الجمهورية الجزائرية الديمقراطية الشعبية
République Algérienne Démocratique et Populaire
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique
Université Frères Mentouri Constantine 1
Faculté des **S**ciences de la **T**echnologie

Département d'Electronique

MEMOIRE

En vue de l'obtention du diplôme de **LICENCE** (L3-LMD)

Filière : **Automatique**

THÈME

LINE FOLLOWER WITH OBSTACLE AVOIDING ARDUINO ROBOT

Dirigé par : **M. AMMARI** et **Kh. CHEBRI**

Présenté par :

-LANACER Youcef

-MEZIANI Ayoub

-MELLIT Rassim

-IMAMI Abdelmoumen

Année universitaire : **2022 / 2023**

Meziani Ayoub

Imami Abdelmoumen

Lanacer Youcef

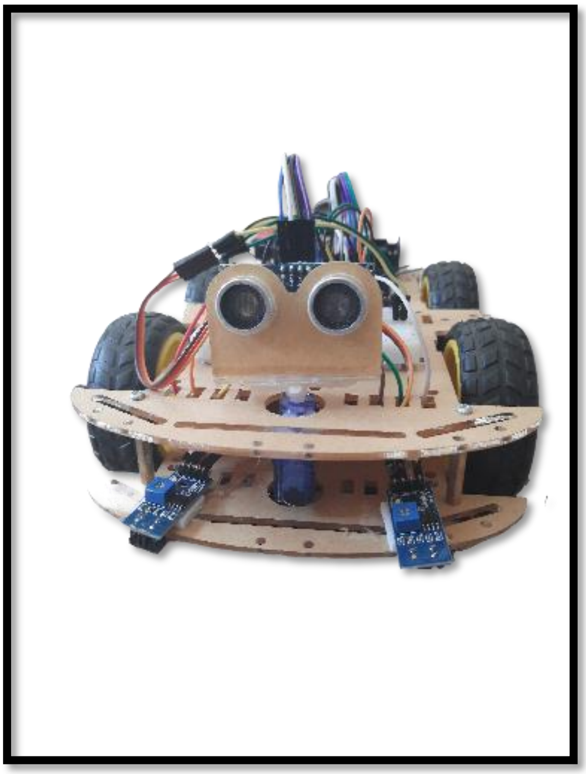
Mellit Rassim

General Introduction:

The Line Following and Obstacle Avoiding Robot Car combines the principles of robotics and coding to create a versatile autonomous vehicle capable of navigating along a predefined path while avoiding obstacles in its environment. This project represents an exciting intersection of various technologies, providing a platform for learning and experimentation in the field of robotics.

The primary objective of this project is to design and build a robot car that can autonomously follow a line or track on the ground, using specialized sensors and algorithms. By detecting and analyzing the contrast between the track and its surroundings, the robot car is able to stay on course, making precise movements based on the visual cues it receives.

In addition to line following, the robot car is equipped with obstacle avoidance capabilities. This feature enables it to detect and navigate around physical barriers or objects that may appear in its path. By incorporating sensors such as ultrasonic distance sensors and infrared sensors, the robot car can perceive its surroundings and make decisions to steer away from obstacles.



Chapter I:

Définition, Historique, différentes méthodes ou techniques utilisées

Introduction:

In this chapter we will tackle The definition of the project itself, The different methods and techniques used to achieve the functionality desired and some of the history involved around the topic.

I-1.1- Definition:

A line follower robot with obstacle avoidance is a robot that can sense a line with the help of its IR sensors and walk on the path which is to be made by the line which can be modified by the user to get his work done with the help of the robot. The robot also has an obstacle sensor that helps it avoid collision and detect collision with an obstacle sensor and hence reaching the target.

I-1.2- History:

Line follower robots are autonomous robots that can follow a line on the ground without human intervention. Obstacle avoidance is an additional feature that allows the robot to detect and avoid obstacles in its path. The Arduino platform has been used extensively for building line follower robots with obstacle avoidance capabilities.

The earliest known line follower robot was built by M. G. Michael in 1979. Since then, line follower robots have been used in various applications such as warehouse automation, transportation systems, and military applications.

I-1.3-different methods and techniques used:

To create a Line Following and Obstacle Avoiding Robot Car we implemented various methods and technologies to achieve its autonomous functionality.

1) Line Detection Sensors: Line detection is crucial for the robot car to follow a path accurately. Various sensors can be employed for this purpose, including: IR Sensors, Reflectance Sensors, Camera-based Vision Systems ...etc.

We chose to go with IR Sensors: IR sensors emit infrared light and detect its reflection to determine the presence of a line. By measuring the intensity of reflected light, the robot car can identify the position of the line.

2) Obstacle Detection Sensors: To avoid obstacles, the robot car needs sensors that can detect objects in its path. Several sensor options include: Ultrasonic Sensors, Infrared Proximity Sensors, LiDAR (Light Detection and Ranging).

We chose Ultrasonic Sensors: Ultrasonic sensors emit high-frequency sound waves and measure the time it takes for the waves to bounce back after hitting an obstacle. This information is used to calculate the distance to the obstacle.

3) Microcontrollers/Single-Board Computers: These devices serve as the brain of the robot car, handling data processing, control algorithms, and actuator commands. Popular choices include Arduino, Raspberry Pi, or specialized microcontrollers designed for robotics. We went with ARDUINO UNO

4) Actuators: Actuators are responsible for physical movement and allow the robot car to navigate its environment. We used: a. DC Motors: DC motors are used for the wheels' propulsion, providing forward, backward, and turning motions. b. Servo Motors: Servo motors are often employed for steering, enabling precise control of the robot car's direction

Conclusion:

It is important to understanding the core of the project, its functionality and its background provides a solid foot standing that allows us to dive into the depths of this topic.

<u>Chapter II:</u>

COMPONENTS USED

Intoduction:

In this chapter, we will discuss the different components that were used in our line follower with obstacle avoiding Arduino robot project. We will define each component and its parameters, as well as provide an example of code that utilizes each component. The main components used in our project include the Arduino UNO, IR sensor, L298 motor driver, and HC-SR04 ultrasonic sensor.

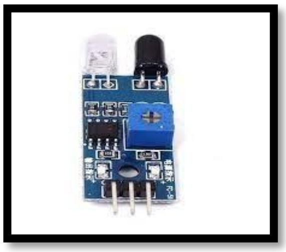
II-1.1-Definition:

<u>Arduino UNO</u>

Arduino Uno is an open-source microcontroller board based on the Microchip ATmega328P microcontroller and developed by Arduino.cc and initially released in 2010. It is categorized as a microcontroller that uses the ATmega328 as a controller in it. The board is equipped with sets of digital and analog input/output (I/O) pins that may be interfaced to various expansion boards (shields) and other circuits. It is mostly preferred by beginners for electronics projects.

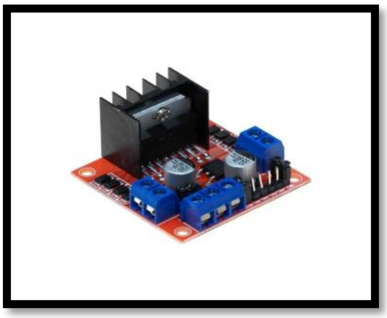
IR Sensor:

An IR sensor is an electronic device that emits light in order to sense some object of the surroundings. An IR sensor can measure the heat of an object as well as detect motion. In many electronic devices, the IR sensor circuit is a very essential module.



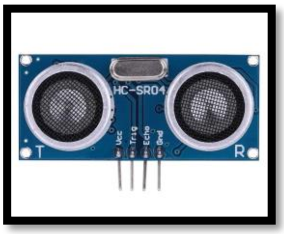
L298 Motor Driver:

This L298N Motor Driver Module is a high power motor driver module for driving DC and Stepper Motors. This module consists of an L298 motor driver IC and a 78M05 5V regulator. L298N Module can control up to 4 DC motors.



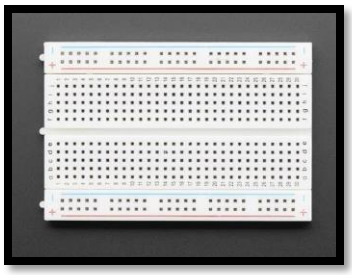
<u>Ultrasonic Sensor hc-sr04:</u>

The HC-SR04 Ultrasonic (US) sensor is an ultrasonic transducer that comes with 4 pin interface named as Vcc, Trigger, Echo, and Ground. It is very useful for accurate distance measurement of the target object and mainly works on the sound waves. The HC-SR04 is an affordable and easy to use distance measuring sensor which has a range from 2cm to 400cm (about an inch to 13 feet). One of the transducers acts as a transmitter which outputs ultrasonic sound pulses and the other acts as a receiver which listens for reflected waves.



<u>Breadboard:</u>

A breadboard is a construction base used to build semi-permanent prototypes of electronic circuits. It is a white rectangular board with small embedded holes to insert electronic components. Unlike a preboard or stripboard, breadboards do not require soldering or destruction of tracks and are hence reusable. For this reason, breadboards are also popular with students and in technological education.



DC Gear motor:

A DC gear motor is a combination of a motor and gearbox that reduces speed while increasing torque output. The gear connecting the motor and the gear head is small, which transfers more speed to the larger teeth part of the gear head and makes it rotate. The first part of the motor converts some energy into mechanical energy, and the second element transfers the available mechanical energy to the output shaft to change its speed. DC gear motors are usually small power gear motors, usually no more than 0.5 hp. They are widely used in smart home, car drives, electronic products, medical equipment, robot drives and other fields.

II-1.2-Parameters:

Arduino UNO:

- Digital I/O Pins: 14

- PWM Pins: 6 (Pin # 3, 5, 6, 9, 10 and 11)

- Analog Input Pins: 6

- DC Current per I/O Pin: 20 mA

- DC Current for 3.3V Pin: 50 mA

- Length: 68.6 mm

- Width: 53.4 mm

- Weight: 25 g

HC-SR04 ultrasonic sensor:

- Power Supply:+5V DC

- Quiescent Current: <2mA

- Working Current: 15mA

- Effectual Angle: <15°

- Ranging Distance : 2cm – 400 cm/1″ – 13ft

- Resolution : 0.3 cm

- Measuring Angle: 30 degrees

L298N Motor Driver Module:

Features & Specification:

- Driver Model: L298N 2A

- Driver Chip: Double H Bridge L298N

- Motor Supply Voltage (Maximum): 46V

- Motor Supply Current (Maximum): 2A

- Logic Voltage: 5V -
Driver Voltage: 5-35V

- Driver Current:2A

- Logical Current:0-36mA

- Maximum Power (W): 25W

- Current Sense for each motor

- Heatsink for better performance

- Power-On LED indicator

<u>DC Geared Motor</u>

Technical:

Gearbox Shape: Straight

Operating Voltage: 3 ~ 12 V

Rated Speed After Reduction: 100 RPM

Shaft Diameter: 0.55 cm (Double)

Shaft Length: 0.85 cm

Dimensions (LxWxH): 7 x 2.3 x 1.9 cm

Motor: No Load Current 40-180 mA.

II-1.3-example from the code:

<u>HC-SR04 ultrasonic sensor:</u>

```
long Ultrasonic_read(){
  digitalWrite(trigger, LOW);
  delayMicroseconds(2);
  digitalWrite(trigger, HIGH);
```

```
  delayMicroseconds(10);
  long time = pulseIn (echo, HIGH);
  return time / 29 / 2;
}
```

L298N Motor Driver Module:

```
 pinMode(enA, OUTPUT); // declare as output for L298 Pin enA
pinMode(in1, OUTPUT); // declare as output for L298 Pin in1
pinMode(in2, OUTPUT); // declare as output for L298 Pin in2
pinMode(in3, OUTPUT); // declare as output for L298 Pin in3
pinMode(in4, OUTPUT); // declare as output for L298 Pin in4
pinMode(enB, OUTPUT); // declare as output for L298 Pin enB
analogWrite(enA, 70); // Write The Duty Cycle 0 to 255 Enable Pin A for Motor1 Speed
analogWrite(enB, 70); // Write The Duty Cycle 0 to 255 Enable Pin B for Motor2 Speed
```

Servo motor:

```
pinMode(servo, OUTPUT);
for (int angle = 70; angle <= 140; angle += 5)  {
   servoPulse(servo, angle);  }
for (int angle = 140; angle >= 0; angle -= 5)  {
   servoPulse(servo, angle);  }
for (int angle = 0; angle <= 70; angle += 5)  {
   servoPulse(servo, angle);  }

distance_F = Ultrasonic_read();
```

```
delay(500);

}
```

IR Sensors:

```
//if Right Sensor is Black and Left Sensor is White then it will call turn Right function else

if((digitalRead(R_S) == 1)&&(digitalRead(L_S) == 0)){turnRight();}


//if Right Sensor is White and Left Sensor is Black then it will call turn Left function else

if((digitalRead(R_S) == 0)&&(digitalRead(L_S) == 1)){turnLeft();}
```

Conclusion:

In conclusion, understanding the components used in a project is crucial to its successful implementation.

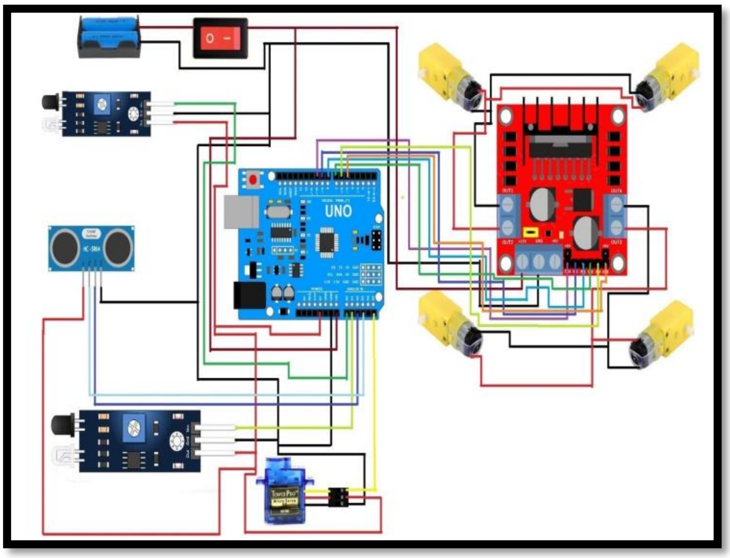
Chapter III:

HOW THE SYSTEM WORK:

Introduction:

In this chapter, we will see the overall scheme of our system, as well as the results obtained.

III-1.1- Overall scheme:



III-1.2- Component Lists:

Solderless Breadboard, 4 wheel robot car kit, Arduino UNO, IR Sensor x 2, L298 Motor Driver, Ultrasonic Sensor Holder, Ultrasonic Sensor hc-sr04, 4Pcs Smart Robot Car Tyres Wheels, Male to Female jumper Wires, Male to Male jumper Wires, Hard Jumper Wire, On/Off Switch, 18650 Battery Holder – 2 Cell , 18650 Battery Cell 3.7V x 2,SG90 mini servo motor,dc gear motor.

III-1.3- Results obtained:

After many unsuccessful attempts we finally got everything working by altering the code,

The positions of the sensors and changing the shape of the track to match the functionality that we want.

When we put the car on the track it follows the line perfectly and whenever it encounters an obstacle it avoids it by steering away either to the right or the left depending on which way is clear.

Conclusion:

- In the end the choice of components plays an important role to achieving the correct functioning of the project, that's why it is advised to spend a decent amount of time choosing the correct components.

General Conclusion:

In conclusion, this Project represents an exciting exploration into the field of robotics. It has successfully brought together technologies such as line detection sensors, obstacle detection sensors, single-board computers, and actuators to create an autonomous vehicle capable of following a line while avoiding obstacles. Through this project, we have gained valuable knowledge and experience in various aspects of robotics and programming, fostering critical thinking, problem-solving skills, and innovation. The skills acquired in this project can be further applied to diverse domains, including industrial automation (for example a cart that transports heavy cargo around a factory), autonomous vehicles (smart cars that steer themselves without human interference), and smart systems, opening a world of possibilities for future exploration and development in robotics.

-Annex:

```
#define enA 10//Enable1 L298 Pin enA

#define in1 9 //Motor1  L298 Pin in1

#define in2 8 //Motor1  L298 Pin in1

#define in3 7 //Motor2  L298 Pin in1

#define in4 6 //Motor2  L298 Pin in1

#define enB 5 //Enable2 L298 Pin enB



#define L_S A0 //ir sensor Left

#define R_S A1 //ir sensor Right

#define echo A2    //Echo pin

#define trigger A3 //Trigger pin

#define servo A5

int Set=15;

int distance_L, distance_F, distance_R;

void setup(){ // put your setup code here, to run once

Serial.begin(9600); // start serial communication at 9600bps

pinMode(R_S, INPUT); // declare if sensor as input

pinMode(L_S, INPUT); // declare ir sensor as input

pinMode(echo, INPUT );// declare ultrasonic sensor Echo pin as input

pinMode(trigger, OUTPUT); // declare ultrasonic sensor Trigger pin as Output

pinMode(enA, OUTPUT); // declare as output for L298 Pin enA
```

```
pinMode(in1, OUTPUT); // declare as output for L298 Pin in1
pinMode(in2, OUTPUT); // declare as output for L298 Pin in2
pinMode(in3, OUTPUT); // declare as output for L298 Pin in3
pinMode(in4, OUTPUT); // declare as output for L298 Pin in4
pinMode(enB, OUTPUT); // declare as output for L298 Pin enB
analogWrite(enA, 85); // Write The Duty Cycle 0 to 255 Enable Pin A for Motor1 Speed
analogWrite(enB, 85); // Write The Duty Cycle 0 to 255 Enable Pin B for Motor2 Speed
pinMode(servo, OUTPUT);
for (int angle = 70; angle <= 140; angle += 5)  {
  servoPulse(servo, angle);  }
for (int angle = 140; angle >= 0; angle -= 5)  {
  servoPulse(servo, angle);  }
for (int angle = 0; angle <= 70; angle += 5)  {
  servoPulse(servo, angle);  }
distance_F = Ultrasonic_read();
delay(500);
}
```



```
void loop(){



//==============================================
//     Line Follower and Obstacle Avoiding
//==============================================
distance_F = Ultrasonic_read();
```

```
Serial.print("D F=");Serial.println(distance_F);

//if Right Sensor and Left Sensor are at White color then it will call forword function

if((digitalRead(R_S) == 0)&&(digitalRead(L_S) == 0)){

 if(distance_F > Set){forword();}

                 else{Check_side();}

}

//if Right Sensor is Black and Left Sensor is White then it will call turn Right function

else if((digitalRead(R_S) == 1)&&(digitalRead(L_S) == 0)){turnRight();}

//if Right Sensor is White and Left Sensor is Black then it will call turn Left function

else if((digitalRead(R_S) == 0)&&(digitalRead(L_S) == 1)){turnLeft();}

delay(10)

}


void servoPulse (int pin, int angle){

int pwm = (angle*11) + 500;      // Convert angle to microseconds

digitalWrite(pin, HIGH);

delayMicroseconds(pwm);

digitalWrite(pin, LOW);

delay(50); // Refresh cycle of servo

}
```

```
//*********************Ultrasonic_read***************************

long Ultrasonic_read(){

  digitalWrite(trigger, LOW);

  delayMicroseconds(2);

  digitalWrite(trigger, HIGH);

  delayMicroseconds(10);

  long time = pulseIn (echo, HIGH);

  return time / 29 / 2;

}



void compareDistance(){

    if(distance_L > distance_R){

  turnLeft(); // RIGHT

  delay(600);

  forword();

  delay(600);

  turnRight();

  delay(400);

  forword();

  delay(300);

  turnRight();

  delay(200);

  }
```

```
  else{
  turnRight(); // LEFT
  delay(600);
  forword();
  delay(600);
  turnLeft();
  delay(370);
  forword();
  delay(300);
  turnLeft();
  delay(270);
  }
}
void Check_side(){
Stop();
    delay(100);
for (int angle = 70; angle <= 140; angle += 5)  {
   servoPulse(servo, angle);  }
    delay(300);
    distance_R = Ultrasonic_read();
    Serial.print("D R=");Serial.println(distance_R);
    delay(100);
  for (int angle = 140; angle >= 0; angle -= 5)  {
```

```
   servoPulse(servo, angle);  }
    delay(500);
    distance_L = Ultrasonic_read();
    Serial.print("D L=");Serial.println(distance_L);
    delay(100);
for (int angle = 0; angle <= 70; angle += 5)  {
   servoPulse(servo, angle);  }
    delay(300);
compareDistance();
}
void forword(){  //forword
digitalWrite(in1, LOW); //Left Motor backword Pin
digitalWrite(in2, HIGH); //Left Motor forword Pin
digitalWrite(in3, HIGH); //Right Motor forword Pin
digitalWrite(in4, LOW); //Right Motor backword Pin
}
void backword(){ //backword
digitalWrite(in1, HIGH); //Left Motor backword Pin
```



```
digitalWrite(in2, LOW); //Left Motor forword Pin
digitalWrite(in3, LOW); //Right Motor forword Pin
digitalWrite(in4, HIGH); //Right Motor backword Pin
}
void turnRight(){ //turnRight
digitalWrite(in1, LOW); //Left Motor backword Pin
```

```
digitalWrite(in2, HIGH); //Left Motor forword Pin

digitalWrite(in3, LOW); //Right Motor forword Pin

digitalWrite(in4, HIGH); //Right Motor backword Pin

}

void turnLeft(){ //turnLeft

digitalWrite(in1, HIGH); //Left Motor backword Pin

digitalWrite(in2, LOW); //Left Motor forword Pin

digitalWrite(in3, HIGH); //Right Motor forword Pin

digitalWrite(in4, LOW); //Right Motor backword Pin

}

void Stop(){ //stop

digitalWrite(in1, LOW); //Left Motor backword Pin

digitalWrite(in2, LOW); //Left Motor forword Pin

digitalWrite(in3, LOW); //Right Motor forword Pin

digitalWrite(in4, LOW); //Right Motor backword Pin

}
```

-Bibliography:

.Introduction to HC-SR04 (Ultrasonic Sensor) - The Engineering Projects. https://www.theengineeringprojects.com/2018/10/introduction-to-hc-sr04-ultrasonic-sensor.html.

. What is Arduino UNO? | Amazing 11 Features of Arduino UNO - EDUCBA. https://www.educba.com/what-is-arduino-uno/.

. Introduction to Arduino Uno - The Engineering Projects. https://www.theengineeringprojects.com/2018/06/introduction-to-arduino-uno.html.

. Complete Guide for Ultrasonic Sensor HC-SR04 with Arduino. https://randomnerdtutorials.com/complete-guide-for-ultrasonic-sensor-hc-sr04/.

. IR sensor Working Principle and Applications | Robu.in. https://robu.in/ir-sensor-working/.

. IR Sensor : Circuit, Types, Working Principle & Its Applications. https://www.watelectronics.com/ir-sensor/.

. What is an IR Sensor?| IR Sensor | Infrared Sensor - Techatronic. https://techatronic.com/what-is-an-ir-sensor/.

. Solderless Breadboards - Digilent Reference. https://digilent.com/reference/learn/fundamentals/electronic-components/solderless-breadboard/start.

. L298N Module Pinout Configuration .

https://components101.com/modules/l293n-motor-driver-module

.IR SENSOR IMAGE SOURCE

https://sharvielectronics.com/product/ir-sensor-module/

.ULTRASONIC IMAGE SOURCE

https://www.thegeekpub.com/wiki/sensor-wiki-ky-050-hc-sr04-ultrasonic-sensor/

.L298 MOTOR IMAGE SOURCE

https://www.robotistan.com/l298n-pair-motor-driver-board-with-voltage-regulator

We would like to express our gratitude to all those who have supported and assisted us throughout this project.Your guidance, encouragement, and contributions have been immensely valuable.

Thank you all for your support.

THE END