**INTRODUCTION TO MECHANICAL ENGINEERING – MECH201**

Project: Line tracking robot

**Abbas-Nahle Dabdoub, Rami Kronbi, Bahaa Doucmak**

Rafik Hariri University, College of Engineering, Department of Mechanical and Mechatronics

*Abstract*

*The objective of this experiment is to build a line tracking robot. An Arduino UNO as well as other circuit components were connected together and a program was written on the Arduino IDE to perform the experiment. The objective of this experiment was achieved, the robot was able to track a black like over a white surface while moving at a good speed.*

**Keywords:** Arduino UNO, Arduino IDE, DC Motor, L298N Motor Drive, IR Sensor, BW Sensor, LM2596 DC Step Down Voltage Regulator.

1. INTRODUCTION

This experiment aims to make a line tracking robot that can track a black line over a white surface through the use of black and white sensors (BW sensors) and an IR sensor. To be more specific, the BW sensors will ensure that the robot remains on track, and the IR sensor will ensure that the robot does not hit any obstacle in its path.

1. **MATERIALS AND METHODS**

**2.1 Materials**

The following components were used to make the project prototype: An Arduino UNO, an L298N motor drive, 4 DC motors, 3 lithium ion batteries, a battery pack for 3 batteries, a switch, 4 BW sensors, a LM2596 DC step down voltage regulator, an IR sensor, a wood plank for the car chassis, and a 3D printed car chassis attachment to place the BW sensors on. The following software were used to make this project prototype: Arduino IDE, Tinkercad and Fritzing.

Figure 1 shows the Arduino UNO with all its pins, being mainly the digital pins for both input and output, the PWM pins (embedded in some of the digital pins, which are those having a tilde symbol next to them) being for analog outputs, the analog input pins being for analog inputs, and the power pins having the 5V and ground pins being for their corresponding voltage values and the Vin pin to supply the Arduino UNO with power. Moreover, figure 2 shows a DC motor, which is used as an output; one of its wires should be connected to ground, and its other wire should be connected to 12V (for maximum speed). Furthermore, figure 3 shows an IR sensor, which is a digital sensor used to determine if there is an obstacle at a certain distance (which can be changed) or not; its blue wire is connected to ground, its brown wire is connected to 5V, and its black wire is connected to a digital pin of the Arduino UNO, and it sends the Arduino UNO a “0” when it detects an obstacle, and sends the Arduino UNO a “1” when it does not detect an obstacle. Figure 4 shows a BW sensor, which sends the Arduino UNO a “1” when it detects black, and sends the Arduino UNO a “0” when it detects white. Nonetheless, figure 5 shows a LM2596 DC step down voltage regulator, which takes a voltage that may change with time as an input and gives a lower voltage that’s value can be changed as an output. Additionally, figure 6 shows an L298N motor drive, which is used to control the direction and speed of DC motors. [3]

|  |  |
| --- | --- |
| **FIGURE 1:** ARDUINO UNO WITH LABELED PINS [3] | **FIGURE 2:** DC MOTOR [3] |
| **FIGURE 3:** IR SENSOR [6] |
| **FIGURE 4:** BW SENSOR [5] | **FIGURE 6:** L298N MOTOR DRIVE WITH LABELED PINS [4] |
| **FIGURE 5:** LM2596 DC STEP DOWN VOLTAGE REGULATOR [7] |

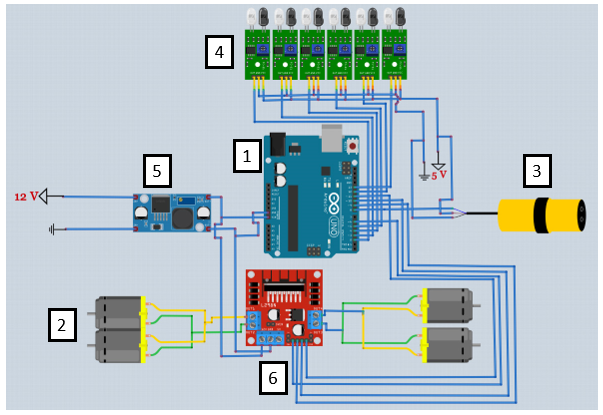
**2.2 Method**

The connections were as follows: the 3 lithium ion batteries were placed in a battery pack for 3 batteries. The positive terminal of the battery pack was connected to the common pin of a switch, and the normally open pin of the switch was connected to the positive input of the LM2596 DC step down voltage regulator (which is set to output 10V). Additionally, the negative terminal of the battery pack was connected to the negative input of the LM2596 DC step down voltage regulator. The positive output of the LM2596 DC step down voltage regulator was connected to the Vin pin of the Arduino UNO (to be able to supply the Arduino UNO with electrical power), and the GND pin of the Arduino UNO was connected to the negative output of the LM2596 DC step down voltage regulator (to create a common ground). Furthermore, the GND pin of the Arduino UNO was connected to the GND pin of the L298N motor drive, and the positive output of the LM2596 DC step down voltage regulator was connected to the 12V pin of the L298N motor drive. The OUT1 pin of the L298N motor drive was connected to the positive terminal of 2 of the DC motors, and the OUT2 pin of the L298N motor drive was connected to the negative terminal of the same 2 DC motors. The OUT3 pin of the L298N motor drive was connected to the positive terminal of the 2 remaining DC motors, and the OUT4 pin of the L298N motor drive was connected to the negative terminal of the same 2 DC motors. The IN1 pin of the L298N motor drive was connected to digital pin 6 of the Arduino UNO, and IN2 pin of the L298N motor drive was connected to digital pin 9 of the Arduino UNO (to use the PWM, making it an analog output). The IN3 pin of the L298N motor drive was connected to digital pin 10 of the Arduino UNO, and IN4 pin of the L298N motor drive was connected to digital pin 11 of the Arduino UNO (also to use PWM to make it an analog output). Nonetheless, the GND pin of each BW sensor was connected to the GND pin of the Arduino UNO, the VCC pin of each BW sensor was connected to the 5V pin of the Arduino UNO, and the D0 pins of the BW sensors were connected each to one of the following digital pins of the Arduino UNO: digital pin 2, digital pin 3, digital pin 4, digital pin 5. Moreover, the blue wire of the IR sensor was connected to the GND pin of the Arduino UNO, the brown wire of the IR sensor was connected to the 5V pin of the Arduino UNO, and the black wire of the IR sensor was connected to digital pin 8 of the Arduino UNO.

The code description is as follows: The Arduino IDE was opened, variables for the IR sensor’s digital pin, each BW sensor’s digital pins and each DC motor’s digital pin were declared and set equal to their digital pin values. Moreover, variables for the readings of the BW sensors and IR sensor were declared. A function called “moveCar” was made to control the speed of each DC motor without having to repeat similar lines of code many times. In the void setup, the BW sensors’ digital pins and IR sensor’s digital pin were set as inputs, and the digital pins of the DC motors were set as outputs. Furthermore, in the void loop, the robot is programmed to read the values from each BW sensor and the IR sensor, and evaluate if statements to determine how to move the DC motors.

The BW sensors were mounted onto the 3D printed part, which was physically connected to the wooden plank that makes up the car chassis, the DC motors were glued to the bottom of the wooden plank, and the other electrical circuit components were mounted onto the wooden plank.

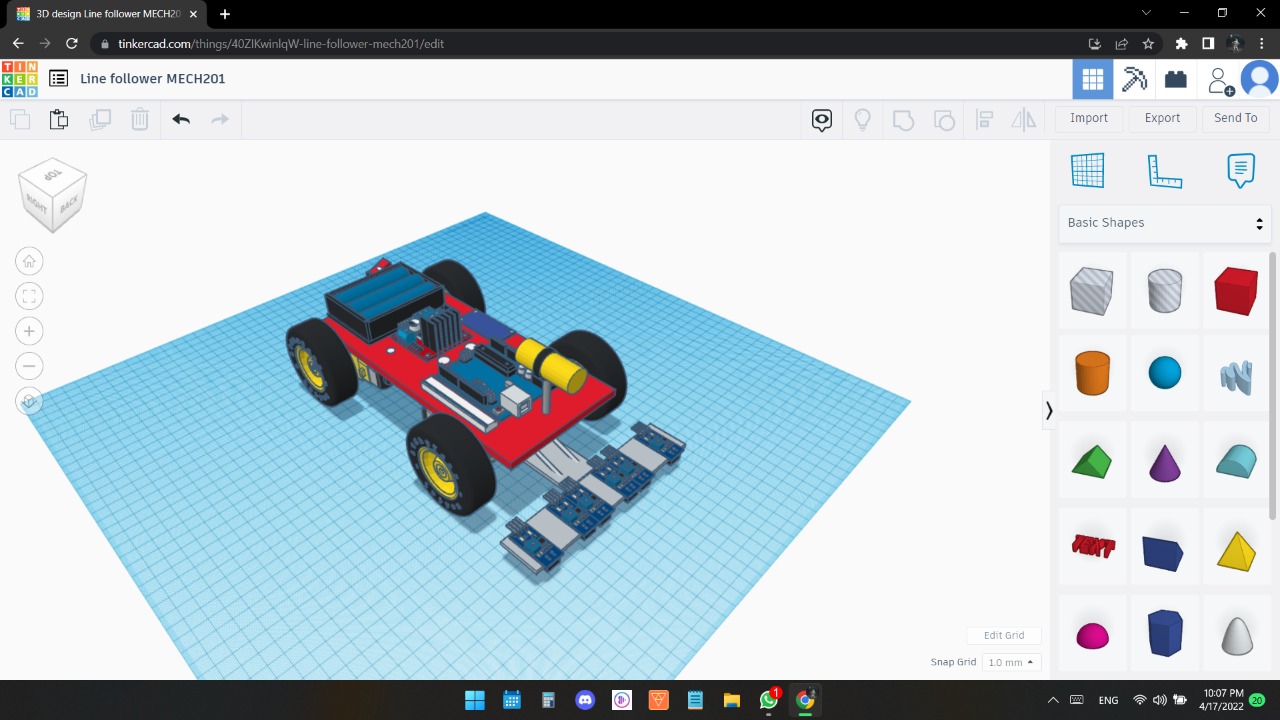
Figure 7 shows the circuit simulated on Fritzing, table 1 shows what each labeled component is, and figure 8 shows the line tracking robot designed on Tinkercad.



**FIGURE 7:** LABELED FRITZING CIRCUIT [8]

**TABLE 1:** CIRCUIT COMPONENTS NUMBERS AND THEIR CORRESPONDING CIRCUIT COMPONENTS

|  |  |  |
| --- | --- | --- |
|  | Circuit Component Number | Circuit Component Name |
|  | 1 | Arduino UNO |
|  | 2 | DC Motor |
|  | 3 | IR Sensor |
|  | 4 | BW Sensor |
|  | 5 | LM2596 DC Step Down Voltage Regulator |
|  | 6 | L298N Motor Drive |

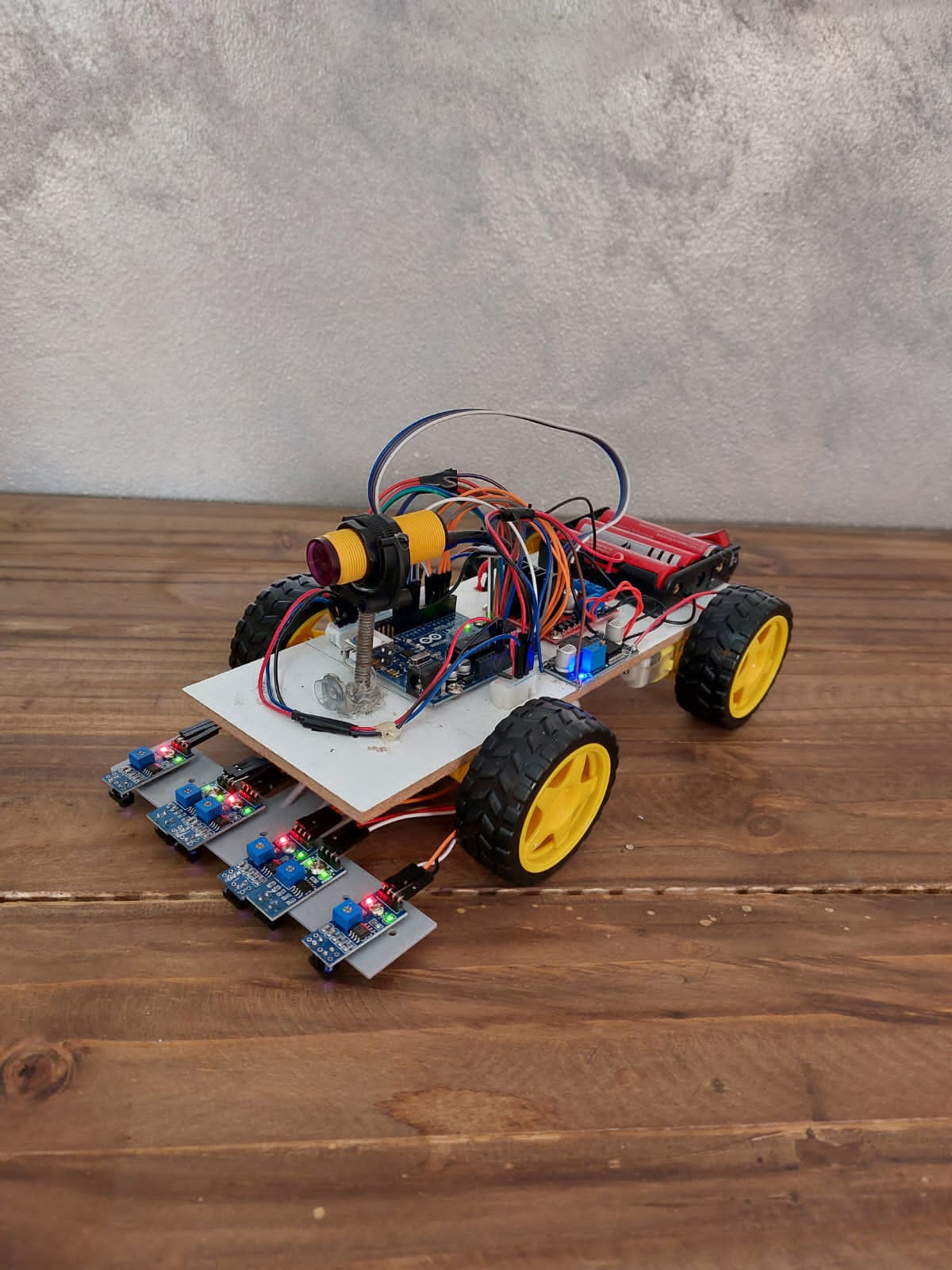


**FIGURE 8:** TINKERCAD DESIGN [1]

1. **RESULTS AND DISCUSSION**

The code mechanism is as follows: The Arduino UNO reads from all the BW sensors and the IR sensor. Moreover, it evaluates the first if statement to determine if the IR sensor is detecting an obstacle in front of the robot or not; if it does not detect an obstacle in front of the robot, it will call the function “moveCar” to move the robot forward by moving all DC motors at around half its maximum speed and then evaluate the if statements that relate to the BW sensors, and if it detects an obstacle in front of the robot, it will call the function “moveCar” to stop moving the robot forward by stopping all DC motors. The if statements that pertain to the BW sensors are as follows: it evaluates the first if statement to determine if the any of the two rightmost BW sensors is detecting a black line or a white surface; if it is detecting a black line, it will call the function “moveCar” to turn the robot to the right by moving the right DC motors (wheels) in reverse and the left DC motors (wheels) forward at their set speeds. After that, it evaluates the second if statement to determine if the middle-right BW sensor is detecting a black line or a white surface; if it is detecting a black line, it will call the function “moveCar” to turn the robot to the right by moving the right DC motors (wheels) in reverse and the left DC motors (wheels) forward at their set speeds. Afterwards, it evaluates the third if statement to determine if the middle-left BW sensor is detecting a black line or a white surface; if it is detecting a black line, it will call the function “moveCar” to turn the robot to the left by moving the left DC motors (wheels) in reverse and the right DC motors (wheels) forward at their set speeds. Next, it evaluates the fourth if statement to determine if the any of the two leftmost BW sensors is detecting a black line or a white surface; if it is detecting a black line, it will call the function “moveCar” to turn the robot to the left by moving the left DC motors (wheels) in reverse and the right DC motors (wheels) forward at their set speeds.

Figure 9 shows line tracking robot made.



**FIGURE 9:** THE LINE TRACKING ROBOT

1. **CONCLUSION**

In conclusion, the goal of the project was achieved; the code and the robot worked as intended. The robot was able to track a black line over a white surface through the use of the 4 BW sensors, and the robot did not hit any obstacle due to the use of the IR sensor.

**REFERENCES**

[1] www.tinkercad.com

[2] www.arduino.cc

[3] Class notes of the MECH201 course

[4] www.components101.com/modules/l293n-motor-driver-module

[5] www.indiamart.com/proddetail/tcrt5000-line-tracking-sensor-module-19967875997.html

[6] www.twinschip.com/Proximity\_Sensor\_E18-D80NK

[7] www.encrypted-tbn0.gstatic.com/images?q=tbn:ANd9GcTRIm-OFngoHH9OWWJkeyru5VN3SIfMm2HTcA&usqp=CAU

[8] www.fritzing.org/

**APPENDIX**

The following is the Arduino code:

//Sensor pins

#define Bw1 4

#define Bw2 12

#define Bw3 2

#define Bw4 5

#define Bw5 13

#define Bw6 3

#define IR 8

//Motor pins

#define Rightpos 10

#define Rightneg 11

#define Leftpos 9

#define Leftneg 6

//Speed values

#define turnSpeed 150

#define maxSpeed 255

//Sensor readings

bool V1;

bool V2;

bool V3;

bool V4;

bool V5;

bool V6;

bool IRreading;

//Function to control motors

void moveCar(int Leftspeed1, int Leftspeed2, int Rightspeed1, int Rightspeed2){

analogWrite(Leftpos, Leftspeed1);

analogWrite(Leftneg, Leftspeed2);

analogWrite(Rightpos, Rightspeed1);

analogWrite(Rightneg, Rightspeed2);

}

void setup(){

//B/w sensors

pinMode(Bw1, INPUT);

pinMode(Bw2, INPUT);

pinMode(Bw3, INPUT);

pinMode(Bw4, INPUT);

pinMode(Bw5,INPUT);

pinMode(Bw6,INPUT);

//IR sensor

pinMode(IR, INPUT);

//Motors

pinMode(Rightpos,OUTPUT);

pinMode(Rightneg,OUTPUT);

pinMode(Leftpos,OUTPUT);

pinMode(Leftneg,OUTPUT);

digitalWrite(8,HIGH);1

}

void loop(){

V1= digitalRead(Bw1);

V2= digitalRead(Bw2);

V3= digitalRead(Bw3);

V4= digitalRead(Bw4);

V5= digitalRead(Bw5);

V6= digitalRead(Bw6);

IRreading= digitalRead(IR);

//Motor control statements

if (IRreading == 1){

moveCar(125,0,125,0);

if (V1== 1 || V2== 1)

moveCar(maxSpeed,0,0,maxSpeed);

else if (V3== 1)

moveCar(maxSpeed,0,0,turnSpeed);

else if (V4== 1)

moveCar(0,turnSpeed,maxSpeed,0);

else if (V5== 1 || V6== 1)

moveCar(0,maxSpeed,maxSpeed,0);

}

else if (IRreading== 0)

moveCar(0,0,0,0);

}