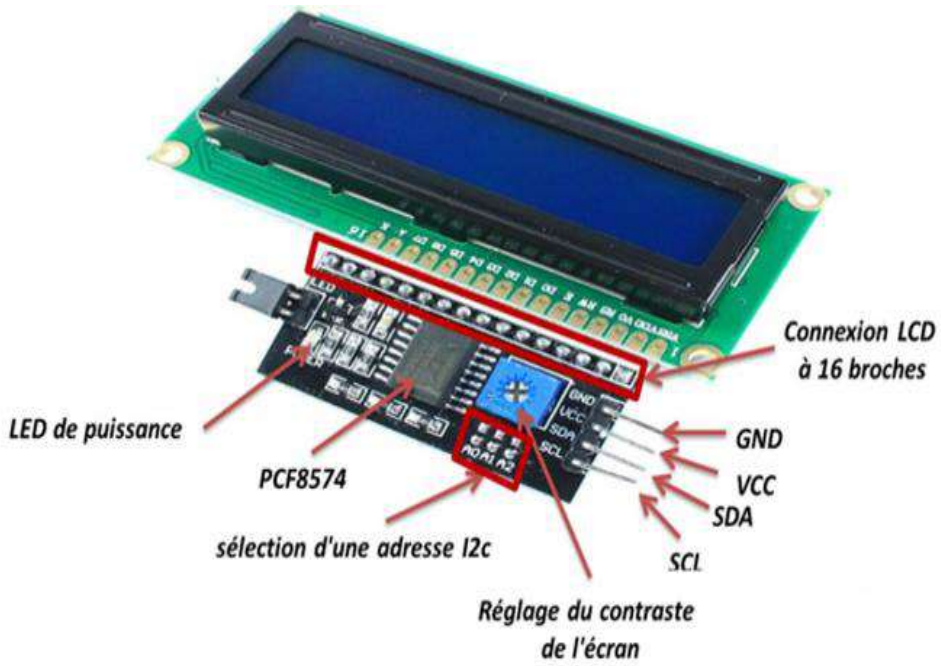


Figure II.22 : Structure d'un écran LCD-I2C (16*2).

II.2.8.1 Présentation du matériel de l'écran LCD I2C

Un écran LCD I2C typique se compose d'un écran LCD à caractères basé sur le HD44780 et d'un adaptateur LCD I2C.

a- Écran LCD

Ces écrans LCD sont idéaux pour l'affichage de texte/caractères uniquement. Par exemple, un LCD 16×2 caractères, doté d'un rétro éclairage par LED et peut afficher 32, avec 16 caractères sur chaque rangée.

b- Adaptateur I2C pour LCD

Le cœur de l'adaptateur est une puce d'extension d'E / S 8 bits-PCF8574. Cette puce convertit les données I2C de l'Arduino en données parallèles requises par l'écran LCD.

II.2.8.2 Adresse I2C du LCD

Si vous utilisez plusieurs périphériques sur le même bus I2C, vous devrez peut-être définir une adresse I2C différente pour la carte, afin qu'elle n'entre pas conflit avec autre périphérique I2C.

Pour ce faire, Il y a trois connexions nommées A0, A1 et A2 pour définir (3 bits) l'adresse I2C entre 0x20 et 0x27. L'adresse par défaut est 0x27 (A0, A1 et A2 laissés ouverts). Une paire est fermée en connectant les deux plots respectifs ensemble. **[20]**

II.2.8.3 Schéma de câblage avec Arduino

La connexion d'un LCD I2C est beaucoup plus facile de connecter un écran LCD I2C que de connecter un écran LCD standard. Il vous suffit de connecter 4 broches au lieu de 12. Commencez par connecter la broche VIN à la sortie 5V de l'Arduino et connectez GND à la terre.

Maintenant, nous restons avec les broches qui sont utilisées pour la communication I2C. Notez que chaque carte Arduino a des broches I2C différentes qui doivent être connectées en conséquence. Sur les cartes Arduino avec la disposition R3, le SDA (ligne de données) et le SCL (ligne d'horloge) sont sur les en-têtes de broche près de la broche AREF. Ils sont également connus sous le nom de A5 (SCL) et A4 (SDA).

La figure II.23 illustre le schéma de câblage d'un l'écran LCD I2C avec la carte Arduino. **[20]**

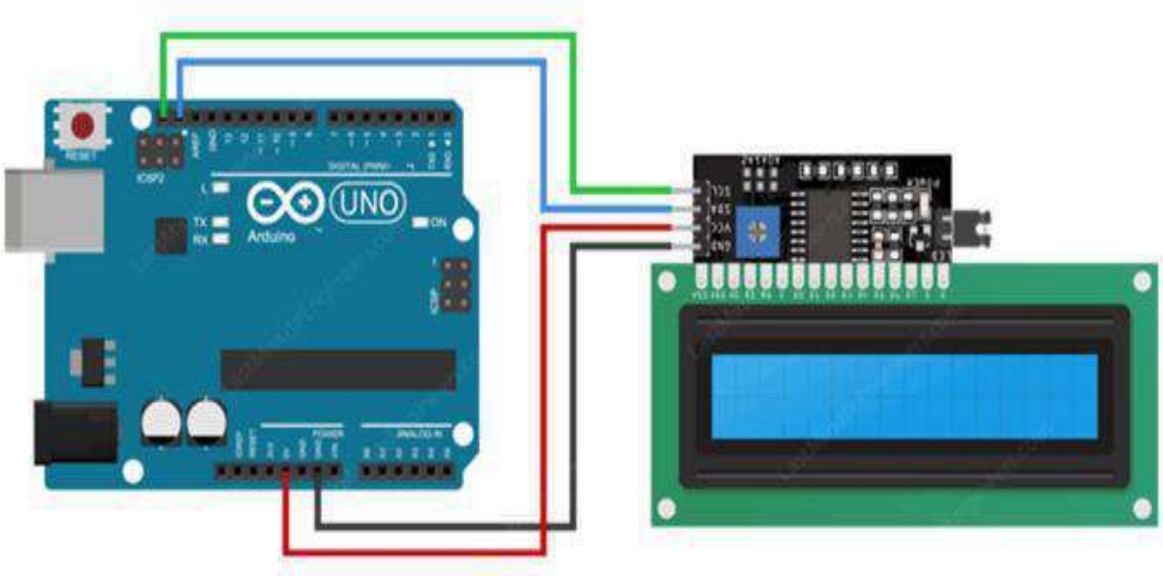


Figure II.23 : Connexion un écran LCD I2C avec la carte Arduino Uno.

II.2.9 Système RFID

Le terme RFID « Radio-Fréquence Identification » englobe toutes les technologies qui utilisentles ondes radio pour identifier automatiquement des objets ou des personnes. Il s'agit d'une technologie qui permet de stocker et de récupérer des informations à distance grâce à une étiquette qui émet des fréquences radio.

Le système RFID ou système d'identification par radiofréquence se compose de deux éléments principaux :

- L'étiquette RFID (carte ou tag) est elle-même équipée d'une puce reliée à une antenne.
- Lecteur RFID équipée d'une antenne intégrée ou externe. **[21]**

II.2.9.1 Description de la carte RFID RC522

Le module RFID RC522 basé sur le circuit intégré MFRC522. Ce module est conçu pour créer un champ électromagnétique de 13,56 MHz pour communiquer avec les étiquettes RFID. Ainsi que, Le lecteur peut communiquer avec un microcontrôleur via une interface périphérique série (SPI) à 4 broches avec un débit maximal de 10 Mbps. Il permet aussi de communiquer via les protocoles I2C et UART.

La tension de fonctionnement du module est de 2,5-3,3V. De plus, les broches logiques sont tolérantes à 5V, donc nous pouvons facilement le connecter à un Arduino ou tout microcontrôleur avec une logique 5V sans utiliser un converti sseur de niveau logique. **[22]**

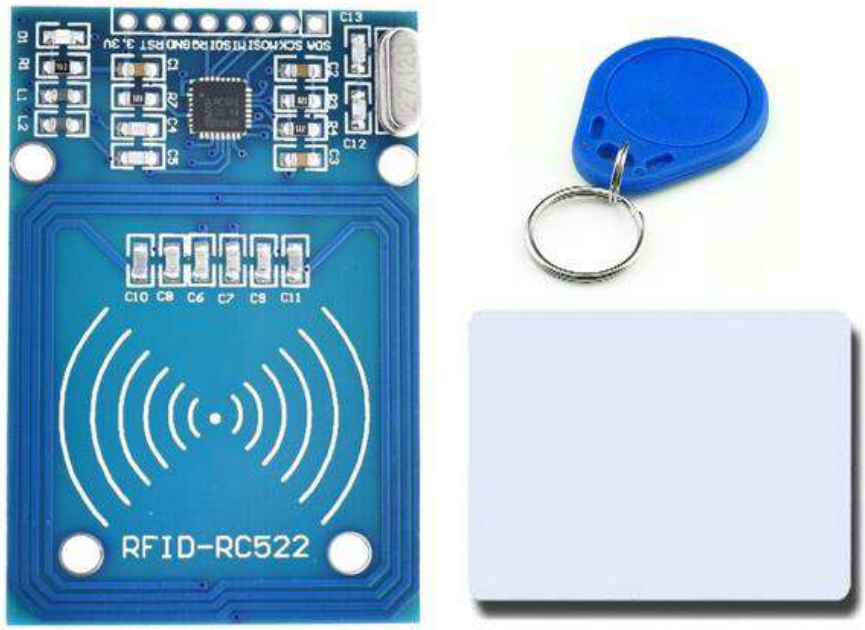


Figure II.24 : La carte RFID RC522 et les tags.

Le tableau suivant indique les spécifications techniques du lecteur RFID :

Spécifications techniques	
Gamme de fréquences	Bande ISM 13,56 MHz
Interface hôte	SPI / I2C / UART
Tension d'alimentation	2,5 V à 3,3
Max. Courant de fonctionnement	V13-26Ma
Min. Courant (hors tension)	10µA
Entrées logiques	Tolérance de 5V

Tableau II.5 : Les spécifications de la carte RFID RC522.

II.2.9.2 Principe de fonctionnement d'un système RFID

L'étiquette RFID (carte ou tag) est elle-même équipée d'une puce reliée à une antenne, l'antenne permet à la puce de transmettre les informations (numéro de série) qui sont lues parun lecteur.

Une fois les informations transmises au lecteur RFID équipée d'une antenne intégrée ou externe, celui-ci n'a plus qu'à convertir les ondes-radios en données et ceux-ci pourront être lus par un logiciel RFID.

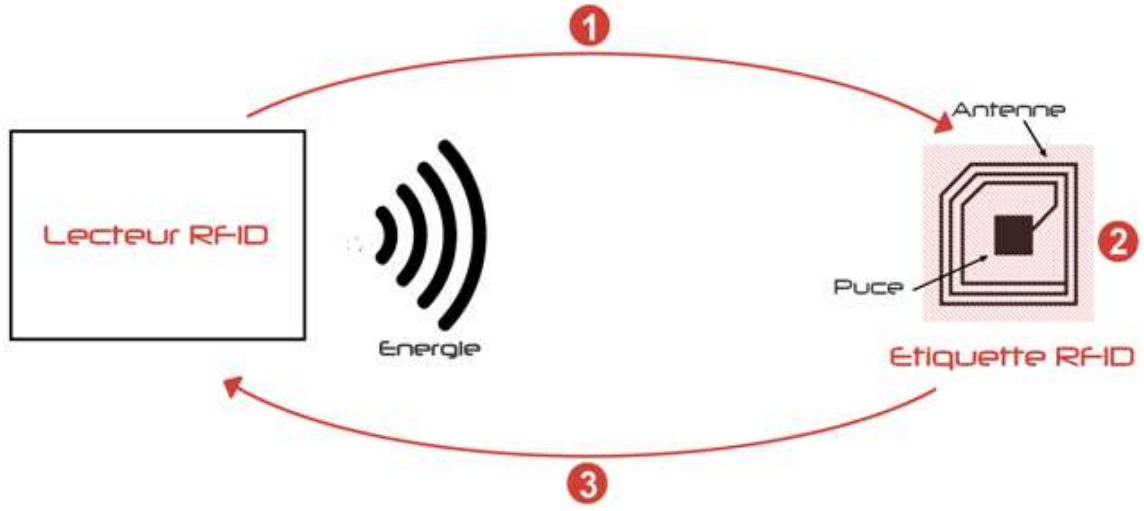


Figure II.25 : Schéma de principe de système RFID.

II.2.9.3 Schéma de câblage avec Arduino

Le câblage est relativement simple, Pour commencer, connectez la broche VCC du module à 3,3 V sur l'Arduino et la broche GND à la terre. La broche RST peut être connectée à n'importe quelle broche numérique sur l'Arduino. Dans notre cas, il est connecté à la broche numérique # 5. La broche IRQ n'est pas connectée car la bibliothèque Arduino que nous allons utiliser ne la prend pas en charge.

Maintenant, nous restons avec les broches qui sont utilisées pour la communication SPI. Comme le module RC522 nécessite beaucoup de transfert de données, ils donneront les meilleures performances lorsqu'ils sont connectés aux broches SPI matérielles d'un microcontrôleur. Les broches SPI matérielles sont beaucoup plus rapides que le « bit-banging » du code de l'interface à l'aide d'un autre ensemble de broches.

Le broche SPI qui doivent être connectées en la carte arduino Uno est numérique 13 (SCK), 12 (MISO), 11 (MOSI) et 10 (SS). **[22]**

La figure II.26 illustre le schéma de câblage du module RFID RC522 avec la carte Arduino.

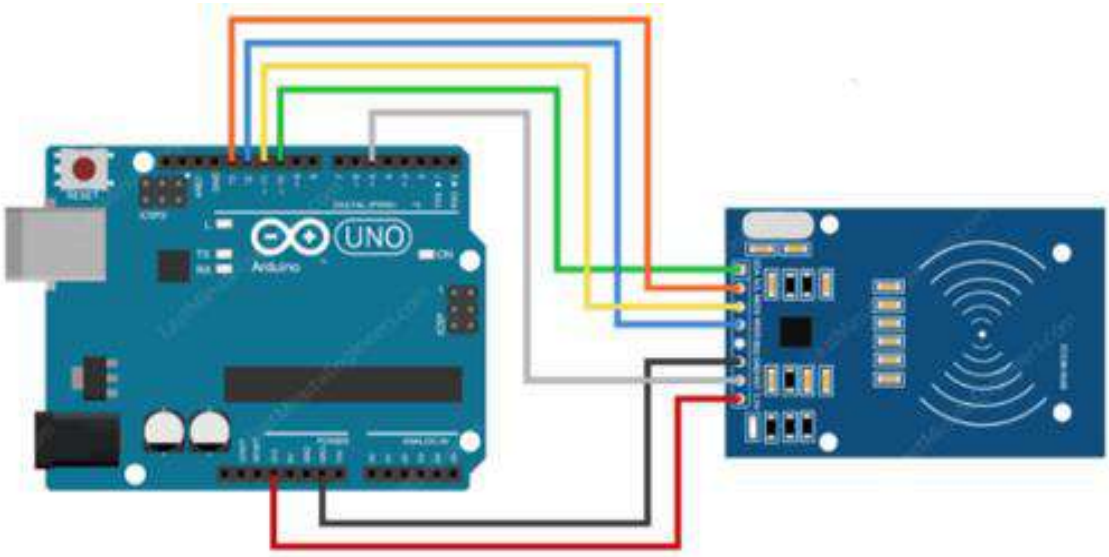


Figure II.26 : Connexion du module RFID RC522 avec la carte Arduino Uno.

II.2.9.4 Les Applications

Les applications de la RFID peuvent être utilisées dans les entreprises, par les individus ainsi que par les états. Ce système est exploité dans plusieurs domaines : le transport, la sécurité, la bibliothèque, et la logistique qui sont autant des domaines dans lesquels cette technologie existe déjà et apparaîtra dans le futur.

Le paiement : le paiement est un des défis importants de la RFID. Parmi les applications de la RFID dans ce domaine, le modèle Pidion BIP-1300 se présente comme un PDA durci, qui accepte des techniques de paiement diverses, comme les cartes bancaires à puce ou à bande.

Transport : Une des applications les plus connues et les plus démocratisées de la technologie RFID reste la carte de transport sans contact. L'usager du métro passe sa carte sur une base (généralement apposée à des tourniquets d'accès), qui l'authentifie, valide son titre de transport, et lui donne accès au réseau. 3,4 millions de titres de transport sans contact circulaient en juin 2008 sur le réseau de transport en commun parisien de la RATP.

Bibliothèques : Certaines bibliothèques ont mis en place un système d'IRF pour faciliter l'emprunt de livres, contrôler les stocks et réduire les traumatismes répétés chez les bibliothécaires. Cependant, les préoccupations au sujet de la surveillance des livres choisis ont mis au jour les problèmes de protection de la vie privée relatifs à l'IRF.

Contrôle d'accès : L'identification des individus passe aussi par l'authentification des papiers d'identités. Le RFID est alors un moyen qui d'une part, s'assure de la validité des documents, mais aussi, il s'assurer que les informations contenues dans le passeport le sont également d'un point de vue numérique. Il est à noter que des questions de sécurité importantes se posent quant à la l'intégrité des données contenues dans les tags RFID des passeports. **[23]**

II.2.10 Buzzer

Un Buzzer est un composant électronique qui produit un son lorsqu'une tension lui e st appliquée.



Figure II.27 : Buzzer.

II.2.10.1 Principe de fonctionnement

Le buzzer possède deux petites pattes de fixation. La puissance sonore de ce composant est d'environ 85 dB / cm et il nécessite une tension continue pour fonctionner. La tension continue doit généralement être comprise entre 3 V et 28 V. Un buzzer conçu pour 6 V peut généralement bien fonctionner à n'importe quelle tension d'alimentation de 4 V à 8 V, tandis qu'un buzzer conçu pour 12 V peut fonctionner parfaitement à une tension de 6 V à 28 V (veuillez-vous référer aux spécifications données fabricant pour éviter les erreurs).

Les buzzers électriques sont utilisés pour émettre des bips. **[24]**

II.2.10.2 Schéma de câblage avec Arduino

- Le VCC vers une broche digitale de la carte (pin 9 dans la figure).
- La GND vers la GND de la carte.

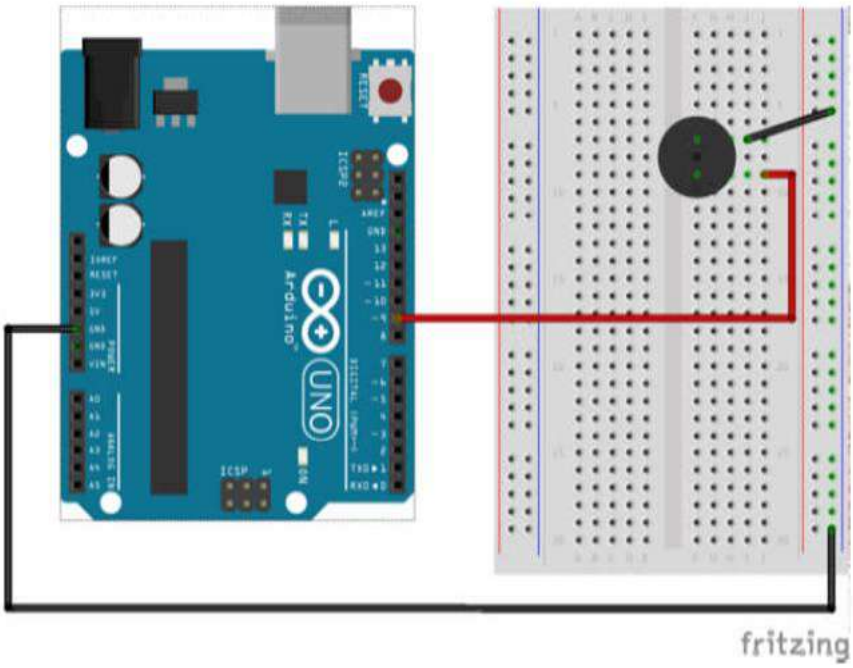


Figure II.28 : Connexion buzzer avec Arduino Uno.

II.2.11 Breadboard

La planche de montage expérimental (appelé en anglais breadboard et platine Labdec en Français) est un outil essentiel pour l'électronique. Ce dispositif permet de réaliser le prototype d'un circuit électronique et de le tester.

L'avantage de ce système est d'être totalement réutilisable, car il ne nécessite pas de soudure.

De plus, il permet une grande variété de composants afin de réaliser des circuits électroniques, du plus simple circuit jusqu'au microprocesseur.

Elle est recouverte de trous (espacés de 0.1") dans lesquels on insère directement les broches des composants électroniques. Sous ces trous, des languettes métalliques jouent un double rôle. Elles agissent d'abord comme des pinces qui retiennent fermement les broches des composants. En plus, ces languettes permettre de connecter électriquement les broches des composants. La disposition des languettes peut varier suivant le type de breadboard.

Par contre, la plupart des modèles de breadboard ont une section plastifiée en leur milieu sur le sens de la longueur. Ces trous sont liés en colonnes et en rangées comme il est illustré dans la figure II. **[18]**

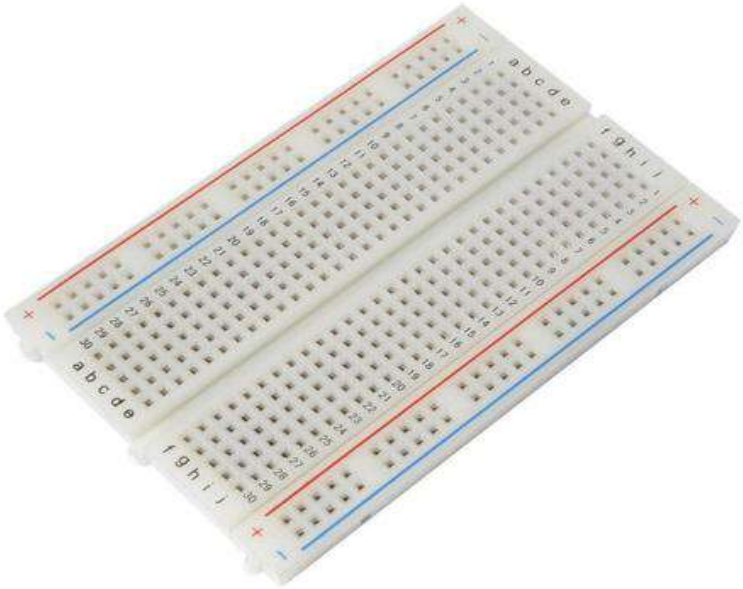


Figure II.29 : Breadboard.

II.3 Partie logicielle

Nous avons utilisé plusieurs logiciels pour réaliser ce travail, dans cette partie nous avons présenté les logiciels de programmation et conception.

II.3.1 Arduino IDE

Arduino (IDE) est un logiciel pour écrire des programmes et de les télécharger sur votre carte.

Dans la page Du logiciel Arduino, vous trouverez deux options :

1. Si vous disposez d'une connexion Internet fiable, vous devez utiliser l'IDE en ligne (Arduino Web Editor). Il vous permettra d'enregistrer vos croquis dans le cloud, en les ayant disponibles à partir de n'importe quel appareil et sauvegardés. Vous disposerez toujours de la version la plus récente de l'IDE sans avoir besoin d'installer des mises à jour ou des bibliothèques générées par la communauté.
2. Si vous préférez travailler hors connexion, vous devez utiliser la dernière version de l'IDE de bureau. **[25]**

II.3.1.1 L'environnement de développement (IDE) de l'Arduino

Arduino (IDE) est un environnement de développement intégré de Microsoft. Même si c'est un IDE Microsoft, vous pouvez l'utiliser sur les ordinateurs Windows, Mac et Linux.

Ce logiciel, Il prend en charge différents langages de programmation, dont C, C++, etc. Ainsi que,Le code que vous écrivez ne requiert que deux fonctions de base :

- La fonction setup () : fonction exécutée à l'initialisation de la carte.
- La fonction Loop () : fonction exécutée en boucle infinie. **[25]**

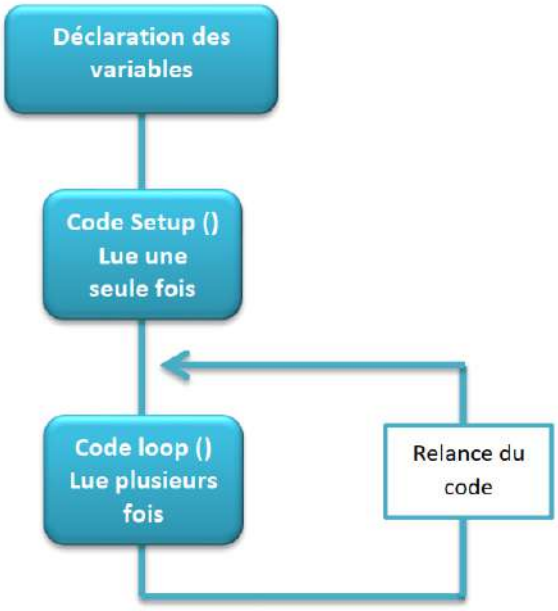


Figure II.30 : Fonctionnement d'un programme Arduino.

II.3.1.2 Structure générale du programme (Arduino IDE)

L'interface de l'IDE Arduino est plutôt simple il offre une interface minimale et épurée pour développer un programme sur les cartes Arduino.

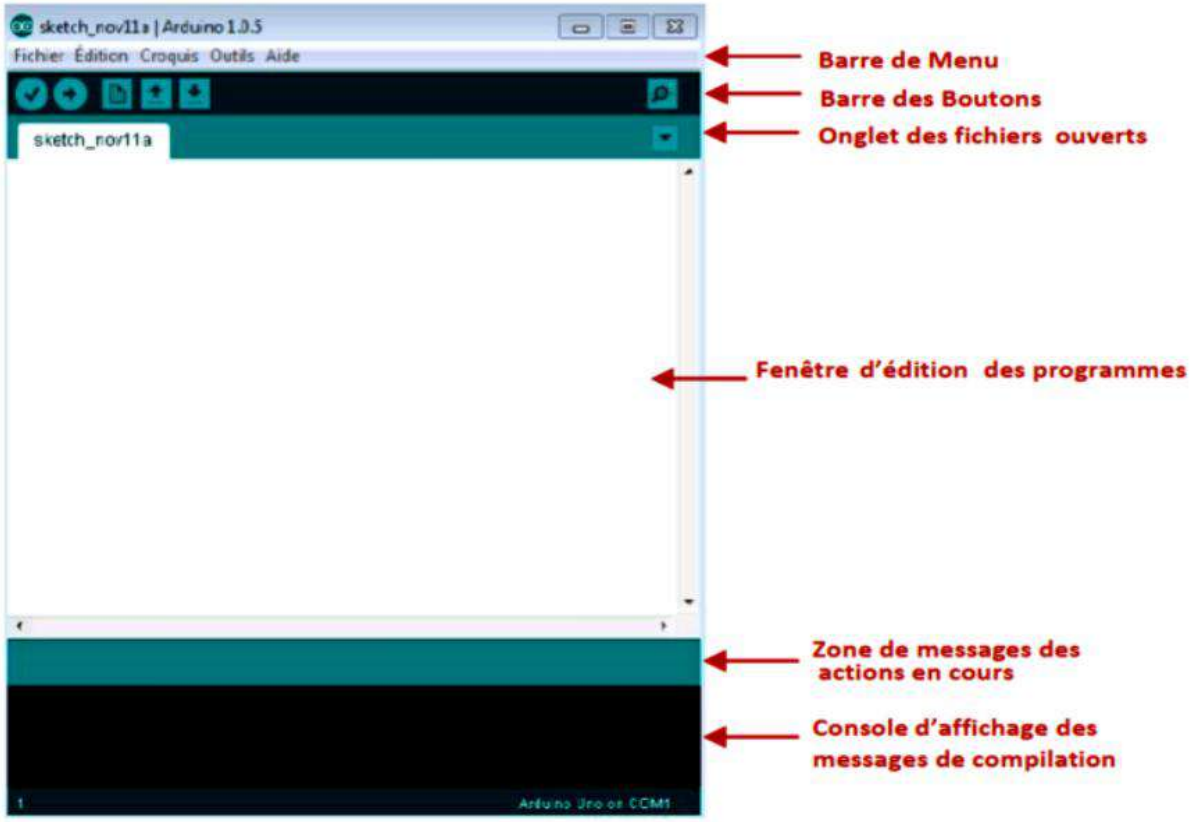


Figure II.31 : L'interface de l'IDE de l'Arduino.

La « Barre de Boutons » qui donne un accès direct aux fonctions essentielles du logiciel est représentée dans la figure II.

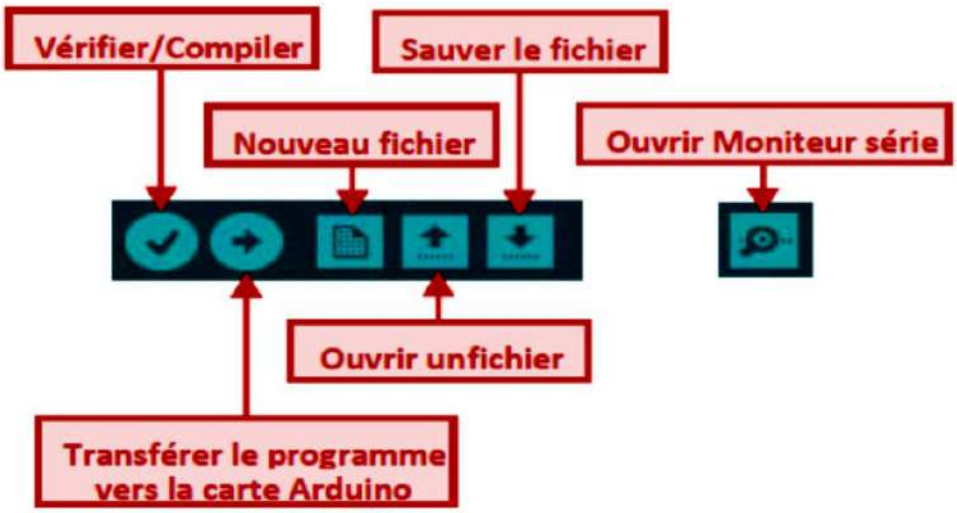


Figure II.32 : Barre de bouton.

II.3.1.3 Description d'un programme Arduino

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

Un programme Arduino est composé de :

Commentaires : Les commentaires sont, en programmation informatique, des portions du code source ignorées par le compilateur ou l'interpréteur, car ils ne sont pas censés à influencer l'exécution du programme. Un commentaire doit commencer par deux slashs (//).

Déclaration des variables : La zone suivante est une zone de déclaration de variables globales. Les variables globales sont des variables communes à l'ensemble des fonctions qui seront décrites par la suite. Il faut faire attention à ce type de variables car elles peuvent être modifiées à tout moment par n'importe laquelle des fonctions qui l'utilisent. Parfois, on préférera des variables locales.

Configuration et initialisation des entrées sorties : Cette étape se fait dans la fonction **void setup ()**. Cette fonction est appelée une seule fois au début de l'exécution du programme. Elle permet d'initialiser l'ensemble des broches d'entrées/sorties ainsi que les différents composants liés à l'application.

Programmation des interactions (exécution) : Cette étape se fait dans la fonction **void loop ()**. Cette fonction est le cœur du programme embarqué. Elle est exécutée une infinité de fois (tant que la carte est alimentée), après l'exécution de la fonction setup ().

II.3.1.4 Injection du programme

Avant d'envoyer un programme dans la carte, il est nécessaire de sélectionner le type de la carte (Arduino UNO) et le numéro de port USB (COM 3) comme à titre d'exemple cette figure suivante.

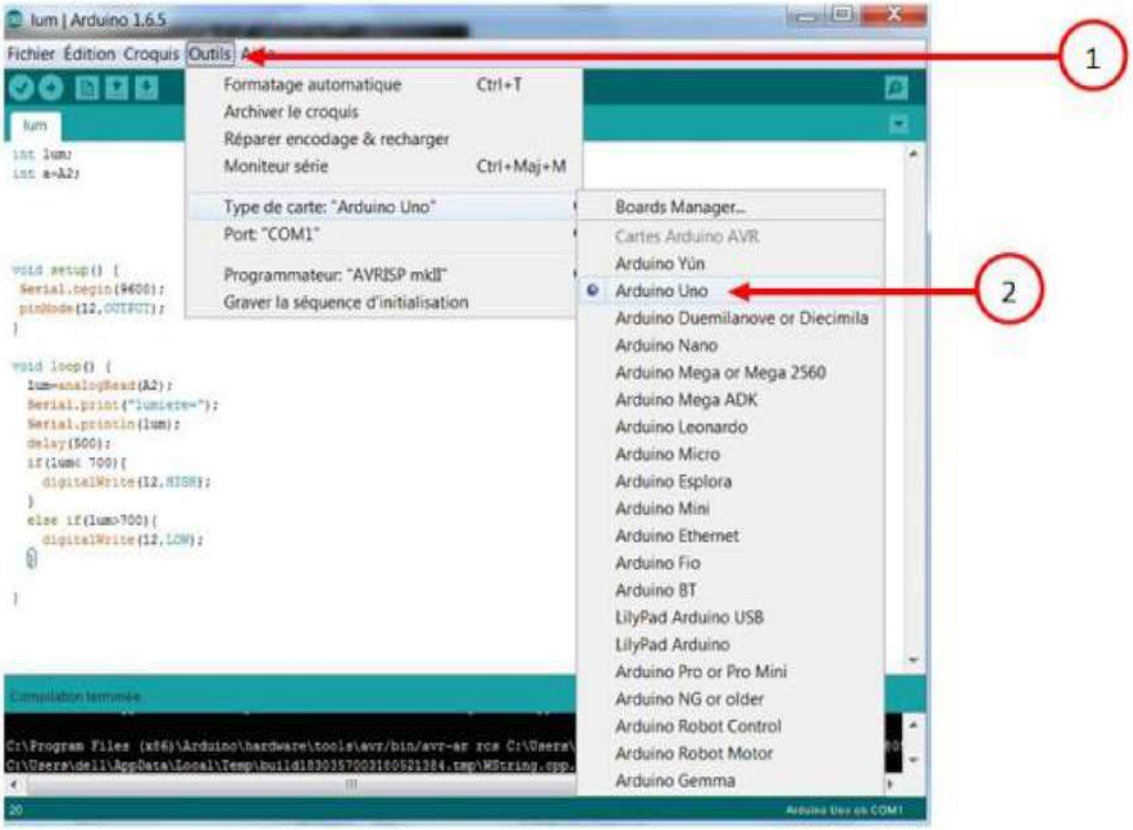


Figure II.33 : Paramétrage de la carte étape 1.

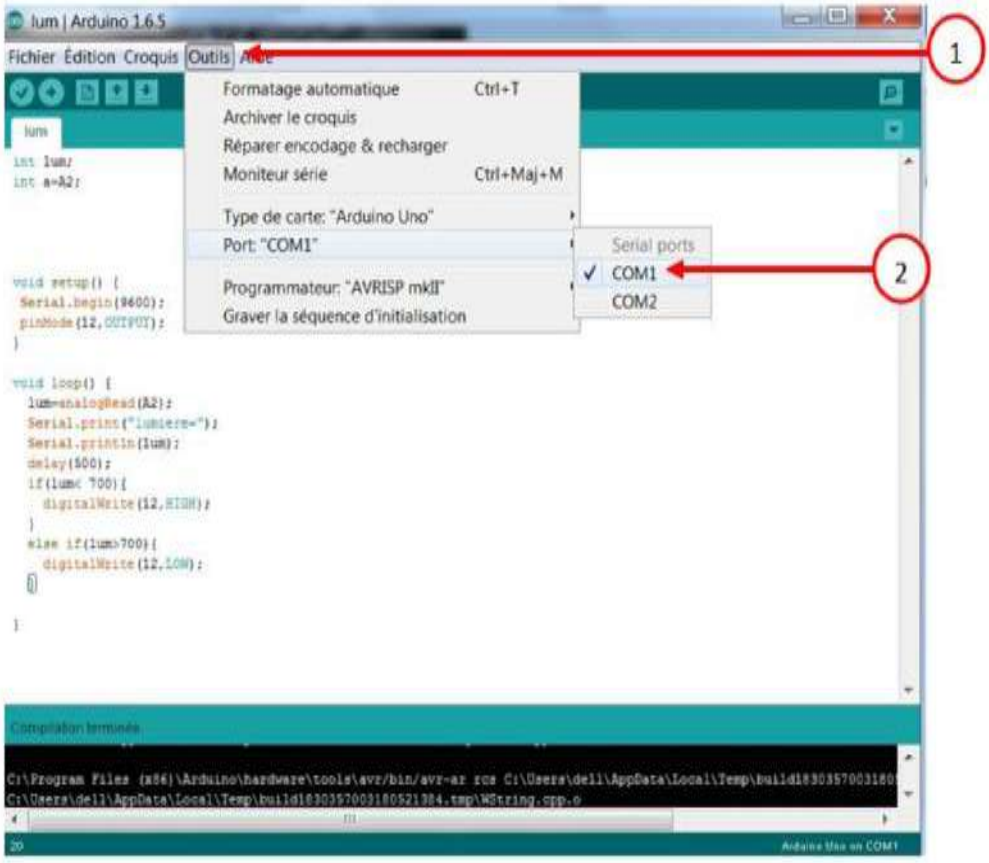


Figure II.34 : Paramétrage de la carte étape 2.

II.3.1.5 Les étapes de téléchargement du programme

Une simple manipulation enchaînée doit être suivie afin d'injecter un code vers la carte Arduino via le port USB.

1. On conçoit ou on ouvre un programme existant avec le logiciel IDE Arduino.
2. On vérifie ce programme avec le logiciel Arduino (compilation).
3. Si des erreurs sont signalées, on modifie le programme.
4. On charge le programme sur la carte.
5. On câble le montage électronique.
6. L'exécution du programme est automatique après quelques secondes.
7. On alimente la carte soit par le port USB, soit par une source d'alimentation autonome (pile 9 volts par exemple).
8. On vérifie que notre montage fonctionne.

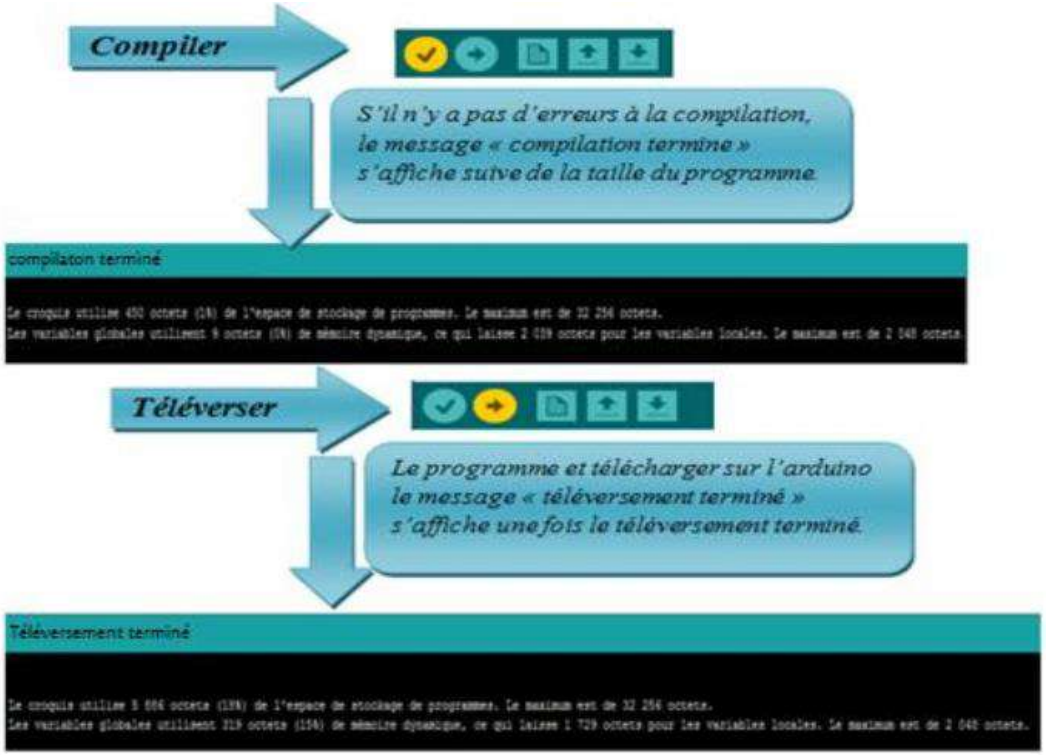


Figure II.35 : Les étapes de téléchargement du code.

II.3.2 logiciel PROTEUS

Proteus Professional est une suite logicielle destinée à l'électronique. Développé par la société Labcenter Electronics, les logiciels incluent dans Proteus Professional permettent la CAO (Construction Assistée par Ordinateur) dans le domaine électronique. Deux logiciels principaux composent cette suite logicielle: ISIS, ARES.

Cette suite logicielle est très connue dans le domaine de l'électronique. De nombreuses entreprises et organismes de formation (incluant lycée et université) utilisent cette suite logicielle. Outre la popularité de l'outil, Proteus Professional possède d'autres avantages :

a• Pack contenant des logiciels facile et rapide à comprendre et utiliser.

b• Le support technique est performant.

c• L'outil de création de prototype virtuel permet de réduire les coûts matériel et logiciel lors de la conception d'un projet. **[26]**

II.3.2.1 ISIS

Le logiciel ISIS de Proteus Professional est principalement connu pour l'édition de schémas électriques.

De plus, le logiciel permet également de simuler ces diagrammes, ce qui permet de détecter certaines erreurs dès la conception. Indirectement, les circuits électriques conçus avec ce logiciel peuvent être utilisés dans des documentations car le logiciel permet de contrôler la plupart des aspects graphiques des circuits. Présentation de l'interface : **[26]**

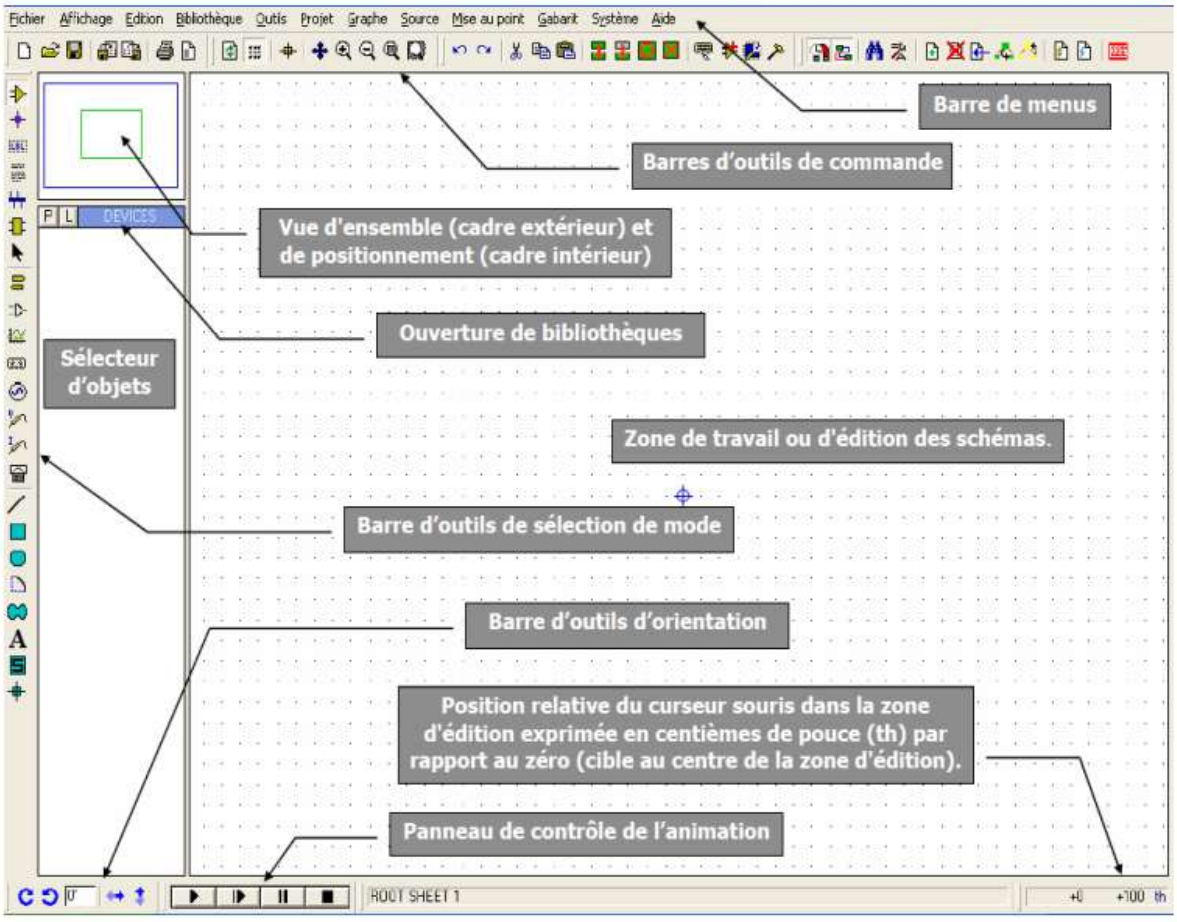


Figure II.36 : L'interface de logiciel ISIS Proteus.

II.3.2.2 ARES

Le logiciel ARES est un outil d'édition et de routage qui complète parfaitement ISIS. Un schéma électrique réalisé sur ISIS peut alors être importé facilement sur ARES pour réaliser le PCB (Printed circuit board) de la carte électronique. Bien que l'édition d'un circuit imprimé soit plus efficiente lorsqu'elle est réalisée manuellement, ce logiciel permet de placer automatiquement les composants et de réaliser le routage automatiquement.

II.3.3 Fritzing

Cette suite logicielle est très connue dans le domaine de l'électronique. De nombreuses entreprises et organismes de formation (incluant lycée et université) utilisent cette suite logicielle. Outre la popularité de l'outil, Proteus Professional possède d'autres avantages :

Il est adapté aux débutants ou confirmés en électronique pour faire rapidement des circuits circuits simple, et est également un bon outils didactiques pour apprendre à bidouiller en électronique par la pratique. **[27]**

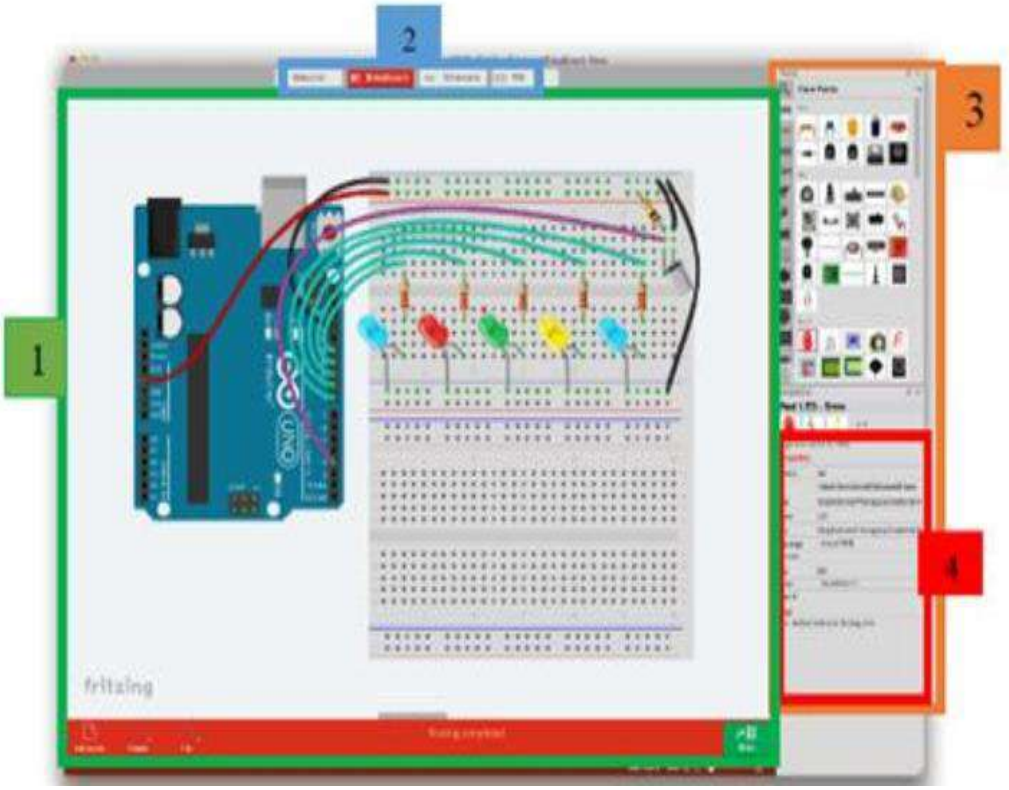


Figure II.37 : L'interface de logiciel Fritzing.

1 : L'espace de travail.

2 : Les déférents vus du circuit.

3 : Palet de composent.

4 : Information sur le composent choisi.

II.3.4 SolidWorks

Le logiciel de conception mécanique SolidWorks est un outil de conception de modélisation volumique paramétré, basé sur des fonctions, qui tire parti des fonctionnalités de Windows TM, connu pour sa convivialité. On peut créer des modèles volumiques 3D entièrement intégrés avec ou sans contraintes tout en utilisant des relations automatiques ou définies par l'utilisateur pour saisir l'intention de conception.

Un modèle SolidWorks est entièrement intégré par rapport aux mises en plan et aux assemblages qui le référencent. Les changements introduits dans le modèle sont entièrement reflétés dans les mises en plan et les assemblages qui lui sont associés. Inversement si on effectue des changements dans le contexte d'une mise en plan ou d'un assemblage, ces changements sont reflétés dans le modèle.

Les relations géométriques telles que les relations parallèles, perpendiculaires, horizontales, verticales, concentriques et coïncidentes sont des exemples des contraintes supportées par SolidWorks. Des équations peuvent également être utilisées pour établir des relations mathématiques entre les paramètres. Par ces moyens on peut garantir que des concepts seront établis et conservés. **[28]**

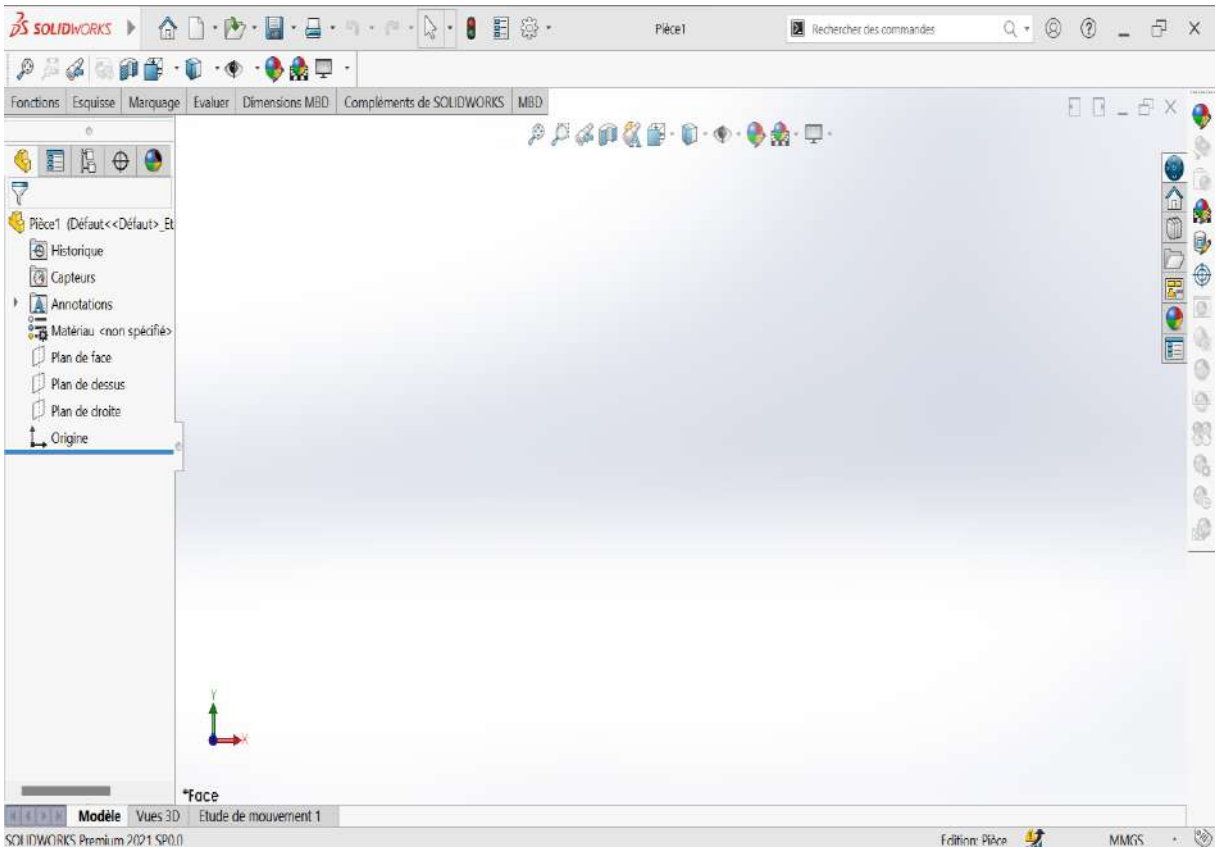


Figure II.38 : L'interface de logiciel Solidworks.

II.4 Conclusion

Dans ce chapitre, nous avons donné une description générale des différents éléments qui constituent notre prototype. Les principaux dispositifs sont la carte Arduino Uno et le module IR et la carte RFID.

Le capteur IR est utilisé pour la détection tandis que la carte Arduino Uno a pour mission de commander les différents périphériques (capteur IR, servomoteur, capteur de température et capteur LDR et l'écran LCD).

La carte RFID permet à régler la sécurité d'un smart parking et contrôler différentes informations (contrôle d'accès, identification, etc…).

CHAPITRE III

Conception et réalisation

III.1 Introduction

Après avoir présenté les différents composants du parking et les outils que nous allons utiliser dans les chapitres précédents, la question que nous abordons dans ce chapitre est la conception et la réalisation de notre projet.

La conception du système doit passer par plusieurs étapes qui sont décrites dans les paragraphes suivants.

III.2 Conception du parking

Cette étape consiste à créer un modèle pour un parking (maquette). Pour cela, nous avons modélisé la structure principale (plan), contenant l'entrée et sortie , les bordures , et un espace de stationnement avec des postes qui contrôle notre parking . Le modèle consiste en une maquette de dimension de 60×40cm.

La Figure III.1 illustre la structure du plan avec le logicielle SolidWorks.

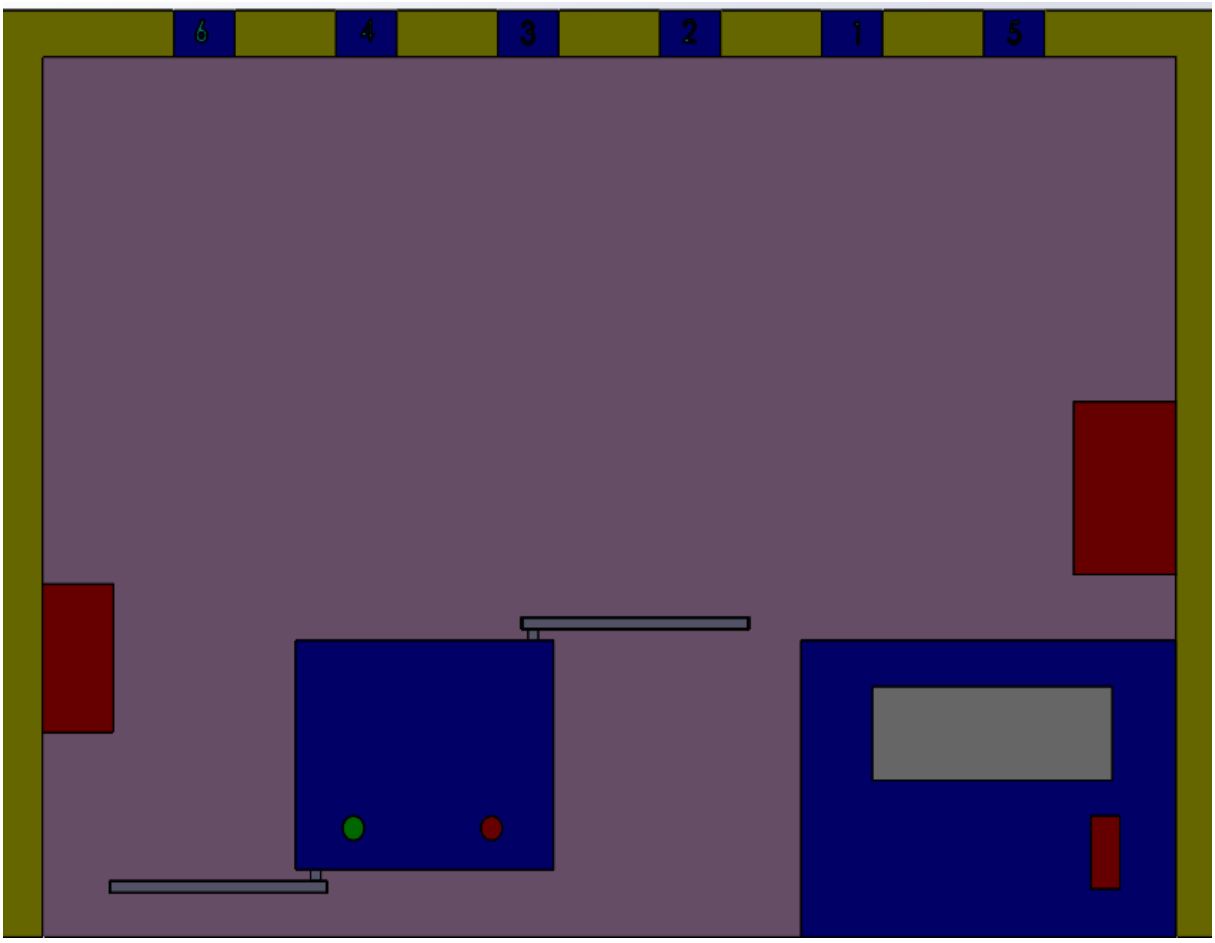


Figure III.1 : Structure du plan de parking.

La conception du système de sécurité est composée de deux parties principales : la première consiste à établir les interconnexions entre des différentes parties du système, tandis que

La deuxième consiste en l'élaboration du programme de fonctionnement du système qui sera implémenté dans les cartes Arduino.

III.2.1 Schémas d'interconnexion des composants

Nous avons utilisé plusieurs de composants qui sont interconnectés sur les Pins d'entrées et sorties de la carte Arduino, pour assurer le fonctionnement du système de sécurité à l'intérieur du parking, tels que l'acquisition et le contrôle des places disponible et les valeur de température, humidité, etc...

La figure III.2 illustré le schéma synoptique du système.

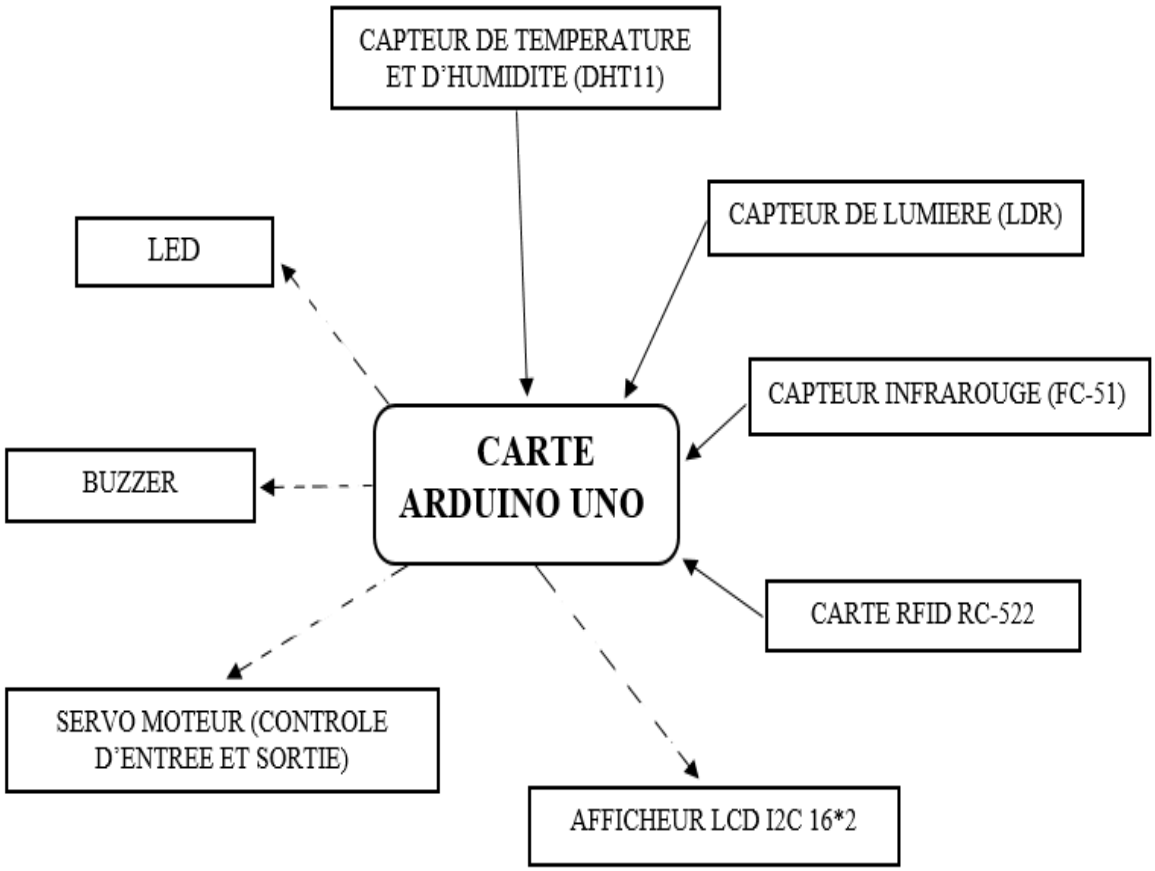


Figure III.2 : Schéma synoptique du système.

III.2.2 Schéma global du système sur le logiciel ISIS

La figure III.3 illustre le schéma global de la réalisation du système sur le logiciel ISIS :

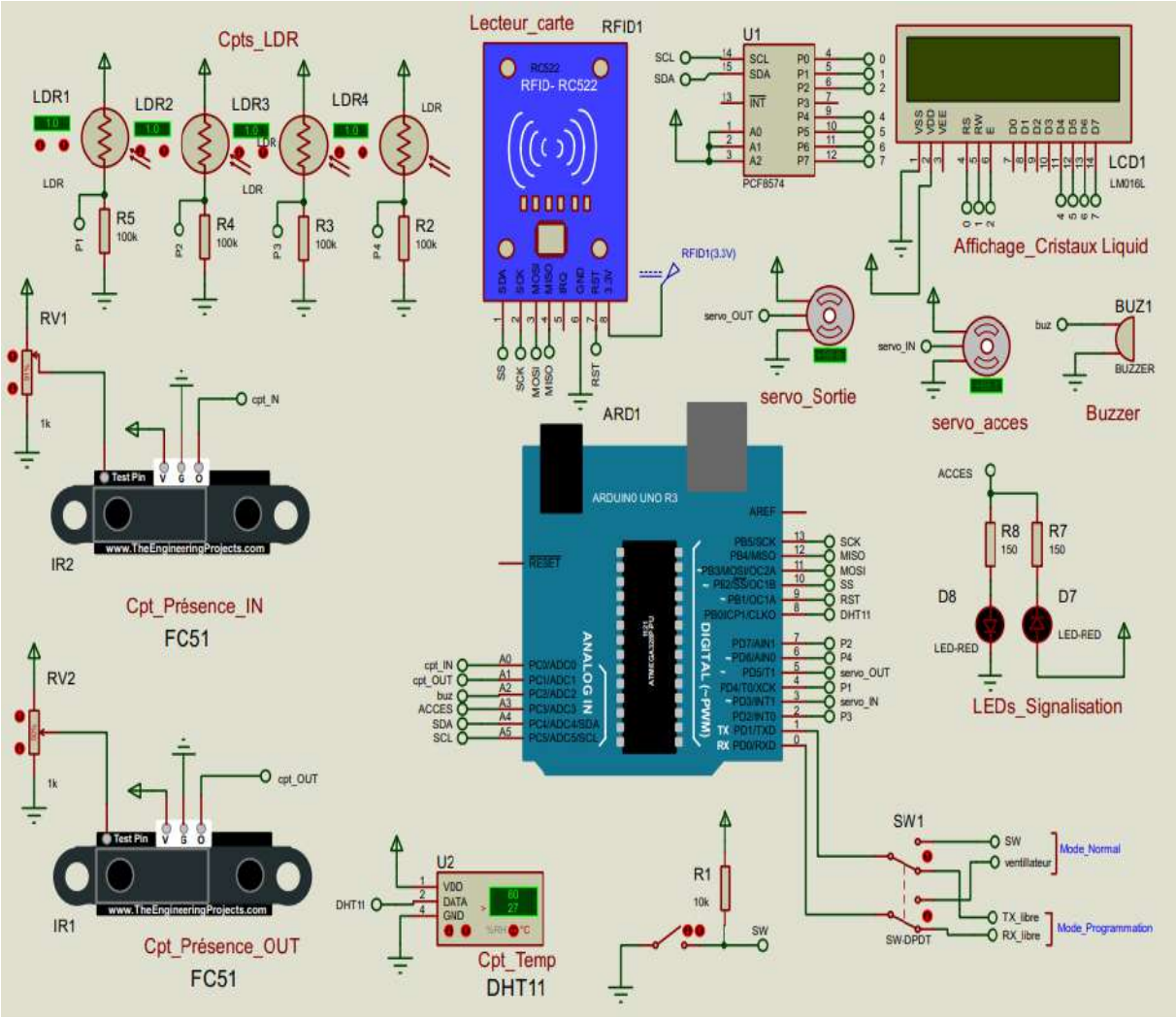


Figure III.3 : Schéma global du système sur le logiciel ISIS.

III.2.3 Contrôle de système

Ce système est composé de plusieurs parties dont chacune assure une tache bien spécifique qui sont détaillées dans les paragraphes ci-dessous :

III.2.3.1 Contrôle de température et d'humidité

Permet d'améliorer la gestion d'énergie au parking. Nous avons inclus un dispositif pour prendre en charge la détection et le contrôle de la température et l'humidité dans la surface du parking.

Pour assurer cette tache on a utilisé : une carte Arduino Uno, un capteur de température et d'humidité (DHT11), et un écran LCD.

La Figure III.4 illustre les interconnexions entre les composants du système d'acquisition de température et d'humidité.

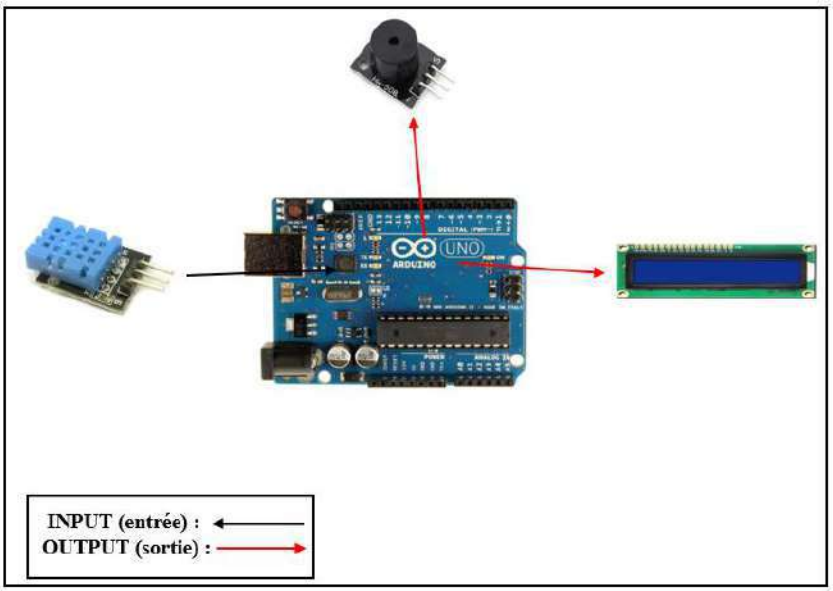


Figure III.4 : Schéma interconnexions entre les composants du système d'acquisition de température et d'humidité.

III.2.3.2 Contrôle d'accès au parking

La mise en place d'un système de contrôle d'accès sécurisé est une contrainte importante pour une bonne sécurité d'un parking. Dans notre système nous avons utilisé le système RFID Pour assurer cette tache on a utilisé : une carte Arduino UNO, un Afficheur LCD 16x2, un Servomoteur placé sur la porte, un Buzzer.

Figure III.5 illustre les Interconnexions entre les composants du système du contrôle d'accès.

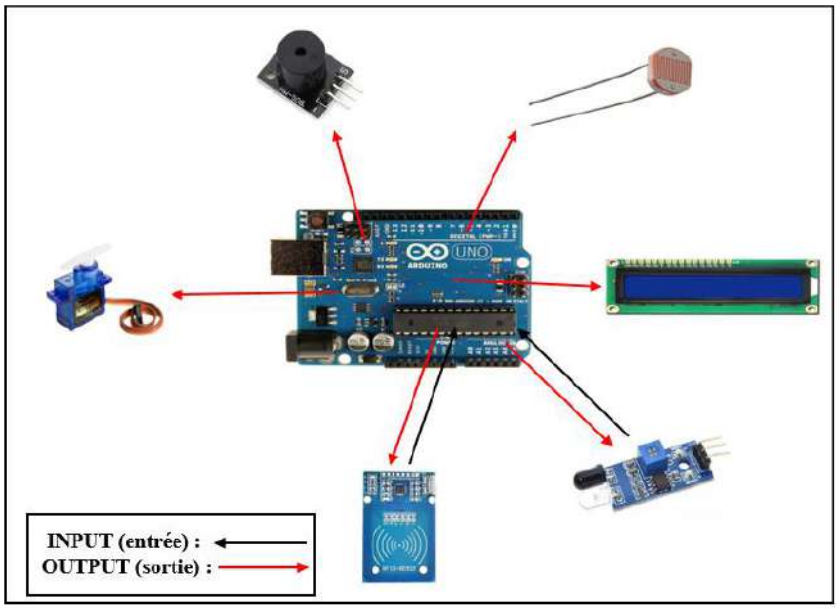


Figure III.5 : Schéma interconnexions entre les composants du système du control d'accès.

III.3 Réalisation du parking

III.3.1 Description du prototype

Notre projet consiste a réalisé d'un parking automatisé (smart parking) à base d'un carte arduino Uno. Pour mieux comprendre le principe de fonctionnement de notre prototype nous avons divisé notre projet en trois parties :

- Accès au parking.
- Gestion du parking.
- Affichage des informations.

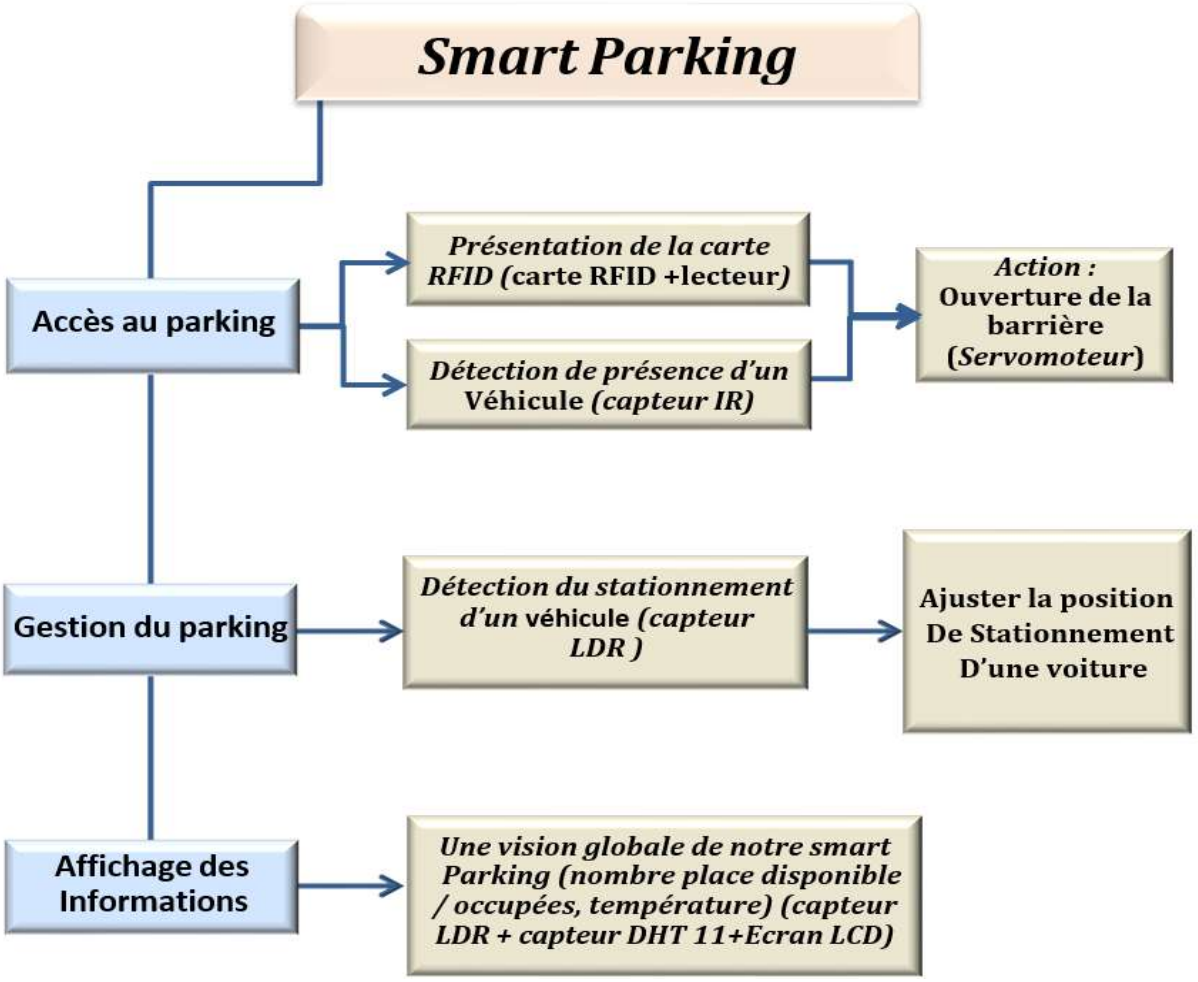


Figure III.6 : Représentation schématique du prototype.

III.3.1.1 Bloc d'accès au parking

Cette bloc d'accès est assuré en utilisant : un système RFID (carte RFID + lecteur) et capteur IR et servomoteur.

Ces trois éléments sont connectés à une carte Arduino Uno dont la sortie du capteur IR est liée avec une entrée analogique de la carte Arduino tandis que le servomoteur est branché avec

une sortie numérique de l'Arduino Uno. En plus, la carte RFID est également connectée de la carte arduino Uno via plusieurs câbles des liaisons.

Sans oublier, pour que le client abonné puisse accéder au parking il doit être présentée de la carte RFID.

III.3.1.2 Bloc de la gestion du parking

Le bon fonctionnement de ce bloc nécessite l'utilisation d'une carte Arduino Uno et des capteurs IR. En effet, une fois que le véhicule est accédé au parking, il suffit de choisir un emplacement pour stationner.

Après le stationnement, le capteur IR détecte la présence de cette voiture. Il délivre un signal électrique à la carte Arduino ATmega 328. Cette dernière est connectée via une connexion à l'aide du câble USB avec l'application déjà installé sur l'ordinateur.
Ceci nous permet de visualiser l'emplacement occupé par le véhicule.

III.3.1.3 Bloc d'affichage des informations

Ce Bloc est réservé pour l'affichage de nombres de places libres / occupées et température. Pour cela il faut utiliser un écran LCD connecté avec le module I2C.

III.3.2 Comment fonctionne ce système ?

Lors de la mise en marche de notre circuit, la carte Arduino Uno s'est connectée au smart parking. Et l'écran LCD affiche un message d'accueil « Bienvenue Smart parking ».



Figure III.7 : Affiche un message d'accueil « Bienvenu Smart parking ».

Après quelque temps (3 seconde) L'écran LCD affiche le nombre de places disponibles (libre). Ainsi que la température.



Figure III.8 : Nombre de places libres et la température.

Dans Le cas où le véhicule représente un client abonné, l'accès au parking nécessite l'utilisation d'un système RFID. Ce système est constitué d'une carte RFID (ou tag) et d'un lecteur RFID (module RC522).

Ce dernier est équipé d'une puce intégrée. Cette puce permet de transmette l'information (Numéro de série) vers le lecteur RFID sous la forme d'ondes radio via une antenne intégrée. Ce module utilise le protocole SPI (Serial Périphérique Interface) pour communiquer avec la carte arduino uno. La communication SPI utilise des boches spécifiques des microcontrôleurs ATméga 328. Le brochage se fait comme suit :

RFID RC-522	Arduino Uno
VCC	3V3
GND (Masse)	GND
RST (Reset)	Pin 9
SPI SS/SDA (Slave select)	Pin 10
SPI MOSI (Master Output Slave Input)	Pin 11
SPI MISO (Master Input Slave Output)	Pin 12
SPI SCK (Serial Clock)	Pin 13

Tableau III.1 : Le brochage du module RC522 avec Arduino Uno.

Dans ce cas, pour que l'abonné puisse accéder au parking, il lui suffit de présenter sa carte RFID (carte ou tag).

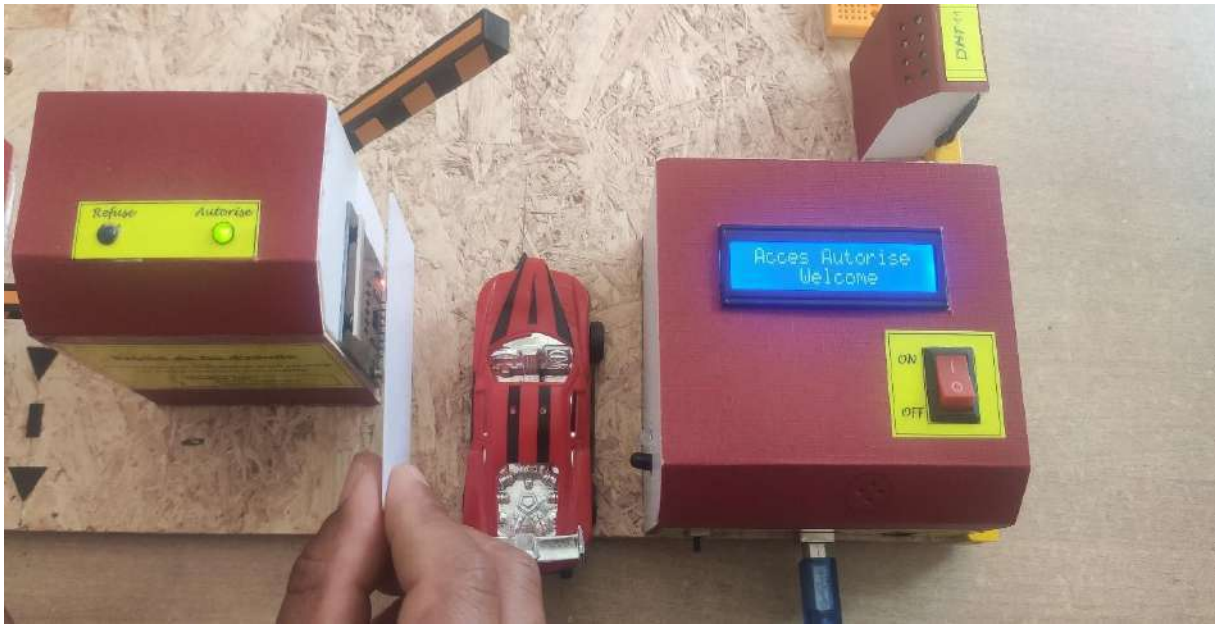


Figure III.9 : Présenter sa carte RFID valide à l'entrée.

Une fois que les informations de la carte RFID sont réceptionnées puis convertis par lecteur RFID, il faut comparer ces données avec les codes déjà enregistrés dans l'arduino IDE.

Si le code de la carte RFID du client correspond bien au code déjà enregistrer dans l'Arduino IDE, la sortie de cette carte envoie un signal électrique. Par conséquent, le servomoteur est excité et change de position. Ceci permet l'ouverture de la barrière et autorise le client abonné d'accéder au parking privé.



Figure III. 10 : Détection d'une voiture en cas d'entrée.

Par conséquent, le nombre de places disponibles et la température sont visualisées sur l'écran LCD.



Figure III.11 : L'afficheur LCD en cas d'entrée d'une voiture.

Une fois accédé au parking, le conducteur devra choisir un emplacement afin de stationner. Cet emplacement est équipé par des capteurs LDR . Ces capteur LDR sont connectes à une carte Arduino Uno.

Numéro du capteur LDR	Arduino Uno
Capteur 1 (place 1)	Pin 4
Capteur 2 (place 2)	Pin 7
Capteur 3 (place 3)	Pin 2
Capteur 4 (place 4)	Pin 6

Tableau III. 2 : Branchement des capteurs LDR avec la carte Arduino Uno.

En fin, lorsque la voiture quitte son emplacement, l'écran LCD nous indique le nombre de place libre.

Pour que le véhicule puisse quitté le parking, il suffit de se dirigé vers la sortie de parking.

Cette sortie est équipée par un capteur IR et un servomoteur commande par carte Arduino UNO (voir la figure III.12).

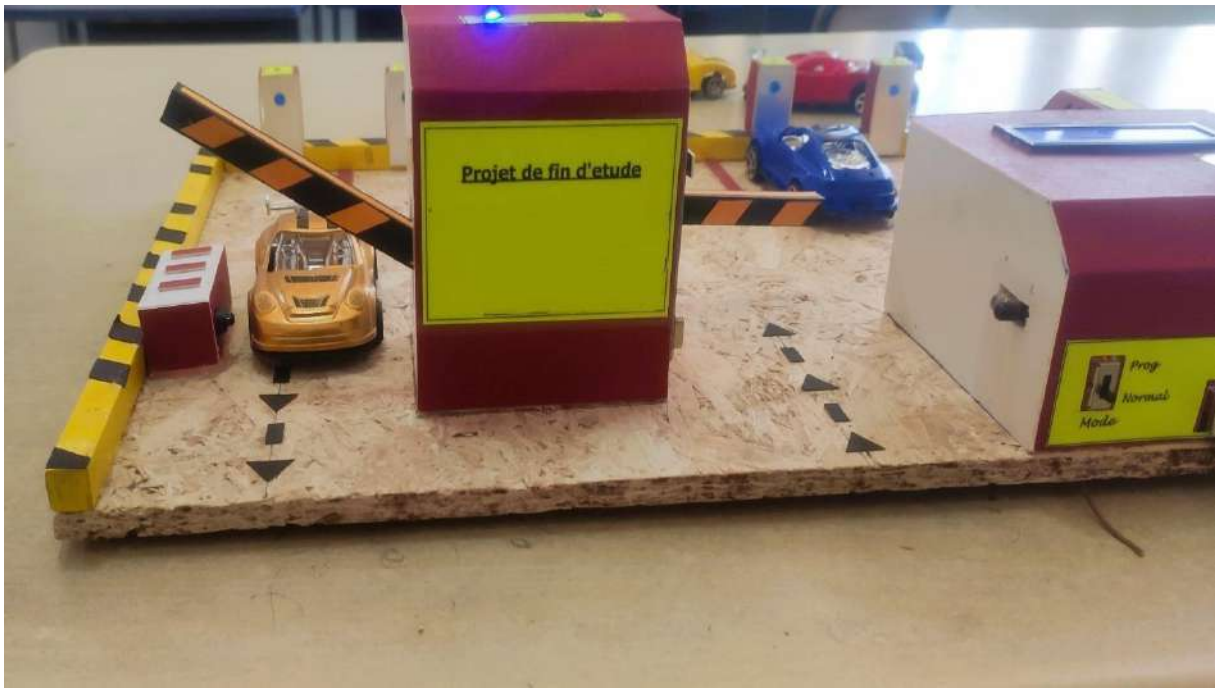


Figure III.12 : Détection d'une voiture en cas de sortie.

Une fois que la voiture quitte définitivement le parking, le nombre des places libres s'incrémentes. Ces données sont affichées sur l'écran LCD (voir la figure III.13).



Figure III.13 : L'afficheur LCD en cas de sortie deux voiture.

Afin de renforcer les mesures de sécurité nous avons installé un détecteur de la température.

Ce système est constitué d'un capteur température et d'un Buzzer. Ces deux dispositifs sont commandés par la carte Arduino Uno.

En effet, lors d'un incendie, le capteur détecte l'augmentation de la température au delà 36 dégrée (le chiffre réglée dans notre maquette) et le réglage des chiffres qui excite notre dispositif est dépend de la saison et la zone de parking.

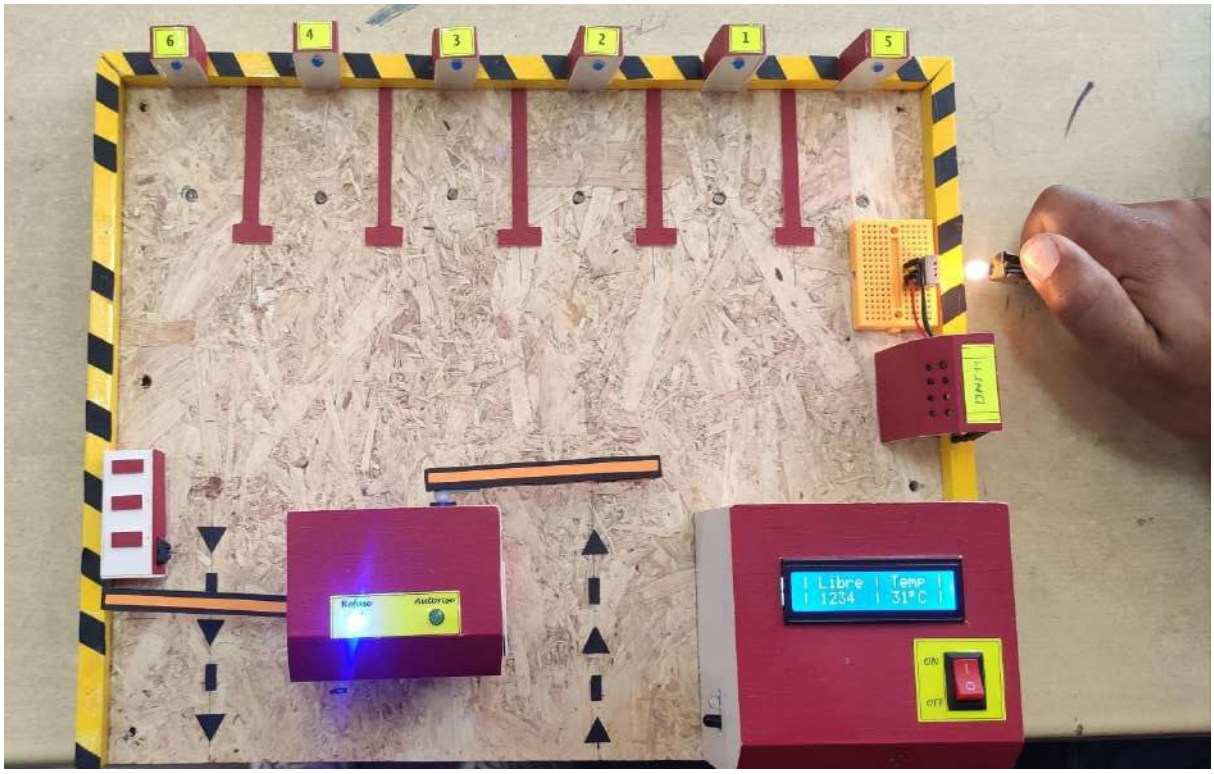


Figure III. 14 : La détection d'incendie.

Par la suite, ce signal électrique est transmis à la carte Arduino UNO. De ce fait, le système d'alerte s'active et le BUZZER se déclenche (un effet sonore).

Une fois déclencher, la carte Arduino transmis un signal numérique à l'afficheur LCD. Ce dernier converti ce signal et affiche le message « Alerte incendie ». Par conséquent l'entrée reste fermée et la sortie ouvre spontanément.

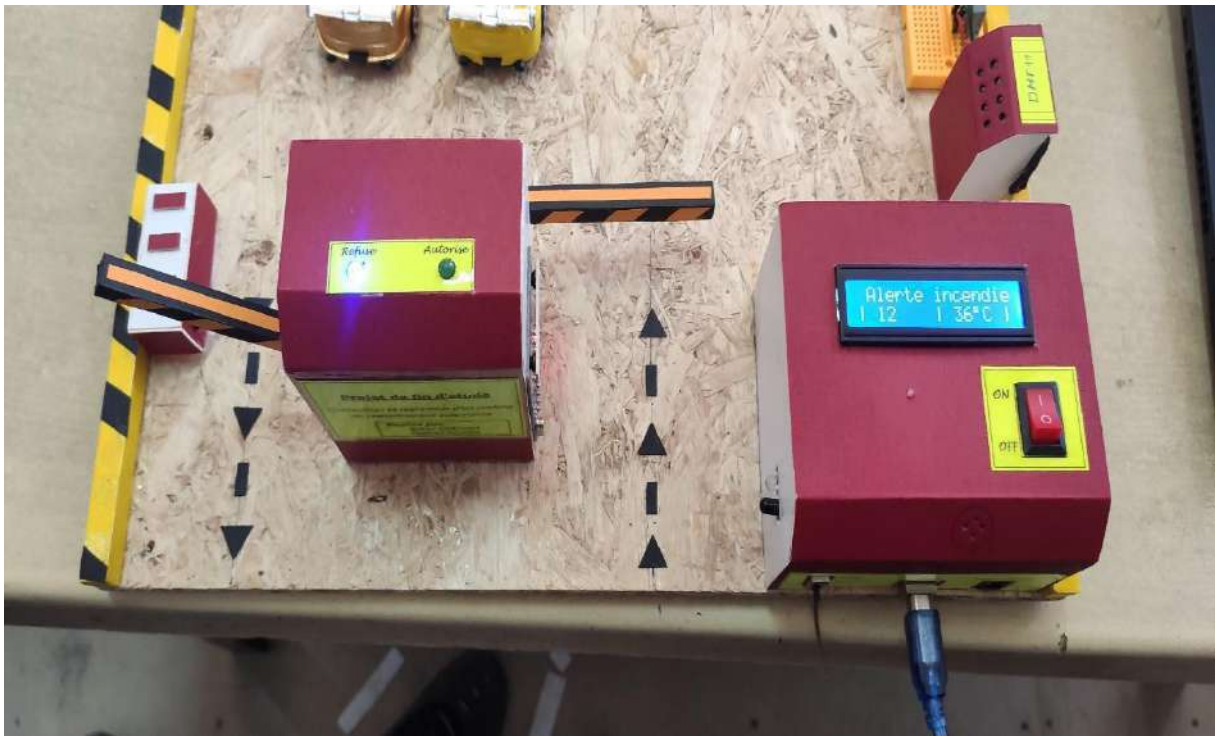


Figure III. 15 : Affichage du message d'alerte en cas d'incendie.

Les figures III.16 et III.17 représentants une vue globale de notre prototype :



Figure III.16 : représentation globale du smart parking.

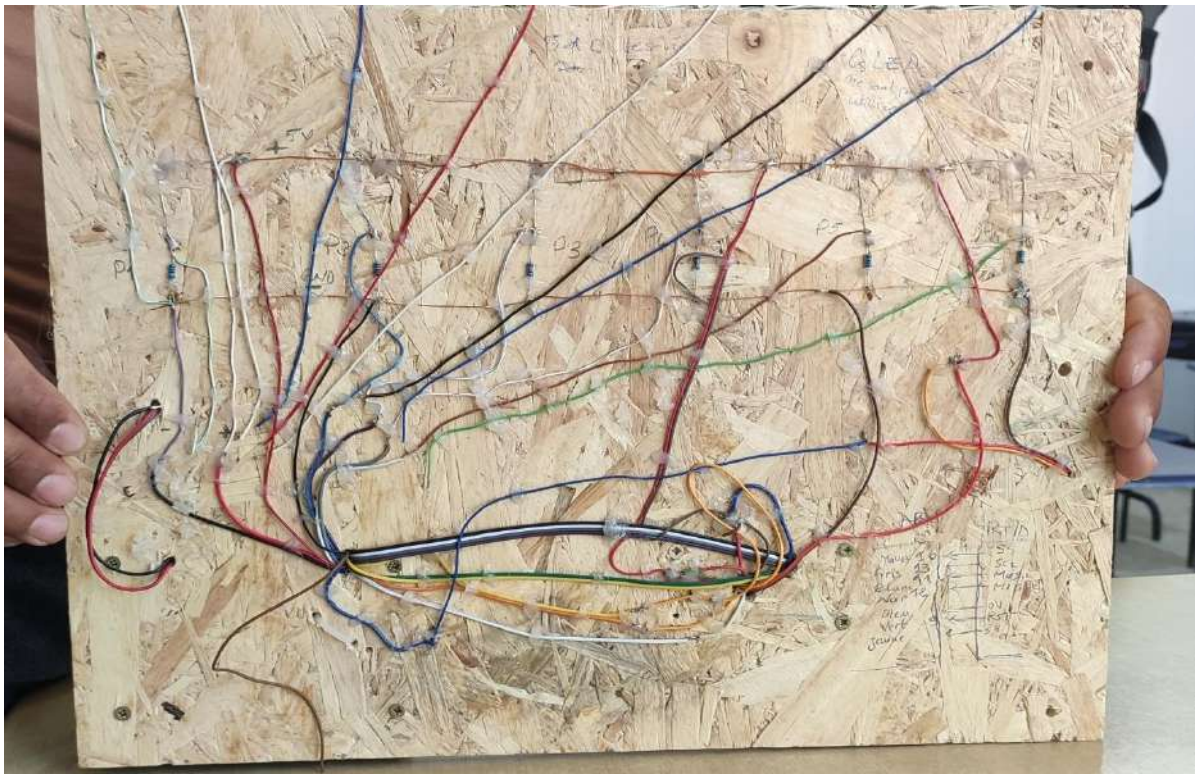


Figure III.17 : représentation globale de circuit du smart parking.

Conclusion

Dans ce chapitre, nous avons élaboré un système pour le parking à sécuriser tenant compte de son plan et de sa structure. Ensuite nous avons détaillé les différentes étapes de conception et de réalisation du système de sécurité.

A la réalisation, nous avons présentée trois blocs : le bloc d'accès au parking, le bloc de gestion du parking et le bloc d'affichage des informations.

Une fois réalisé, plusieurs essais ont été effectués. Ces tests nous a permis de valider le bonne fonctionnent de notre système.

Conclusion générale

Dès l'aube de l'humanité, l'homme cherche à se faciliter sa vie et à protéger ses propriétés contre toute sorte de risque.

Nous nous sommes intéressés à travers ce projet à développer un outil permettant les gens à trouver un endroit sécurisé (contre intrusion, hôte température, incendie, humidité) et d'accés réservé pour leur véhicule, la commande est automatique et à distance sur les actionneurs (porte, led) à l'aide d'une application sur un écran LCD qui à connecté via infra-rouge.

La conception de ce projet nous a permis d'acquérir des connaissances supplémentaires en conception et en développement, en fait, nous avons pu accéder à la maîtrise du langage d'Arduino, également appris à travailler sur des cartes programmables ayant des capacités et des performances élevées, à savoir Arduino.

Dans notre travail, nous avons utilisé une méthode qui repose sur l'installation des capteurs (température et humidité, LDR, IR), et les relier à des cartes électroniques Arduino, en plus nous avons fait la liaison entre les Arduino, ces cartes Arduino gèrent l'ensemble de ces capteurs, modules, actionneurs et déclenchement, en fonction de la situation, une certaine signalisation d'alarme et agissent convenablement à chaque événement détecté. L'application sur l'écran LCD permet à l'utilisateur de reconnaître la nature de l'incident.

Notre projet a abouti au développement d'un système qui contrôle l'accés réservé au parking et détecte touts risques naturels au humaines, les essais ont donné des résultats très satisfaisante et efficace, ce qui nous permet de le réaliser réellement.

La réalisation de ce projet nous a énormément appris, autant au niveau de l'électronique, de la programmation des cartes électroniques. Nous avons aussi appris de nouvelles connaissances au niveau de la gestion du temps et des équipes.

Ce travail reste, comme toute œuvre humaine, incomplète et perfectible, nous recommandons d'en améliorer la conception et pour cela nous proposons ci-dessous des améliorations pour les futurs développements :

- Améliorer le système de sécurité par des caméras de surveillance.
- Ajouter des nouvelles fonctionnalités pour améliorer le confort (message vocaux, …).
- On peut aussi utiliser des modèles wifi, internet, pour une large possibilité de transmission et de réception d'informations.

Nous espérons que ce modeste travail sera utile et sera enrichi par les promotions futures.

Bibliographie

[1] : https://mechapark.ch/fr/2021/07/13/comment-fonctionne-un-systeme-de-stationnement-automatise/

[2] : Tikialine Y, Khedda K. 2018 « Etat de l'art de smart parking, étude et conception d'un prototype de stationnement intelligent », », Projet de fin d'étude pour l'obtention du diplôme de Master en Génie électrique, Université Djilali Bounaama - khemis Miliana

[3] : https://www.evo-park.com/guide-comparatif-types-parking

[4] : Katir H, Houdjedje Y. 2017 **«** Conception d'un système de parking automatisé pour les voitures légères », Projet de fin d'étude pour l'obtention du diplôme de Master en Génie industriel, Université Abou Bekr Belkaid -Tlemcen

[5] : Benfriha L, Bouaicha Y. 2018 « Study and design of an automated parking : the structure », Projet de fin d'étude pour l'obtention du diplôme de Master en Génie Mécanique, Université Kasdi Merbah - Ouargla

[6] : Amnache A, Ouadi A. 2011 **«** Gestion d'un ensemble de parkings automobiles », Projet de fin d'étude pour l'obtention du diplôme de Master en Génie électrique, Université Mouloud Mammeri de TIZI OUZOU

[7] : https://skyline-parking.com/automated-parking-systems/aps-benefits

[8] : https://www.bookingninjas.com/blog/pros-and-cons-of-parking-management-systems

[9] : https://vdocuments.net/arduino-uno-datasheet-569d589fca336.html

[10] : https://www.microchip.com/en-us/product/ATmega328

[11] : https://components101.com/motors/servo-motor-basics-pinout-datasheet

[12] : https://www.carnetdumaker.net/articles/controler-un-servomoteur-avec-une-carte-arduino-genuino/

[13] : https://osoyoo.com/2018/12/21/arduino-lesson-ir-obstacle-avoidance-module/

[14] : https://www.electronicshub.org/ir-sensor/

[15] : https://www.playembedded.org/blog/detecting-obstacle-with-ir-sensor-and arduino/

[16] : https://www.electroduino.com/ldr-sensor-module-how-ldr-sensor-works

[17] : Makouf S. 2020 « Conception d'un automate programmable à base d'arduino » Projet de fin d'étude pour l'obtention du diplôme de Master en Génie électrique, Université Mohamed Khider Biskra

[18] : Ketreb I, Lakbal A. 2019 « Conception et réalisation d'un système de sécurité pour une maison à base de communication Arduino GSM » Projet de fin d'étude pour l'obtention du diplôme de Master en Génie électrique, Université Mouloud Mammeri de TIZI-OUZOU

[19] : https://www.elprocus.com/a-brief-on-dht11-sensor/

[20] : https://lastminuteengineers.com/i2c-lcd-arduino-tutorial/

[21] : https://rfid.ooreka.fr/comprendre/systeme-rfid

[22] : https://lastminuteengineers.com/how-rfid-works-rc522-arduino-tutorial/

[23] : Djeddou D., 2012 « Conception d'un outil de simulation des protocoles d'authentification des systèmes RFID » Projet de fin d'étude pour l'obtention du diplôme de Master en Mathématique et l'informatique Université de M'SILA

[24] : https://www.sonelec-musique.com/electronique_theorie_buzzers.html

[25] : https://www.arduino.cc/en/Guide

[26] : Bairi A., Ammour A., 2019 « Réalisation d'un système de stationnement intelligent » Projet de fin d'étude pour l'obtention du diplôme de Master en électronique, Université Abdelhamid Ibn Badis Mostaganem

[27] : Nafa N, Arabi B. 2016 « Conception et réalisation d'un système de sécurité commandé à distance » Projet de fin d'étude pour l'obtention du diplôme de Master en Génié électrique, Université de Boumerdes

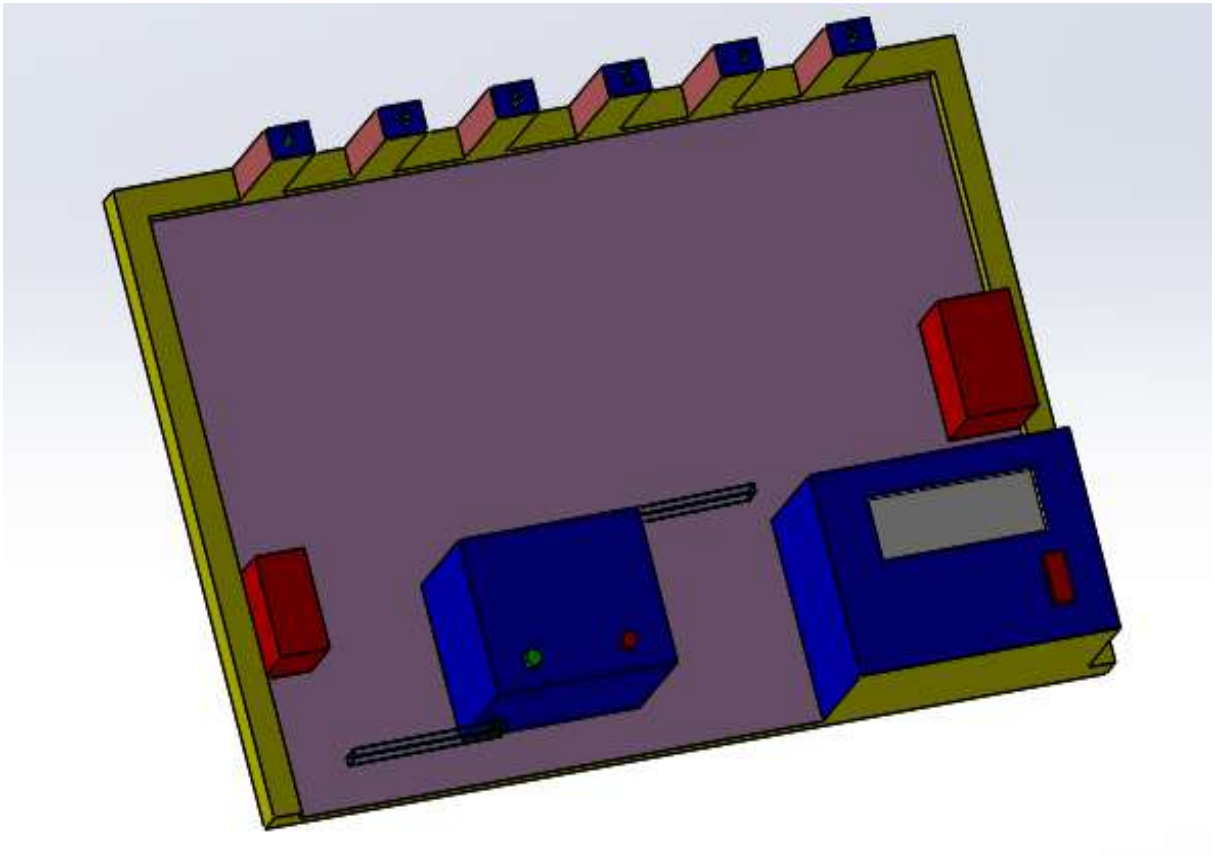
[28] : Belabbas A, Mimoune Z. 2020 **«** Conception et simulation d'une machine vibratoire avec Solidworks **»** Projet de fin d'étude pour l'obtention du diplôme de Master en Génié mécanique, Université de M'SILA

Annexe

Tableau II.2 : Les caractéristiques de la carte Arduino Uno.

Spécifications techniques	
Microcontrôleur	ATmega328
Tension de fonctionnement	5 V
Tension d'entrée (recommandé)	7 à 12 V
Tension d'entrée (limites)	6 à 20 V
Broches d'E/S numériques	14 (dont 6 disposent une sortie PWM)
Broches d'entrée analogique	6 (10bits)
Courant continu par broche E/S	40 mA
Courant continu pour la broche 3.3V	50 mA
Mémoire Flash	32 KB (dont 0.5 **KB** sont utilisés par le boot loader (chargeur d'amorçage))
Vitesse d'horloge	16 MHz
EEPROM	1 KB
SRAM	2 KB
Interface	USB (permet d'alimenté la carte et de transféré les programmes vers la carte)
Alimentation externe	Jack
Dimensions	6.86 cm * 5.3 cm

Plan 3D de la maquette



Programme

```
//.....
//..... Déclaration des librairies .....
//.....
#include <Servo.h>
#include <SPI.h>
#include <MFRC522.h>
#include <LiquidCrystal_I2C.h>
LiquidCrystal_I2C lcd(0x27,16,2);
//.....Déclarer les deux servos.....
Servo bar_IN;
Servo bar_OUT;
//.....Déclarer l'objet RFID .....
#define SS_PIN    10
#define RST_PIN   9
MFRC522 rfid(SS_PIN, RST_PIN);
//.....Déclarer le capteur température .....
#include "DHT.h"
#define DHTPIN 8     //#define DHTPIN A3
#define DHTTYPE DHT11
DHT dht(DHTPIN, DHTTYPE);
//.....Déclarer les différentes constantes et variables nécessaires.....
#define Buzzer A2
#define SW 1
#define servo_IN 3
#define servo_OUT 5
#define cpt_IN  A0
#define cpt_OUT A1
#define Acces A3
#define place1 4
#define place2 7
```

```
#define place3 2
#define place4 6
byte code[4];
byte code_autorise1[4] = {0xD7, 0x20, 0x55, 0x0A};  //  8B 70 60 0D
byte code_autorise2[4] = {0x8B, 0x70, 0x60, 0x0D};
bool ouvre_barriere=false;
bool Alarme=false;
int  temp;
int temp_cel;
float h ;
float t ;
long now = millis();
long lastMeasure = 0;
bool presence_vehicule=true;   // voiture dans le parking
bool servo_true=true,Parking_Complet=false;
//.....
//.....configuration et initialisation.....
//.....
void setup () {
  lcd.init ();
  lcd.clear();
  lcd.backlight();
  pinMode(Buzzer,OUTPUT);
  pinMode(cpt_IN,INPUT);
  pinMode(cpt_OUT,INPUT);
  pinMode(SW,INPUT);
  pinMode(place1,INPUT);
  pinMode(place2,INPUT);
  pinMode(place3,INPUT);
  pinMode(place4,INPUT);
  pinMode(Acces,OUTPUT);
  digitalWrite(Acces,LOW);
```

```
  bar_IN.attach(servo_IN);
  bar_OUT.attach(servo_OUT);
  bar_IN.write(100);
  bar_OUT.write(82);
  lcd.setCursor(0,0);
   lcd.print("== Bienvenue ==");
  lcd.setCursor(0,1);
  lcd.print(" Smart Parking  ");
  delay(3000);
  tone(Buzzer,1000);
  delay(1000);
  noTone(Buzzer);
  dht.begin();
  SPI.begin(); // init SPI bus
  rfid.PCD_Init(); // init MFRC522
  lcd.setCursor(0,0);
  lcd.print("| Libre | Temp |");
  lcd.setCursor(0,1);
  lcd.print("| 1234  |      |");
  etat_places();
  delay(5000);
}
//.....
//..... .fonction etat des places.....
//.....
void etat_places(){
   lcd.setCursor(0,0);
   lcd.print("| Libre | Temp |");
   if(digitalRead(place1)==LOW && digitalRead(place2)==LOW &&
digitalRead(place3)==LOW && digitalRead(place4)==LOW){
     lcd.setCursor(1,1);
     lcd.print("Complet");
```

```
 Parking_Complet=true;
 //return;
}else {
 Parking_Complet=false ;
 lcd.setCursor(1,1);
 lcd.print(" ");
 lcd.setCursor(6,1);
 lcd.print(" ");
 lcd.setCursor(7,1);
 lcd.print(" ");
}
if(digitalRead(place1)==LOW && Parking_Complet==false){
   lcd.setCursor(2,1);
   lcd.print(" ");                //  place N°1 occupée
 }else if(digitalRead(place1)==HIGH && Parking_Complet==false){
  lcd.setCursor(2,1);
  lcd.print("1");
 }
 if(digitalRead(place2)==LOW && Parking_Complet==false){
   lcd.setCursor(3,1);
   lcd.print(" ");                //  place N°2 occupée
 }else if(digitalRead(place2)==HIGH && Parking_Complet==false){
  lcd.setCursor(3,1);
  lcd.print("2");
 }
 if(digitalRead(place3)==LOW && Parking_Complet==false){
   lcd.setCursor(4,1);
   lcd.print(" ");                //  place N°3 occupée
 }else if(digitalRead(place3)==HIGH && Parking_Complet==false){
  lcd.setCursor(4,1);
  lcd.print("3");
 }
```

```
    if(digitalRead(place4)==LOW && Parking_Complet==false){
      lcd.setCursor(5,1);
      lcd.print(" ");                  // place N°4 occupée
    }
else if(digitalRead(place4)==HIGH && Parking_Complet==false){
     lcd.setCursor(5,1);
     lcd.print("4");
    }
 }
//.....................................................................
//.....................Fonction lecture RFID.....................................
//.....................................................................
void lire_rfid(){
  if (rfid.PICC_IsNewCardPresent()) { // new tag is available
    if (rfid.PICC_ReadCardSerial()) { // NUID has been readed
      tone (Buzzer, 700);
      delay(800) ;
      noTone(Buzzer);
     for (byte i = 0; i < 4; i++){
       code[i] = rfid.uid.uidByte[i];
     }
     if (  (code[0] == code_autorise1[0] || code[0]==code_autorise2[0])  &&
          (code[1] == code_autorise1[1] || code[1] ==code_autorise2[1]) &&
          (code[2] == code_autorise1[2] || code[2] ==code_autorise2[2]) &&
          (code[3] == code_autorise1[3] || code[3] ==code_autorise2[3])){
             digitalWrite(Acces,HIGH);
             lcd.clear();
             lcd.setCursor(1,0);
             lcd.print("Acces Autorise");
             lcd.setCursor(0,1);
             lcd.print("    Welcome     ");
             ouvre_barriere=true;
```

```
      }
      else{
       ouvre_barriere=false;
       lcd.clear();
       lcd.setCursor(1,0);
       lcd.print("Access Refuse");
       lcd.setCursor(0,1);
       lcd.print("UID: ");

       for (byte i = 0; i < 4; i++){
         lcd.setCursor((3*i+5),1);
         lcd.print(code[i],HEX);
       }
       delay(2000);
       lcd.clear();
     }
     rfid.PICC_HaltA(); // halt PICC
     rfid.PCD_StopCrypto1(); // stop encryption on PCD
   }
  }
}
//.....................................................................................
//.....................................Fonction d'ouverture barrière OUT.....................
//.....................................................................................
void open_barriere(){

   for (int position = 82; position >= 45; position--) {    // ouverture de la barrière Angle
90-45
      bar_OUT.write(position);
      delay(45);
   }
    delay(5000);
```

```
    for (int position = 45; position <= 82; position++) {
        bar_OUT.write(position);
        delay(45);
     }
   delay(100);
  if(analogRead(place1)<920){
    lcd.setCursor(2,1);
    lcd.print("1");
  }
  if(analogRead(place2)<900){
    lcd.setCursor(3,1);
    lcd.print("2");
   }
  if(digitalRead(place3)<800){
    lcd.setCursor(4,1);                    // actualiser l'affichage, on libère la place approprie
    lcd.print("3");
   }
  if(digitalRead(place4)<850){
    lcd.setCursor(5,1);
    lcd.print("4");
   }
}
//.....................................................................................
//.............................Fonction signale d'alarme.....................................
//.....................................................................................
void Alarme_temp2(){
  for(int i=700;i<800;i++){
    tone(Buzzer,i);  //i=la fréquence du signal sonore
    delay(15);
  }
  for(int i=800;i>700;i--){
    tone(Buzzer,i);
```

```
    delay(15);
  }
}
//.....
//.....Fonction lecturede la températuredе.....
//.....
void DHT11_temp(){
    h = dht.readHumidity();
    temp_cel = dht.readTemperature();
  }
//.....
//.....Fonction du programme principal.....;.....
//.....
void loop() {
  lcd.clear();
  while(digitalRead(SW)==LOW){  // switch ON/OFF
        digitalWrite(Acces,LOW);
        DHT11_temp();
        lcd.setCursor(0,0);
        lcd.print("| Libre | Temp |");
        lcd.setCursor(0,1);
        lcd.print("|");
        lcd.setCursor(8,1);
        lcd.print("|");
        lcd.setCursor(10,1);
        lcd.print(temp_cel);
        lcd.print((char)223);
        lcd.print("C");
        lcd.setCursor(15,1);
        lcd.print("|");

        while(temp_cel>35 ){
```

```
            bar_OUT.write(45);
            bar_IN.write(100);
            Alarme_temp2();

            DHT11_temp();
            lcd.setCursor(10,1);
            lcd.print(temp_cel);
            lcd.print((char)223);
            lcd.print("C");
            lcd.setCursor(15,1);
            lcd.print("|");
            lcd.setCursor(0,0);
            lcd.print(" Alerte incendie");
          }
          noTone(Buzzer);
        etat_places();

        if(digitalRead(cpt_OUT)==HIGH){
            open_barriere();
        }
        if(Parking_Complet==false){
          if(temp_cel<=34 )  bar_OUT.write(82);    //if(temp_cel<=28 )
          if(digitalRead(cpt_IN)==LOW || digitalRead(cpt_OUT)==HIGH ){     // test s'il y
a présence de véhicule
              if(digitalRead(cpt_IN)==LOW){        //test l'acces de véhicule
                  lire_rfid();
                  if(ouvre_barriere==true){
                      ouvre_barriere=false;
                      for (int position = 100; position >= 45; position--) {    // ouverture de la
barrière Angle 100-40
                        bar_IN.write(position);
                        delay(45);
```

```
                    }
                     delay(5000);
                     for (int position = 45; position <= 100; position++) {
                       bar_IN.write(position);
                       delay(45);
                     }
                   lcd.clear();
                  }
                 }
               else if(digitalRead(cpt_OUT)==HIGH){
                    open_barriere();
                }
               presence_vehicule=true;
            }
      }
}
```