

See discussions, stats, and author profiles for this publication at: <https://www.researchgate.net/publication/353484098>

# Projet Tutoré PARKING INTELLIGENT

Article · July 2021

CITATIONS

0

READS

46,345

2 authors:



[Hassan Hammioui](#)

Sidi Mohamed Ben Abdellah University

5 PUBLICATIONS 0 CITATIONS

[SEE PROFILE](#)



[Ali Frakcha](#)

2 PUBLICATIONS 0 CITATIONS

[SEE PROFILE](#)

Filière : SMP

## Projet Tutoré

### Semestre S<sub>6</sub>

## *Mémoire*

# PARKING INTELLIGENT

Présenté par :

- Hassan Hammioui
- Ali Frakcha

Encadrant : Pr. Noredine Chaibi

Soutenu le : 10/06/2021

Jury : Examineur 1 : Pr. Adil Essalmi

Examineur 2 : Pr. Bensalem Boukili

Encadrant : Pr. Noredine Chaibi

# Remerciements

Il n'y a pas de travail intéressant, sauf un soutien moral et financier pour les personnes proches de nous :

Nous voudrions exprimer notre profonde gratitude pour ce qui suit :

- Nos parents et mères qui nous entourent toujours et nous motivent à devenir de mieux en mieux

Nos frères et sœurs : Qui nous a aidés en ces temps difficiles et nous a servi.

Nos amis et amis du monde entier qui nous encouragent tout le temps

- Tous nos enseignants ont leur disponibilité et leurs conseils, en particulier le maître :
- Dr. NOREDDINE CHAIBI pour nous aider et son précieux intérêt
- Aux membres du jury d'avoir accepté de juger ce travail :
  - ADIL ESSALMI
  - BENSALAM BOUKIL

Merci et merci à Dieu qui m'a donné la force dans les moments de difficulté pour éditer ces mémos.

Je trouve ici une expression de ma plus profonde gratitude et gratitude

# Table des matières

Remerciements.....	1
INTRODUCTION GÉNÉRALE.....	5
Chapitre I : .....	6
Matériaux et composants.....	6
1 La carte de Arduino .....	7
1.1 Les types de la carte .....	7
1.2 Composantes de la carte Arduino UNO : .....	11
2 Servo-moteur .....	12
2.1 Les types de servo-moteur.....	12
2.2 Composition d'un servo-moteur .....	13
2.3 Principe de fonctionnement d'un servo-moteur .....	14
3 Le module LED.....	15
4 Cristal liquide d'affichage (LCD) .....	16
4.1 Description de l'écran LCD .....	16
4.2 Les points de connexion de l'écran à cristaux .....	16
4.3 Écran à cristaux liquides 16 * 2 avec système de connexion I2C .....	17
5 Capteur Infrarouge .....	17
5.1 Définition.....	17
5.2 Principe de fonctionnement .....	18
5.3 Caractéristiques Techniques .....	19
Chapitre II : .....	20
Simulation .....	20
1 Simulation des different circuits .....	21
1.1 Simulateur Isis Proteus .....	21
1.2 Interface utilisateur.....	22
1.3.1 Définition .....	22
2 Schéma de parking intelligent.....	24
2.1 Premier pas : utiliser un servomoteur .....	24
2.2 Deuxième pas : utiliser un capteur IR SENSOR.....	26
2.3 Troisième pas : simulation du parking intelligent.....	27

<b>Chapitre III :</b> .....	<b>31</b>
<b>Réalisation</b> .....	<b>31</b>
<b>3   Réalisation du parking intelligent.</b> .....	<b>32</b>
<b>3.1    Parking intelligent sur Isis Proteus</b> .....	<b>32</b>
<b>3.2    Réalisation de parking intelligent</b> .....	<b>32</b>
<b>Conclusion générale</b> .....	<b>34</b>
<b>Bibliographie</b> .....	<b>35</b>

# Table des Figures

<b>Figure 1:</b> Carte Arduino LEONARDO .....	8
<b>Figure 2:</b> Carte Arduino Mega .....	9
<b>Figure 3:</b> Carte Arduino DUE .....	9
<b>Figure 4:</b> Carte Arduino Nano.....	10
<b>Figure 5:</b> Carte Arduino Mini Pro.....	10
<b>Figure 6:</b> Carte Arduino YUN.....	11
<b>Figure 7 :</b> Composants et caractéristiques L'Arduino UNO.....	12
<b>Figure 8:</b> types de servomoteur .....	13
<b>Figure 9:</b> Vue interne d'un servomoteur .....	14
<b>Figure 10:</b> simulation d'un pont en H réaliser par Proteus à base des transistors.....	15
<b>Figure 11:</b> Module LED RGB .....	15
<b>Figure 12:</b> Ecran LCD AVEC 16*2 .....	16
<b>Figure 13:</b> Ecran LCD avec système de connexion I2C .....	17
<b>Figure 14:</b> Capteur infrarouge .....	18
<b>Figure 15:</b> les ondes infrarouges .....	18
<b>Figure 16:</b> Simulation Isis Proteus .....	21
<b>Figure 17:</b> Interface utilisation .....	22
<b>Figure 18:</b> Langage Arduino .....	22
<b>Figure 19:</b> Interface d'utilisateur .....	23
<b>Figure 20:</b> schéma électrique du parking intelligent .....	24
<b>Figure 21:</b> montage reliant un servo-moteur et un Arduino.....	25
<b>Figure 22:</b> montage reliant un <b>capteur IR SENSOR</b> et un Arduino.....	26
<b>Figure 23:</b> montage reliant les défèrent composantes du parking intelligent.....	28
<b>Figure 24:</b> Parking intelligent sur Isis Proteus .....	32
<b>Figure 25:</b> image de réalisation du parking intelligent.....	33

# INTRODUCTION GÉNÉRALE

Aujourd'hui, beaucoup d'états comme la Chine et les États-Unis investissent des milliards de dollars pour le convoi du développement exponentiel des machines intelligentes comme dans le domaine militaire où on entend parler des drones qui s'avèrent être très efficaces, ou dans le domaine industriel où on utilise de nos jours les robots de soudure et de peinture notamment dans l'automobile. Arrivé à ce stade de développement, beaucoup de gens commencent de nos jours à s'inquiéter de ces robots performants, rapides et infatigables et se demandent : Sommes-nous donc tous voués à être remplacés et finir sur la paille ? Et pour demain, quelle sera la place de l'humain ?

Des questions qui nous mènent à revoir les enjeux et les risques qui accompagnent ces machines intelligentes et savoir si ces robots vont constituer vraiment l'avenir de l'être humain ou sa fin dans cette planète.

Le but de ce projet est l'initiation et la mise en œuvre un module ARDUINO. Les cartes Arduino sont conçues pour réaliser des prototypes et des maquettes de carte électronique pour l'informatique embarquée. Elles peuvent être utiliser dans des applications comme par exemple un Alarme détectrice de la chaleur, ou une porte automatique.

D'après l'arrivant de la technologie est aidé dans le développement des parkings et transférer tous qui manuelles vers l'automatique, tout ça est vérifier grâce à des composantes électroniques de fabrication bien organise contacter avec l'élément principale qui s'appelle Arduino.

Les cartes Arduino font partie de la famille des microcontrôleurs. Ce dernier est une unité de calcul accompagnée de mémoire, de ports d'entrée/ sortie et de périphériques permettant d'interagir avec son environnement. L'Arduino permet d'exploiter le microcontrôleur, pour le rendre plus facile à programmer par des ingénieurs, des chercheurs et les développeurs.

Notre projet consiste à concevoir d'un parking intelligent, constitué d'une porte d'entrée et une autre de sortie gérée automatiquement par une carte ARDUINO UNO.

# **Chapitre I :**

# **Matériaux et composants**



## Introduction

Le choix des matériaux est une tâche fondamentale et très complexe, et pour réussir à bien l'aborder, on a posé un groupe de critères qu'on devait les satisfaire pour bien réussir notre choix. Les critères imposés sont les suivants :

- Au niveau de matériau : caractéristiques mécaniques, esthétiques, thermiques, électriques, économiques, environnementales, physiques.
- Au niveau du procédé : volume, masse, géométrie, taille de la série, caractéristiques économiques et environnementales.

## 1 La carte de Arduino

Une carte Arduino est une petite carte électronique équipée d'un microcontrôleur. Le microcontrôleur permet, à partir d'événements détectés par des capteurs, de programmer et commander des actionneurs ; la carte Arduino est donc une interface programmable. Elle peut être programmée pour analyser et produire des signaux électriques, de manière à effectuer des tâches très diverses comme le contrôle des appareils domestiques, éclairage, chauffage, le pilotage d'un robot, de l'informatique embarquée, ect

### 1.1 Les types de la carte

#### ✓ **La carte Arduino UNO :**

C'est la carte idéale pour découvrir l'environnement ARDUINO. Elle permet à tout débutant de se lancer dans tous ses premiers petits projets. Comme c'est la carte la plus utilisée, il est très facile de se référer aux tutoriels très nombreux sur le net et ainsi de ne pas rester seul dans son exploration.

Sa simplicité devient par contre un handicap lorsqu'il s'agit de multiplier les périphériques, de manipuler des algorithmes lourds ou d'interagir avec les OS / Android pour lesquels d'autres cartes Arduino sont plus adaptées.

#### ✓ **La carte Arduino LEONARDO :**

C'est la carte qui est prévue pour succéder à la carte Arduino UNO en présentant des caractéristiques équivalentes mais une ergonomie revue et une stabilité plus éprouvée. Sa diffusion moins importante limite le support utilisateur disponible sur le net.



**Figure 1:** Carte Arduino LEONARDO

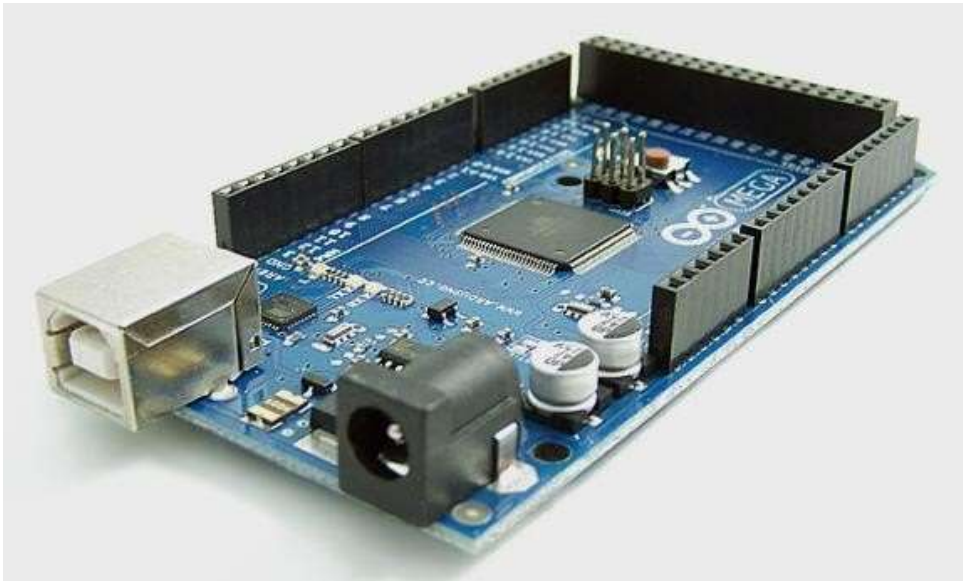
✓ **La carte Arduino Mega :**

La carte Arduino Mega est la carte la plus diffusée après la carte Arduino UNO. Elle offre un nombre d'entrées/sorties beaucoup plus important (54 contre 14), un processeur plus puissant doté d'une mémoire plus vaste qui permet d'exploiter des algorithmes plus complexes.

✓ **La carte Arduino Mega ADK :**

La carte Arduino Mega ADK offre les mêmes caractéristiques techniques que la carte Arduino Mega mais son port USB permet de la connecter avec un environnement androïde ouvrant de nouvelles perspectives d'interaction avec le monde des Smartphones et des capteurs dont ils sont dotés. Sa mise en œuvre nécessite par contre

de solides connaissances en Java et la capacité à développer ses propres applications.



**Figure 2:** Carte Arduino Mega

✓ **La carte Arduino Due :**

La carte Arduino Due est une évolution de la carte Arduino Mega et offre des performances réputées 3 fois supérieures. Elle permet de manipuler rapidement des algorithmes lourds particulièrement utiles dans le monde de la robotique par exemple.



**Figure 3:** Carte Arduino DUE

✓ **La carte Arduino Nano :**

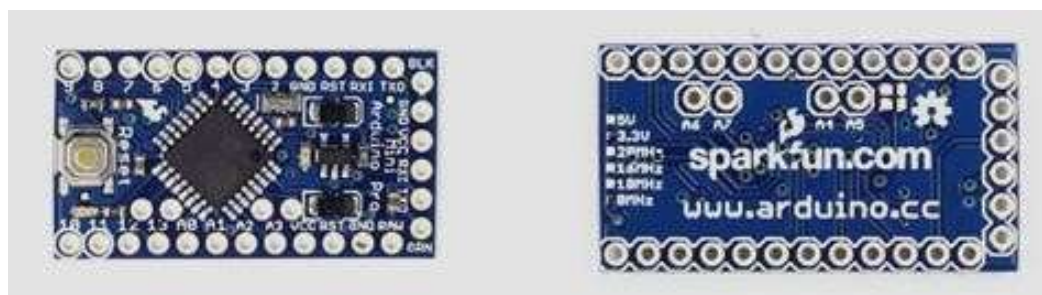
La carte Arduino nano n'est ni plus ni moins qu'une carte Arduino UNO miniaturisée. Sa taille et son poids réduits la destinent à une utilisation dans des espaces réduits (en textile par exemple) ou dans des applications de robotique ou de modélisme pour lesquels le poids et la taille sont des facteurs déterminant (hélicoptères, drones...).



**Figure 4:** Carte Arduino Nano

✓ **La carte Arduino Mini Pro :**

La carte Arduino Mini Pro est une carte Arduino UNO simplifiée à l'extrême permettant néanmoins de piloter de petits projets ou certains éléments d'un projet. Attention, cette carte n'intègre pas de port USB ce qui rends sa connectivité délicate.



**Figure 5:** Carte Arduino Mini Pro

✓ **La carte Arduino Yun :**

La carte Arduino Yun, récemment proposée par Arduino, est conçue pour contrer les avantages de la carte Raspberry. Elle est un dérivé de la carte Leonardo et a pour objectif de combiner la puissance de Linux avec la facilité d'utilisation d'une carte Arduino. Elle est également la première carte Arduino à être dotée nativement d'un wifi intégré.

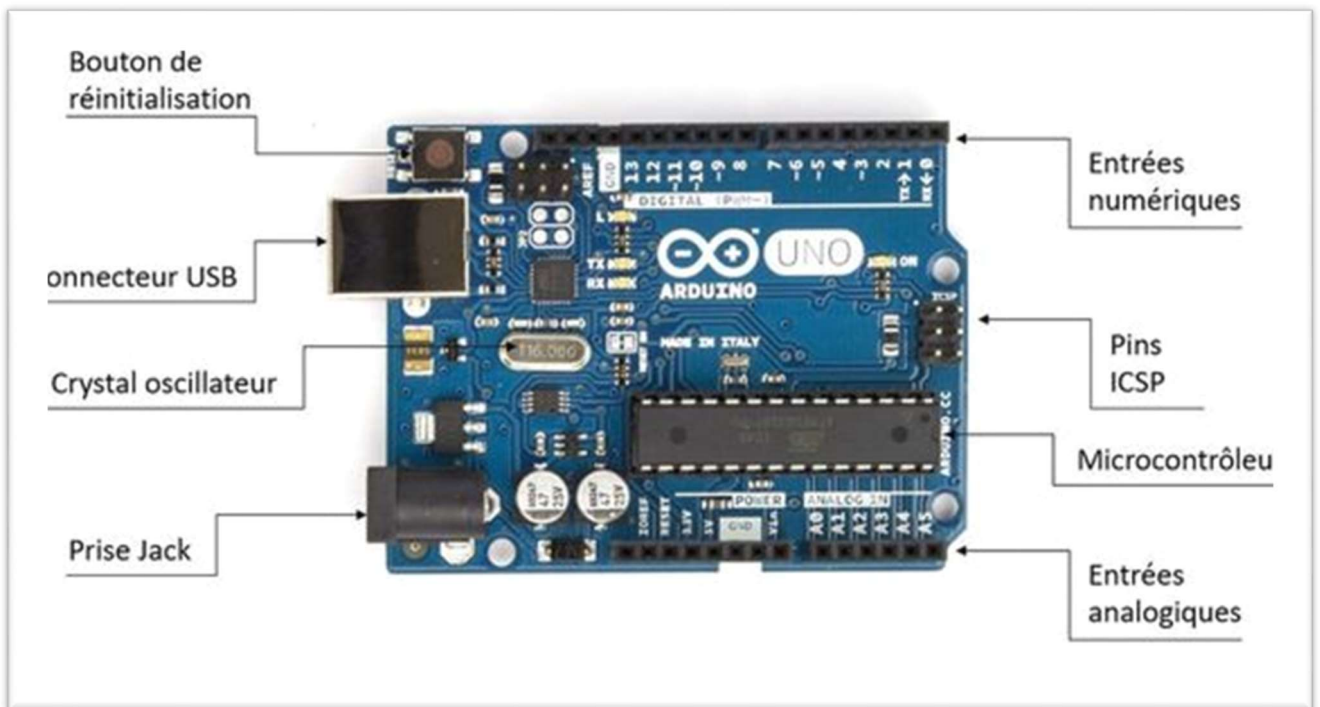


**Figure 6:** Carte Arduino YUN

## 1.2 Composantes de la carte Arduino UNO :

Une carte Arduino UNO se compose essentiellement de :

- Prise jack : permet de brancher une alimentation (pile, batterie, adaptateur Secteur, + au centre Vin 7 à12 V
- Microcontrôleur : stocke le programme et l'exécute.
- Entrées et analogique : Permet de brancher des capteurs et des détecteur analogiques.
- Connexion USB (Universal Serial Bus) : Permet d'alimenter la carte en Énergie électrique (5V). Permet de téléverser le programme dans la carte.
- Entrées et sorties numériques (Digital) : Permet de brancher des actionneurs. Permet de brancher des détecteurs.



**Figure 7 :** Composants et caractéristiques L'Arduino UNO

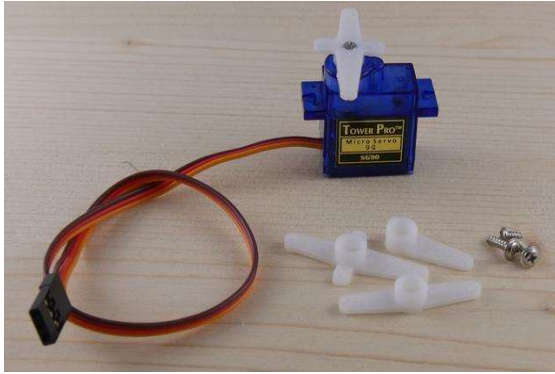
## 2 Servo-moteur .

Un servo-moteur est un type de moteur électrique. C'est un dispositif typiquement utilisé en modélisme pour, par exemple, contrôler la direction d'une voiture télécommandée. Sur un servo-moteur, l'angle de l'axe reste fixé dans une position et peu varier entre 0 et 180° en fonction du signal envoyé

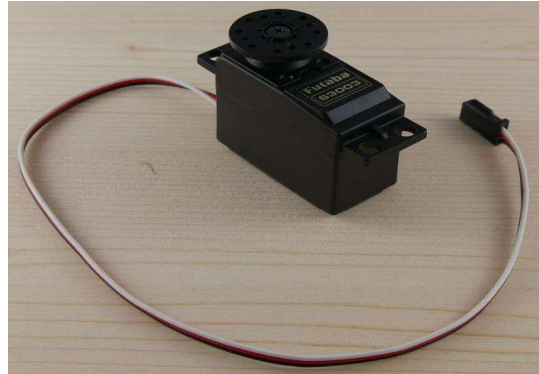
### 2.1 Les types de servo-moteur

Il existe divers types de servomoteur, de taille, poids et couple (force) différents. la photographie ci-dessous présente un servomoteur très classique en modélisme : le futaba S3003. Un peu plus bas l'article, on utilisera un autre servomoteur, communément appelé "servomoteur 9 grammes", par souci de consommation électrique





**a-** Servomoteur "9 grammes"



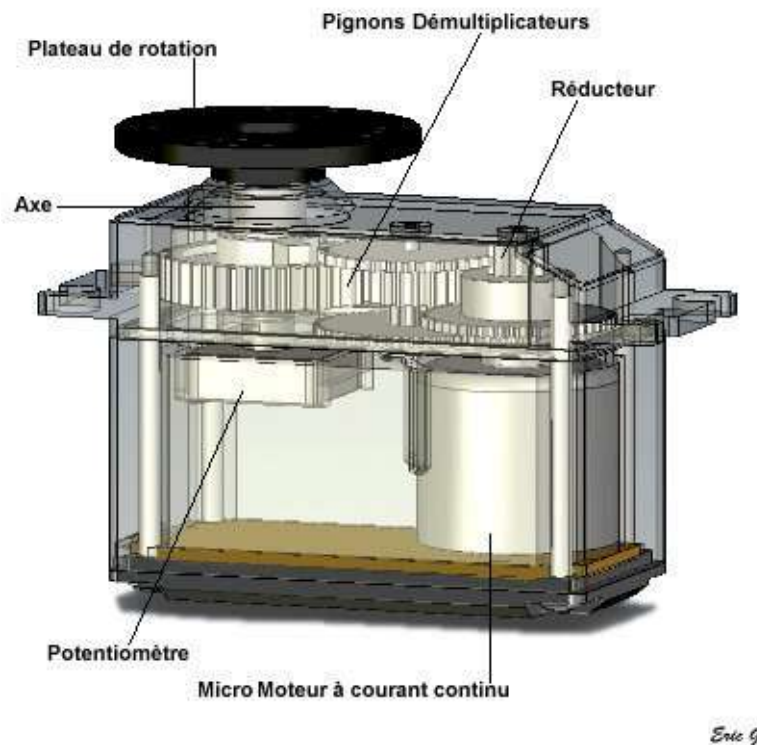
**b -** .Servomoteur Futaba S3003

**Figure 8:** types de servomoteur

## 2.2 Composition d'un servo-moteur

Le servomoteur est composé de plusieurs éléments visibles et invisible :

- Un moteur à courant continu
- Des engrenages pour former un réducteur (en plastique ou en métal)
- Un capteur de position de l'angle d'orientation de l'axe (un potentiomètre)
- Une carte électronique pour le contrôle de la position de l'axe et le pilotage du moteur à courant continu
- Les fils, qui sont au nombre de trois
- L'axe de rotation sur lequel est monté un accessoire en plastique ou en métal
- Le boîtier qui le protège

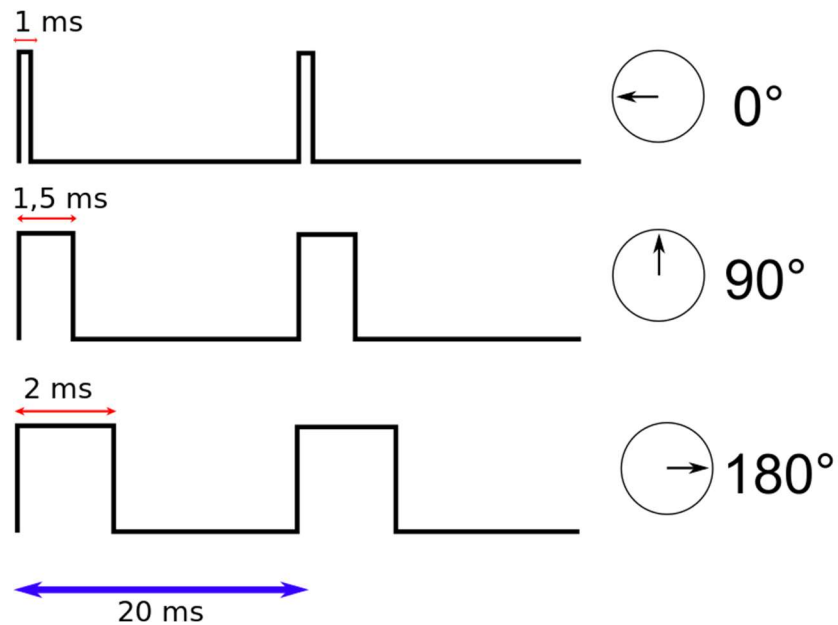


**Figure 9:** Vue interne d'un servomoteur

## 2.3 Principe de fonctionnement d'un servo-moteur

La plupart des servomoteurs sont commandés par l'intermédiaire d'un câble électrique à trois fils qui permet d'alimenter le moteur et de lui transmettre des consignes de position sous forme d'un signal codé en largeur d'impulsion plus communément appelé PWM. Cela signifie que c'est la durée des impulsions qui détermine l'angle absolu de l'axe de sortie et donc la position du bras de commande du servomoteur. Le signal est répété périodiquement, en général toutes les 20 millisecondes, ce qui permet à l'électronique de contrôler et de corriger continuellement la position angulaire de l'axe de sortie, cette dernière étant mesurée par le potentiomètre.





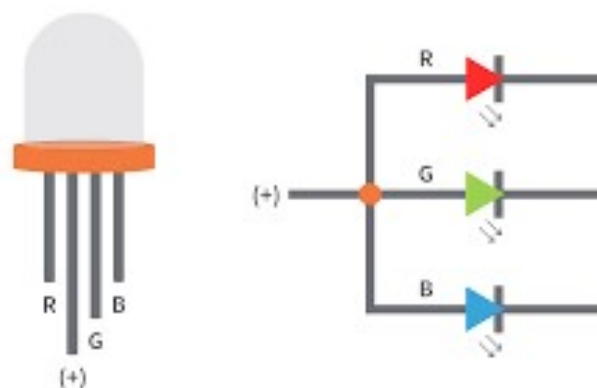
**Figure 10:** simulation d'un pont en H réaliser par Proteus à base des transistors

### 3 Le module LED.

Une LED RGB est une LED qui peut s'allumer dans toutes les couleurs en fonction de trois couleurs : le rouge le vert et le bleu.

La LED RGB a 4 broches : une commune à l'ensemble des LEDS et une pour chaque couleur de la LED.

La broche commune pourra, selon les modèles, être le+(anode commune) ou le (cathode commune). Ici la LED RGB a pour broche commune la cathode (borne-). Certaines LED RGB intègrent déjà les résistances de limitation de courant dans des modules.



**Figure 11:**Module LED RGB

## 4 Cristal liquide d'affichage (LCD)

### 4.1 Description de l'écran LCD

Ce type d'écran est le moins et le plus commun parmi les passionnés d'électronique et les projets électroniques, car ils sont disponibles à bas prix et faciles à programmer, et ce type est disponible en différentes tailles et couleurs, avec 16x2 et les moins entre eux.

20 × 2, 20 × 4 et autres.

Nous entendons par 16 × 2, c'est-à-dire, il y a deux lignes et chaque ligne peut être écrite 16 caractères

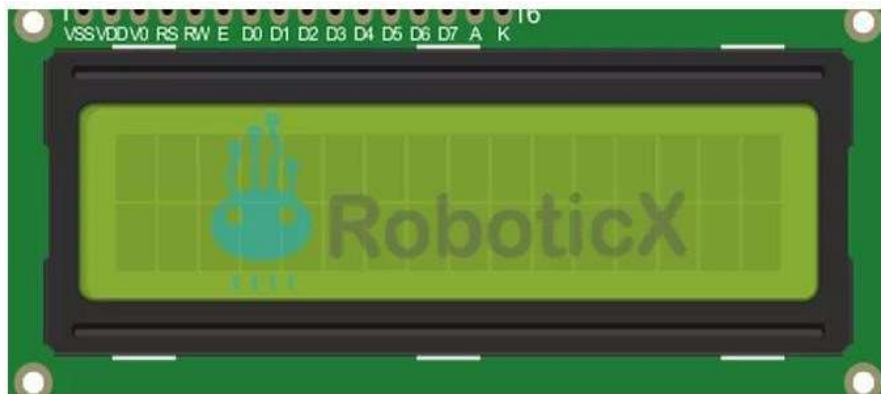


Figure 12: Ecran LCD AVEC 16\*2

### 4.2 Les points de connexion de l'écran à cristaux

Les points de connexion de l'écran à cristaux Sélection de l'enregistreur

1-RS : Ce port nous détermine dans quelle mémoire nous écrivons des données !

L'enregistreur de données ou l'enregistreur d'instructions est sélectionné Option

2-R / W pour lire ou écrire, ce port spécifie soit sélectionner le mode lecture ou écriture.

3-enable pin Ce port, s'il est activé, vous permet d'écrire sur l'écran LCD broche de données

4- (D0-D7) : ce sont les ports par lesquels nous transmettons des données aux enregistreurs.

### 4.3 Écran à cristaux liquides 16 \* 2 avec système de connexion I2C

Il est caractérisé par l'utilisation du système I2C opération et contrôle à travers deux fils seulement, ce qui est une énorme provision pour les prises du panneau de contrôle utilisé dans notre projet et facilite le processus de connexion et de fonctionnement. (htt4)



**Figure 13:** Ecran LCD avec système de connexion I2C

## 5 Capteur Infrarouge

### 5.1 Définition

Le capteur d'évitement d'obstacles à **capteur infrarouge IR** est une solution peu coûteuse pour la détection d'évitement pour la robotique, la voiture intelligente et d'autres utilisations électroniques. Dans cette leçon, nous allons vous montrer comment fonctionne le capteur d'évitement d'obstacles et comment l'utiliser avec la carte ARDUINO UNO.

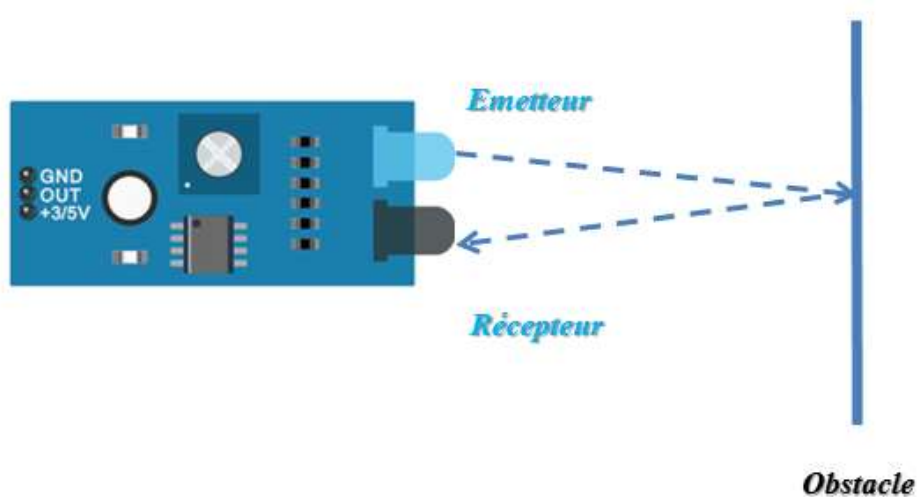


**Figure 14:** Capteur infrarouge

Le capteur infrarouge ou le capteur IR a deux parties principales. Émetteur IR et récepteur IR. La responsabilité de l'émetteur IR ou de l'émetteur infrarouge est la transmission des ondes infrarouges tandis que le travail du récepteur IR est de recevoir ces ondes infrarouges. Le récepteur IR envoie constamment des données numériques sous la forme de 0 (LOW) ou 1 (HIGH) à la broche V out du capteur.

## 5.2 Principe de fonctionnement

L'émetteur infrarouge envoie un signal infrarouge qui, en cas de surface réfléchissante (par exemple de couleur blanche), rebondit dans certaines directions, y compris celle du récepteur infrarouge qui capte le signal détectant l'objet



**Figure 15:** les ondes infrarouges

Lorsque la surface est absorbante (par exemple de couleur noire), le signal IR n'est pas réfléchi et l'objet ne peut pas être détecté par le capteur. Ce résultat se produirait même si l'objet est absent.

### 5.3 Caractéristiques Techniques

- Voltage de fonctionnement : DC 3,3 V-5 V
- Intensité du courant  $\geq 20\text{mA}$
- Température de fonctionnement :  $-10\text{ }^{\circ}\text{C} \rightarrow +50\text{ }^{\circ}\text{C}$
- Plage de distance : 2-40 cm
- Interface IO : interface 4 fils (GND / Vcc / OUT / EN)
- Signal de sortie : tension TTL
- Mode d'hébergement : régulation de résistance multi-cercle
- Angle effectif :  $35^{\circ}$
- Taille :  $41,7 * 16,7\text{ mm}$
- Poids : 5g

#### Conclusion

Le choix des matériaux était la phase la plus délicate dans notre projet. En effet, il ne portait pas seulement sur un aspect purement technique répondant à des exigences fonctionnelles, mais aussi à des attentes relevant des préférences des utilisateurs dans le cadre d'un marché spécifique. En gros, le choix des matériaux a été analysé sous l'angle de l'ingénierie des matériaux, du temps de réponse, sans oublier enfin le côté esthétique (le design industriel

# **Chapitre II : Simulation**

## Introduction

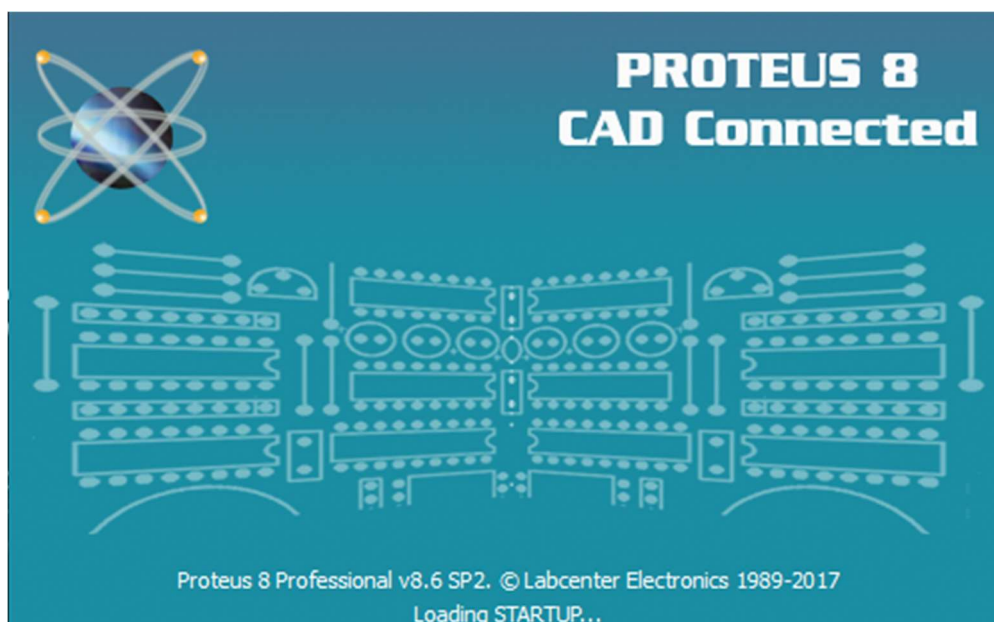
Notre simulation de parking intelligent était avant l'avant-dernière tâche que nous devions faire, les études étaient déjà terminées et les matériaux étaient déjà en place, il ne manquait plus qu'une structure pour accueillir ces appareils puis vérifier si les mesures que nous avons faites concilieraient les côtés techniques avec le côté esthétique.

## 1 Simulation des different circuits

### 1.1 Simulateur Isis Proteus

Isis Proteus est une suite logicielle destinée à l'électronique. Développé par la société Labcenter Electronics, elle est principalement connue pour éditer des schémas électriques. Par ailleurs le logiciel permet également de simuler ces schémas.

Les circuits électriques conçus grâce à ce logiciel peuvent être utilisé dans des documentations car le logiciel permet de contrôler la majorité de l'aspect graphique des circuits.



**Figure 16:** Simulation Isis Proteus

## 1.2 Interface utilisateur

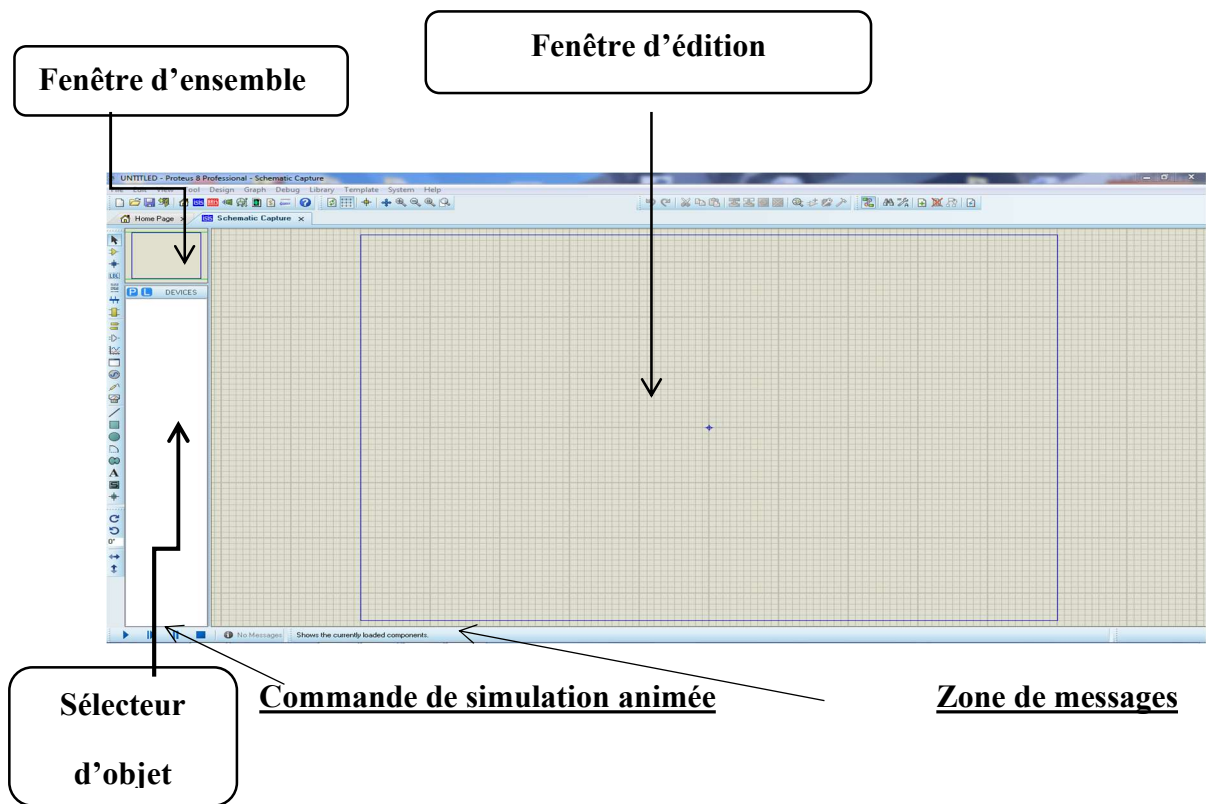


Figure 17: Interface utilisation

## 1.3 Langage Arduino

### 1.3.1 Définition

Le langage Arduino est inspiré de plusieurs langages. On retrouve notamment des similarités avec le C, le C++.

Le langage impose une structure particulière typique de l'informatique embarquée. La programmation se fait de manière séquentielle c'est-à-dire une suite d'instruction élémentaires

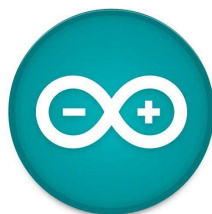
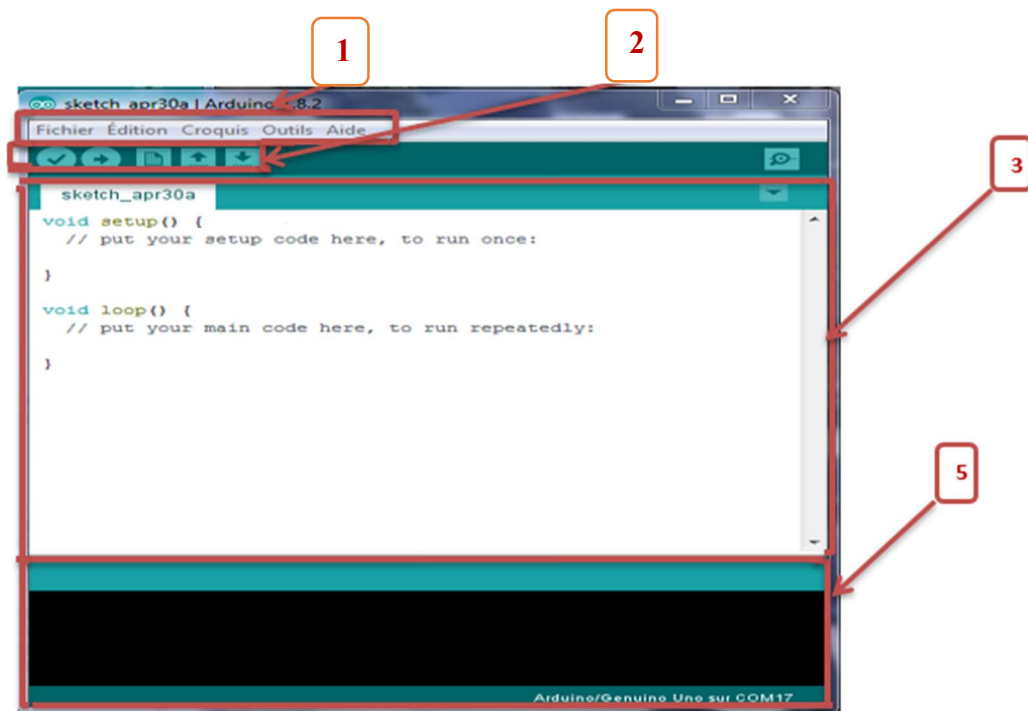


Figure 18: Langage Arduino



### 1.3.2 Interface d'utilisateur



**Figure 19:** Interface d'utilisateur

**Le cadre numéro 1 :** ce sont les options de configuration du logiciel

**Le cadre numéro 2 :** il contient les boutons qui vont nous servir lorsque l'on va programmer nos cartes

**Le cadre numéro 3 :** ce bloc va contenir le programme que nous allons créer

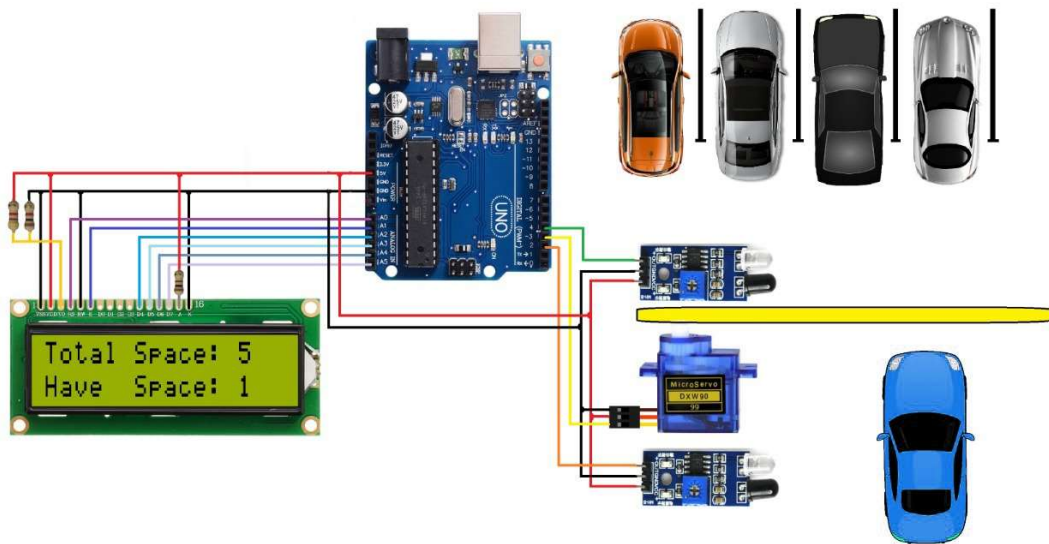
**Le cadre numéro 4 :** celui-ci est important, car il va nous aider à corriger les fautes dans notre programme. C'est le débogueur.

**Un programme (ou "sketch") Arduino est constitué de 2 fonctions distinctes :**

- La fonction de configuration "**void setup**" exécutée une seule fois au lancement du Programme. On appelle aussi cette fonction : **fonction d'initialisation**.
- La fonction "**void loop**" qui est ensuite exécutée indéfiniment en boucle

## 2 Schéma de parking intelligent

Voici le schéma de notre robot, on a abouti ce schéma en utilisant le logiciel Fitting avec l'aide de Photoshop.



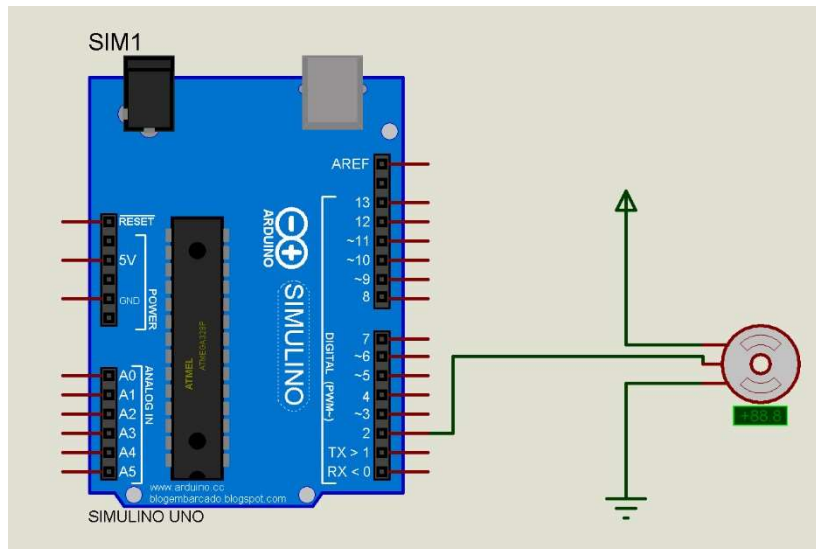
**Figure 20:** schéma électrique du parking intelligent

### 2.1 Premier pas : utiliser un servomoteur

Afin de bien maîtriser la programmation et l'utilisation de la carte Arduino, On va tout d'abord commencer par des programmes simples dont le but est de réaliser un programme optimal pour notre parking intelligent.

Commençons au début par un programme assez simple, dans lequel on cherche à sert à Faire tourner le bras du servo moteur

Le schéma réaliser sur ISIS



**Figure 21:** montage reliant un servo-moteur et un Arduino

### Le code Arduino utilisé :

```
//mettre l'angle du bras de votre servomoteur à 90°
#include <Servo.h>
Servo monServo; // création d'un nouveau servomoteur
void setup()
{
  monServo.attach(2, 1000, 2000);
  monServo.write(90);
}
void loop()
{
}
```

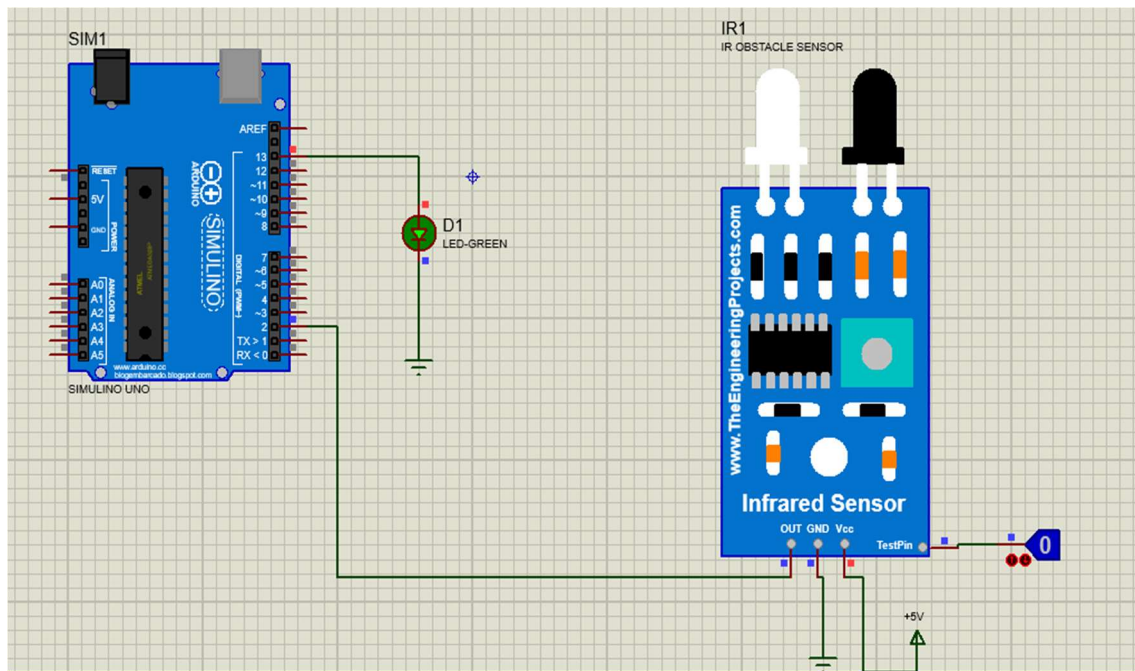
### Commentaires :

- On a utilisé la fonction `monServo.attach(2, 1000, 2000)` pour Attacher le servo moteur au "pin" 2 et fixer l'angle minimal et maximal du tournage respectivement sur les angles correspondants aux valeurs 1000 et 2000 (la durée de la pulsation enmicroseconds).
- La fonction `monServo.write(90)` sert à Faire tourner le bras du servo moteur pour une angle de 90

## 2.2 Deuxième pas : utiliser un capteur IR SENSOR

Le schéma réaliser sur ISIS

Afin de mettre en évidence le fonctionnement du capteur IR SENSOR, on réalise la simulation suivante sur Proteus, on relie le capteur a la carte Arduino et une LED on va lire les résultats indiqués par le capteur avec un LED



**Figure 22:** montage reliant un **capteur IR SENSOR** et un Arduino

### Le code Arduino utilisé

```
int LED = 13;
int IR = 2;
int IRETAT;

void setup () {
    // put your setup code here, to run once:
    pinMode(LED,OUTPUT);
    pinMode(IR,INPUT);
}

void loop() {
```

```

    // put your main code here, to run repeatedly:
    IRETAT=digitalRead(IR);
    if(IRETAT==0){
        digitalWrite(LED,HIGH);
    }
    else
    {digitalWrite(LED,LOW);
    }
}

```

### Éclaircissements et Commentaires :

L'instruction `const int LED = 13` sert à attacher la constante LED au port 3 de la carte Arduino.

- L'instruction `const int IR` à attacher la constante IR au port 2 de la carte Arduino.
- L'instruction `pinMode(LED , OUTPUT)` sert à fixer le port 13 pour être une sortie.
- L'instruction `pinMode(IR , INPUT)` sert à fixer le port 2 pour être une entrée.
- L'instruction `digitalWrite(LED , LOW)` sert à mettre le port 13 (LED) au niveau bas (0V)
- L'instruction `digitalWrite(LED , HIGH)` sert à mettre le port 13 au niveau haut (5V)

## 2.3 Troisième pas : simulation du parking intelligent

Après avoir étudié séparément et choisi minutieusement les composants contribuant dans la construction de notre parking intelligent, on les a assimilés tous ensemble et voilà ce qu'on a obtenu :

### Schéma électrique du parking intelligent

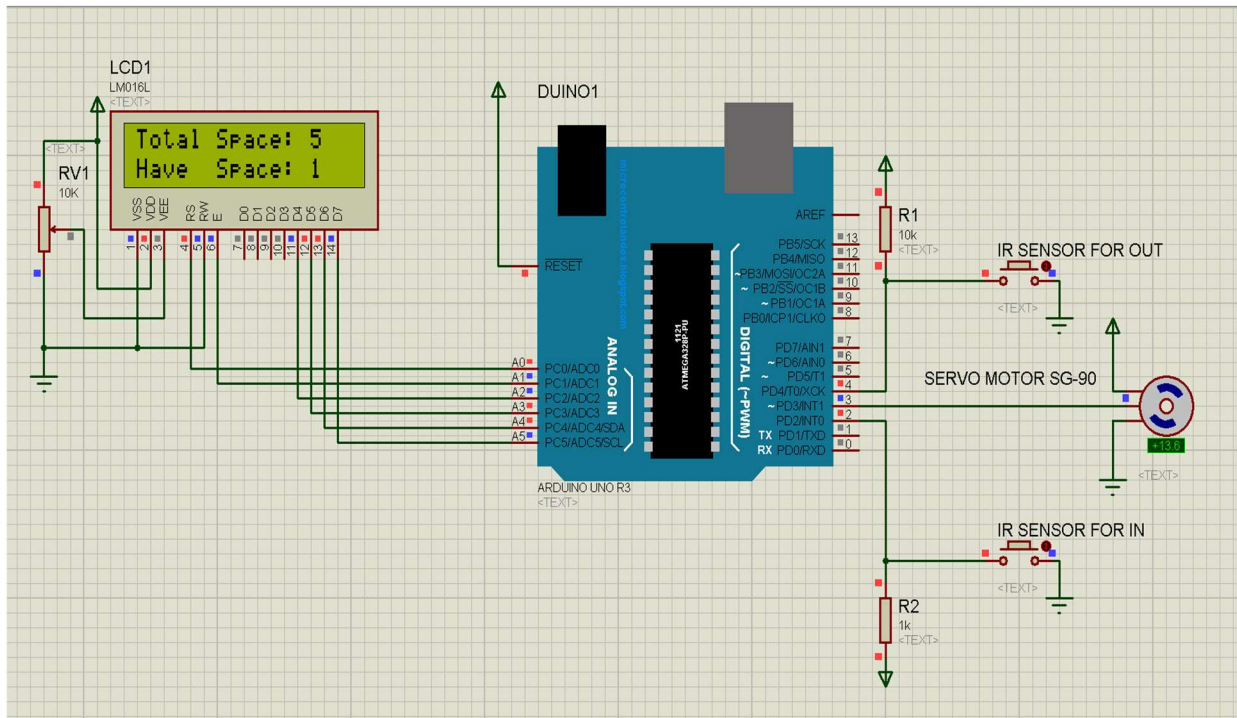


Figure 23: montage reliant les différents composants du parking intelligent

### Le code Arduino utilisé pour programmer le parking intelligent :

```
#include <LiquidCrystal.h> // initialize the library with
the numbers of the interface pins
LiquidCrystal lcd(A0, A1, A2, A3, A4, A5);
#include <Servo.h> //includes the servo library

Servo myservol;

int ir_s1 = 2;
int ir_s2 = 4;

int Total = 5;
int Space;

int flag1 = 0;
int flag2 = 0;

void setup() {
  pinMode(ir_s1, INPUT);
  pinMode(ir_s2, INPUT);
```

```

myservo1.attach(3);
myservo1.write(100);

lcd.begin(16, 2);
lcd.setCursor (0,0);
lcd.print("  Car  Parking  ");
lcd.setCursor (0,1);
lcd.print("      System      ");
delay (2000);
lcd.clear();

Space = Total;
}

void loop(){

if(digitalRead (ir_s1) == LOW && flag1==0){
if(Space>0){flag1=1;
if(flag2==0){myservo1.write(0); Space = Space-1;}
}else{
lcd.setCursor (0,0);
lcd.print(" Sorry not Space ");
lcd.setCursor (0,1);
lcd.print("      Available      ");
delay (1000);
lcd.clear();
}
}

if(digitalRead (ir_s2) == LOW && flag2==0){flag2=1;
if(flag1==0){myservo1.write(0); Space = Space+1;}
}

if(flag1==1 && flag2==1){
delay (1000);
myservo1.write(100);
flag1=0, flag2=0;
}

lcd.setCursor (0,0);
lcd.print("Total Space: ");
lcd.print(Total);

lcd.setCursor (0,1);

```

```
lcd.print("Have Space: ");  
lcd.print(Space);  
}
```

## Conclusion

Dans ce chapitre, nous avons cité en détail les différentes étapes d'un schéma bloc du circuit de Maquette d'un parking avec barrière automatique.

Notre objectif dans cette partie de ce travail était de réaliser tous les composants de notre projet c'est-à-dire le servomoteur avec le lisse, l'écran LCD et les différents capteurs. Ensuite, la connexion et programmation les composants principaux avec le microcontrôleur Arduino ; dans laquelle nous avons tester l'entrée et sorti des véhicules compter et décompter les nombres des véhicules avec afficher le tous les séquences dans l'écran.



# **Chapitre III :**

## **Réalisation**

### 3 Réalisation du parking intelligent.

#### 3.1 Parking intelligent sur Isis Proteus

Avant de réaliser ce projet il suffit de réaliser sur le simulateur le montage suivant :

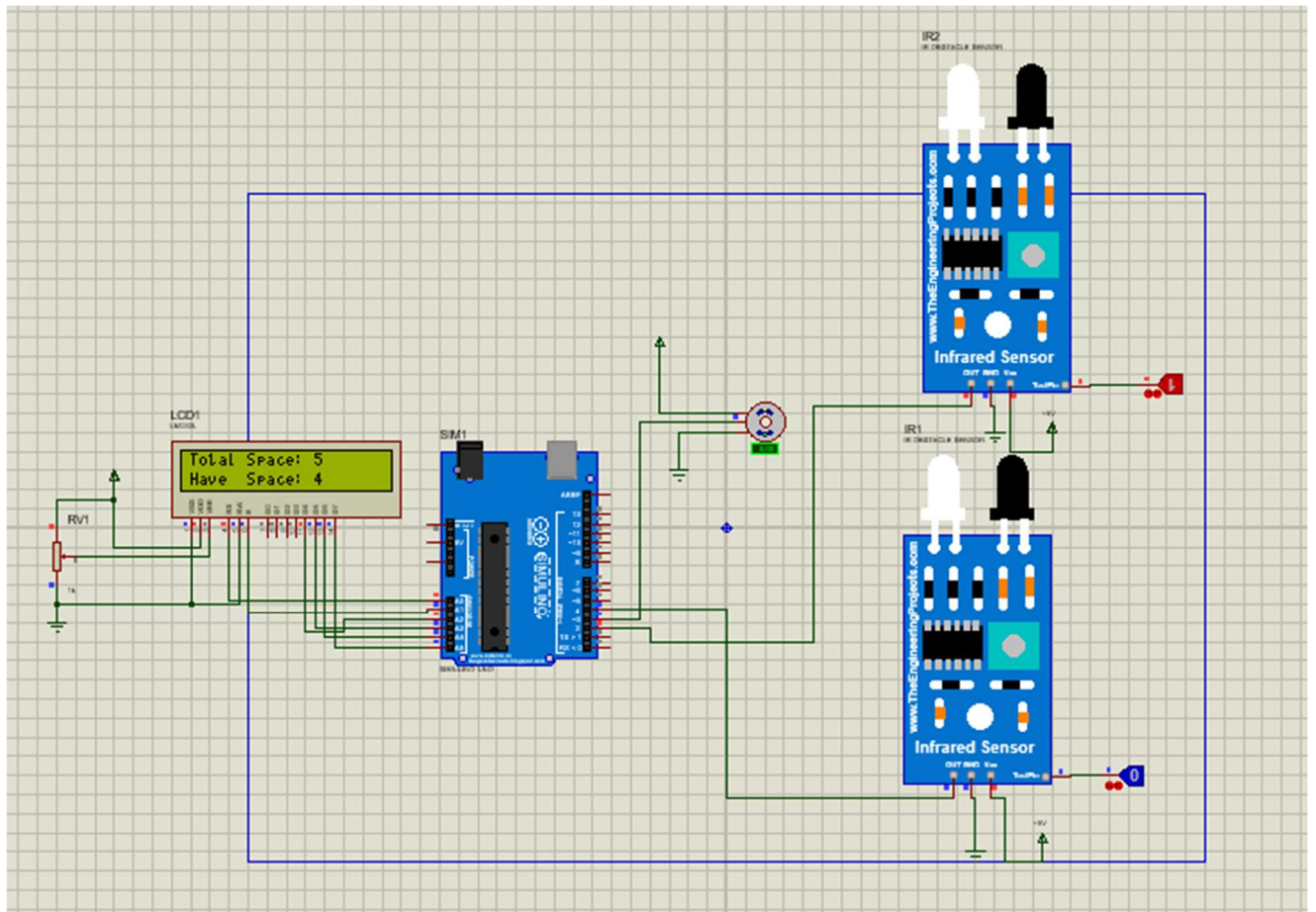
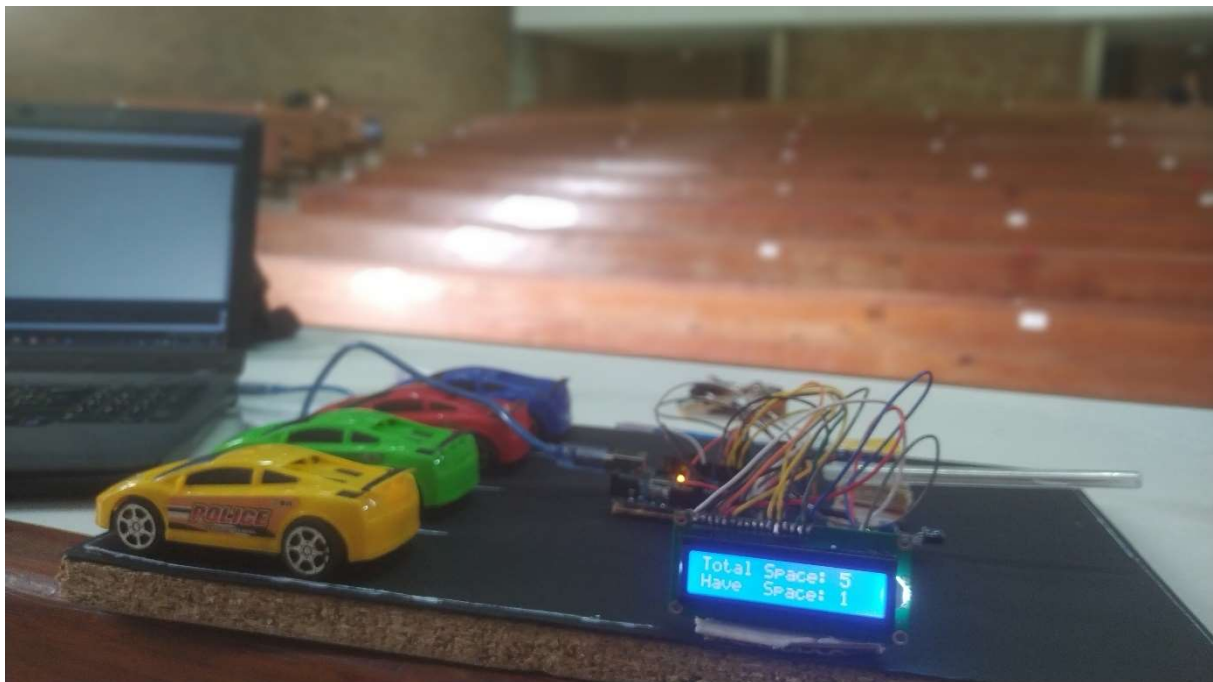
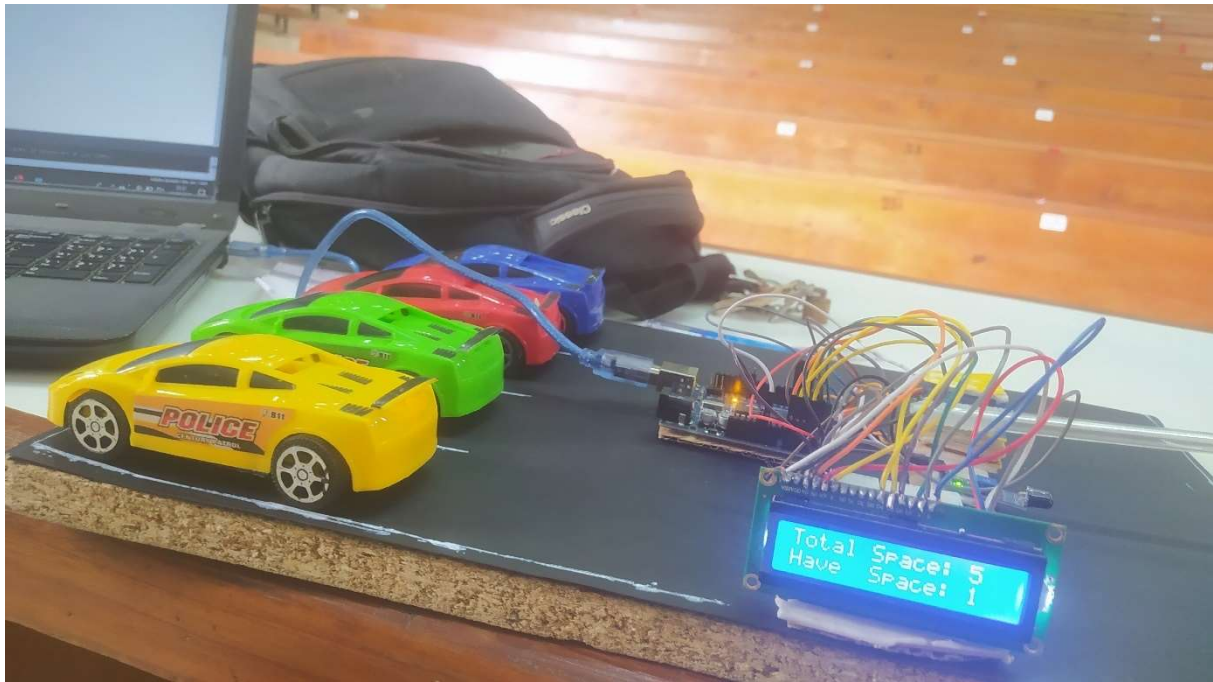


Figure 24: Parking intelligent sur Isis Proteus

#### 3.2 Réalisation de parking intelligent

La réalisation de notre projet qui donne les images suivant :



**Figure 25:** image de réalisation du parking intelligent

## Conclusion

Ce projet de fin d'étude nous a permis de mettre en application les différentes connaissances théoriques acquises durant notre formation en réalité.

# Conclusion générale

Pour répondre aux problèmes de stationnement, le parking intelligent représente une alternative au garage classique, il tire un grand profil de l'espace disponible.

Le système de parking constitue la solution idéale pour automatiser Le processus d'entrée et de sortie.

La réalisation d'une parking intelligent est présentée dans notre travail.

Le premier chapitre : traite des généralités sur l'ARDUINO et les servomoteurs. Ces deux composants sont très importants dans le domaine de la microélectronique et les systèmes embarqués et automatiques. L'ARDUINO et les servomoteurs peuvent être utilisés dans des dizaines d'applications liées à cette technologie.

Pour cette raison, Nous avons représenté les différents types de ces composants ainsi que le mécanisme de fonctionnement de chacun d'eux.

Aux deuxièmes chapitres : Méthode de connexion et programmation des composants principaux avec le Arduino ; dans laquelle nous avons réalisé le système globale (Tester l'entrée et sorti des véhicules, monte et descend la lisse, compter et décompter les nombres des véhicules avec afficher le tous les séquences dans L'écran).

Aux troisièmes chapitre : nous avons réalisé une application de parking intelligent qui utilise ces composants électroniques, sans oublier de donner le câblage et codage utilisés dans cette manipulation.

# Bibliographie

<https://www.wikipedia.org/>

<https://www.arduino.cc/>

<https://fr.wikipedia.org/wiki/Servomoteur>

[https://github.com/embeddedlab786/Car\\_Parking\\_System/blob/master/Car\\_Parking\\_System.ino](https://github.com/embeddedlab786/Car_Parking_System/blob/master/Car_Parking_System.ino)

Manuel d'utilisation de l'afficheur I2C LCD 16x2

<https://www.robotique.tech/tutoriel/definition-de-led-rgb/>

<https://zineddine-academie.blogspot.com/2020/04/capteur-infrarouge-avec-carte-arduino.html>