A picture containing indoor

Description automatically generated

ROBOT PROJECT

USING LFW ALGORITHM

CSCI291

Bassam Babar (7054865)

Suraksha Kotte (6996656)

# INTRODUCTION

In this project, we had to build a robot out the of materials provided and program it to follow the black-lined path. We had to code the robot using the techniques we learned in CSCI291. The software we used for this project was Arduino. We were allowed to use 6 IR sensors. This report will include the explanation of the program with the design implementation and explain with the

# SOFTWARE USED

The program used for this robot is Arduino, which is a free and open-source electronics platform with basic hardware and software. Arduino boards can read inputs such as a light on a sensor and convert them into outputs such as operating a motor. We will program and execute commands for the robot using the ROMEO BLE board.

# MATERIALS

Breadboard

3 IR Sensors

LDR Sensors & Resistor

2 DC motors

3 wheels

L298N Motor drivers

Arduino Uno

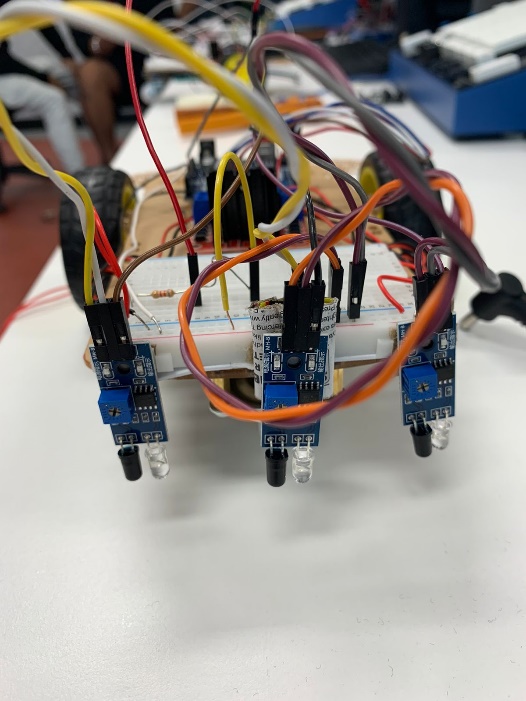
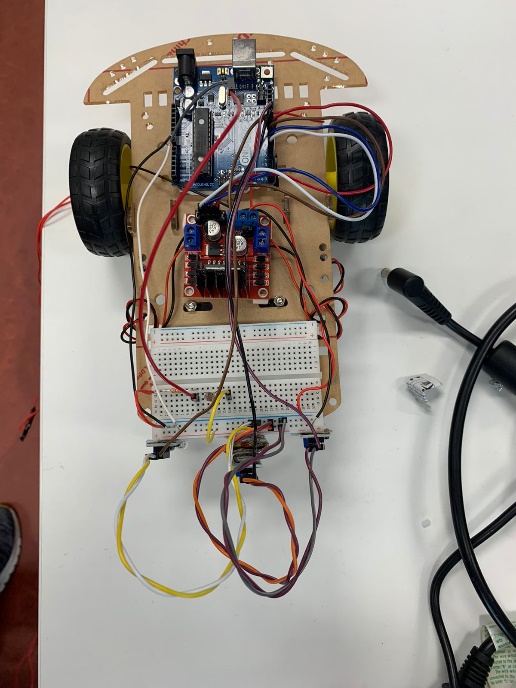
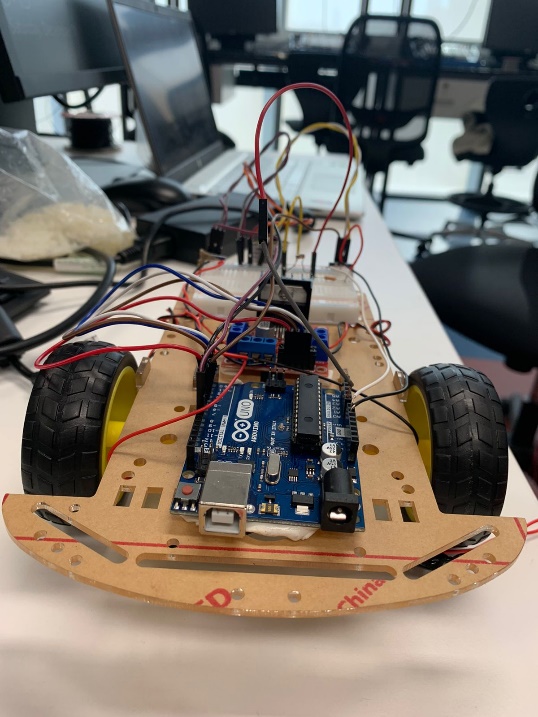
Wires & Jumper cables

Black tape

## FUNCTIONALITY OF THE MATERIALS

* The Arduino Uno is a free and open-source microcontroller board based on the Microchip ATmega328P. The board is constructed in such a way that it can be reset by software running on a computer to which it is linked.
* We utilized a sensor called an LDR to detect the intensity of light (light dependent resistor). The LDR is a type of resistor that lets higher voltages (low resistance) to flow through it when there is a high intensity of light.
* Infrared Ray Sensors are used to determine the position of a line follower in relation to the position of the robot. Line sensors detect the existence of a black line by emitting infrared (IR) light and detecting the light that return to the sensor. The device will produce infrared light, and the sensor will measure the amount of light reflected from the surface underneath it.
* The L298N is a dual H-Bridge motor driver that can control the speed and direction of two DC motors at the same time.
* DC motors generate to make the wheels move.

# ASSEMBLING OF THE ROBOT

A picture containing automaton, miller

Description automatically generated

1. We attached the motors to the cardboard base using screws and bolts.
2. Set the wheels on the side and one below the base.
3. Attached the breadboard.
4. Set up the IDR sensor and resistor.
5. Attached the Arduino uno
6. With the help of the wires, we placed the sensors vertically in the front part.

# POSITIONING OF THE SENSORS

We used 3 infrared sensors in total. One in the middle and one on each side are held up vertically to ease the connection.

The central sensor protrudes out and takes the input to detect the black tape. This sensor only provides directional output to go straight.

The other sensors are solely utilized for directional purposes, such as the left sensor, which leads the robot to turn left, and the right sensor, which makes the robot to turn right.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Inputs** | |  | **Outputs** | | **Direction** |
| **Left Sensor**  **L** | **Right Sensor**  **R** | **Middle**  **Sensor**  **MA** | **Right Motor M1** | **Left Motor M2** |
| 0 | 0 | 1 | 1 | 1 | Forward |
| 1 | 0 | 0 | 0 | 1 | Left |
| 0 | 1 | 0 | 1 | 0 | Right |
| 0 | 0 | 0 | 0 | 0 | Stop |

# CONSTRUCTION OF MAP

Icon

Description automatically generated

# DESIGN SOLUTION DESCRIPTION

SWITCH ON

TRUE

MOVE FORWARD

SENSOR

L=0,R=0,MA=1

FALSE

TRUE

SENSOR

L=1,R=0,MA=0

MOVE LEFT

FALSE

SENSOR

L=0,R=1,MA=0

TRUE

MOVE RIGHT

FALSE

TRUE

STOP

SENSOR

L=0,R=0,MA=0

# PROGRAM CODE

Graphical user interface, text, application, email

Description automatically generated

The speed and direction pin go to the red board. From the red board there are out pins and each pin is connected to their respective component.

M1 is right, M2 is left.

The right, left and middle sensors are connected to pin 8,10,9 respectively.

We initialized the speed for both the right and left wheels as 50.

In void setup()

The function pinmode was used to configure if the functions are input or output.

In void loop()

The outputs of sensors are saved as integer values.

Table

Description automatically generated with low confidence

If statements were used in the void loop to control the robot's movement.

Each if statement included a different condition under which the robot would perform a certain duty.

If left, right sensor doesn’t detect black line and middle sensor detects then call the function

carF() – function to go forward

If left sensor doesn’t detect black line and middle, right sensors detects then call the function

carF() – function to go forward

If left, middle sensors doesn’t detect black line and right sensor detects then call the function

carR() – function to go right

If right sensor doesn’t detect black line and left, middle sensors detects then call the function

carL() – function to go left

If left, middle and right sensors detects then call the function

carL() – function to go left

If no black lines are detected then call the function

carL() – function to go left and delay

Table

Description automatically generated

Definitions of the functions

The movement of each motor was controlled by two functions called analogWrite and digitalWrite.

Due to setup of our motors with the Arduino and breadboard, we were able to move the desired motors with these functions

void carStop()- for the motor to stop completely

void carF()- for the motor to move forward

void carBack() – to move the motor backward

void carR() – to move the motor right

void carL() – to move the motor left

# PROJECT TESTING EVALUATION

As we had to improvise while testing, we had to use DC supply voltage to power the robot which can also be considered as a limitation as there is no constant voltage.

Another limitation is that the wires kept getting tangled with the robot and restricted the flow and momentum. It also messes up the sensor calibration if the wires hit it.

We tested the robot many times with different speeds and voltages for each function. We also tried to add the delay function to test if each function is working.

Despite the limitations, the robot followed every direction and implemented well.

The code is proper and runs well without any errors.

# PROJECT VIDEO LINK

<https://we.tl/t-gbfvIKfmSc>

Project management

WEEK 4: First, we received the materials of the robot. Then, we started assembling the board by reading the instructions and fixed the wheels along with the motors.

WEEK 5: From Moodle, we read the left hand on the wall algorithm case study and tried understanding the robotic maze solving tutorial.

WEEK 6: We installed the Arduino and went through the project files given on Moodle.

WEEK 7: We placed the IR sensors a bit later because we were not sure how many we would need. After watching the video on how to place the sensors, we got an idea about the sensors placing and decided on 3 IR sensors.

WEEK 8: We accessed the program files from Moodle and started coding.

WEEK 9: We set up the black lined path and started testing out the robot. There were many trial and errors made during the process.

WEEK 10: Few functions started working such as the robot moving straight.

Week 11: We worked on the other functions such as right, left and recorded the video of the robot working. Then, we started making the report with the design implementation and explanation of the code.

# Intra Peer Assessment

**Student Names:**

(i) As a group we agree to divide the marks evenly.

E-Signatures: Suraksha

Bassam

(ii) We have decided to apply the following weighting to the work carried out during completion of the project.

Name Weighting (%)

Suraksha Kotte \_\_\_\_\_40\_\_\_\_\_\_\_\_\_\_

Bassam Babar \_\_\_\_\_60\_\_\_\_\_\_\_\_\_\_

Signatures: Suraksha, Bassam

# APPENDIX

//defnine which pin is connected to which component

#define speedPin\_M1 5

#define speedPin\_M2 6

#define directionPin\_M1 4

#define directionPin\_M2 7

#define R 8

#define MA 9

#define L 10

int rightSpeed = 50;

int leftSpeed = 50;

void setup()

{

//M1 RIGHT

//M2 LEFT

Serial.begin(9600);

pinMode(R,INPUT); // Refering the pins as an input or output

pinMode(MA,INPUT);

pinMode(L,INPUT);

pinMode(speedPin\_M1, OUTPUT);

pinMode(speedPin\_M2, OUTPUT);

pinMode(directionPin\_M1, OUTPUT);

pinMode(directionPin\_M2, OUTPUT);

}

void loop(){

int rv = digitalRead(R); //saving the outputs of the sensors to int values

int lv = digitalRead(L);

int cv = digitalRead(MA);

if(lv == 0 && cv == 1 && rv == 0) // if statement to go forward if only middle IR has detected a black line

{

carF();

}

else if(lv == 0 && cv ==1 && rv == 1) //if statement to go forward if only right and middle IR has detected a black line

{

carF();

}

else if(lv == 0 && cv ==0 && rv == 1) //if statement to go right if only right IR has detected a black line

{

carR();

}

else if(lv==1 && cv==1 && rv == 0)//if statement to go left if left and middle IR has detected a black line

{

carL();

}

else if(lv==1 && cv==1 && rv == 1)//if statement to go left if all IR has detected a black line

{

carL();

}

else if(lv==0 && cv==0 && rv == 0)//if statement to take u turn f no black lines detected.

{

carL();

delay(1);

}

}

void carStop() { // Motor Stop

digitalWrite(speedPin\_M2, 0);

digitalWrite(directionPin\_M1, LOW);

digitalWrite(speedPin\_M1, 0);

digitalWrite(directionPin\_M2, LOW);

}

void carF() { //Move forward

analogWrite (speedPin\_M2, 50);

digitalWrite(directionPin\_M1, LOW);

analogWrite (speedPin\_M1, 50);

digitalWrite(directionPin\_M2, LOW);

}

void carBack() { //Move backward

analogWrite (speedPin\_M2, 50);

digitalWrite(directionPin\_M1, HIGH);

analogWrite (speedPin\_M1, 50);

digitalWrite(directionPin\_M2, HIGH);

}

void carR() { //Turn Right

analogWrite (speedPin\_M2, 50);

digitalWrite(directionPin\_M1, HIGH);

analogWrite (speedPin\_M1, 50);

digitalWrite(directionPin\_M2, LOW);

}

void carL() { //Turn Left

analogWrite (speedPin\_M2, 50);

digitalWrite(directionPin\_M1, LOW);

analogWrite (speedPin\_M1, 50);

digitalWrite(directionPin\_M2, HIGH);

}