Assignment 3 - Final Design Project Report and Demonstration

EEET2610

Project: Autonomous toy car

Group: 10

RMIT UNIVERSITY VN

2022 - SEM 2

Members:

Tran Xuan Hoang Dat - s3651550

Nguyen Thanh Long – s3864007

Nguyen Cong Phuong - s3804846

Tran Vu Hoang Viet- s3740797

Professor: Peter Claiden

RMIT UNIVERSITY

Executive summary/ Abstract

The problem of slow reaction to braking while driving is the usual mistake that leads to accidents. This project aims to ensure safety driving by implementing the avoidant system that uses the ultrasonic sensor. The project will extensively adapt a toy car, with the intention of adding capabilities such as collision avoidance, line following and manual remote control. This will be done with the on-board addition of sensors (infra-red) connected to an Arduino with bespoke programmer. In addition, this hardware will be controlled remotely via Raspberry Pi.

Table of Content

Table of Figure

Table

1.Introduction/ Project Overview

After 11 weeks of working, this report will show with tasks have been achieved and what is the next task we need to focus on. In the most recent week, our project has been spirited into 3 main tasks based on the features of the Arduino car project. The first task is to avoid automatic objects by using ultrasonic. The second task is the following line by using the single line detector. The last task is switch from automatic mode to manual mode and control the car by the remote. Besides three main tasks, the motor of the car needs to be set up to turn right and left. The plastic geared micro servo motor, which holds the ultrasonic, needs to maintain look forward and rotate when the ultrasonic sensor detects a barrier. Also, all the sensor and motor need to connect logically and neatly inside the motor driver shield. Therefore, the car needs to be designed big enough to fit in the car body.

2.Motivation / Scope of work

Vietnam is developing day by day. Also, the people's living standard is increasing day by day. Nowadays, people mainly travel by motorbike and public transport. But in the next 10 years, people will use cars as their daily transport. The project will help the car user travel without worrying about how to get there. The car can travel by itself to the destination without hitting another car, obstacles or walker. That is the main reason why our team chose this project.

3.Robot design

The front of the car consists of 2 circuit boards connected, a switch and a power box for the car (battery), a sensor's rotation motor and an obstacle detection sensor.

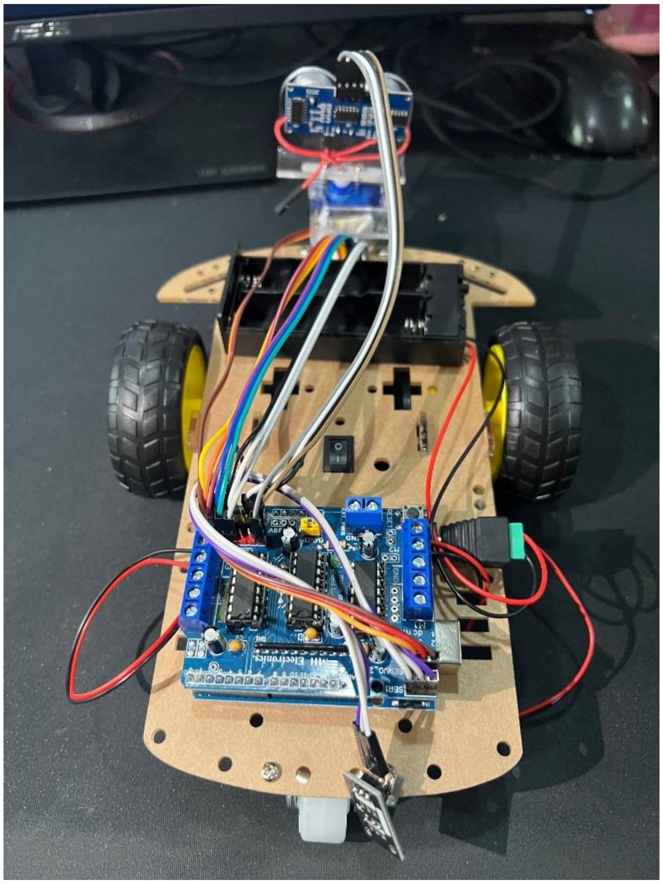


Figure 1:The top and bottom of the car [1],[2],[3],[4],[5],[6],[7],[8] and [9]

The rear of the car includes the following parts: swivel wheel, 2 rear wheel motors and 3-line detector circuits (TCRT5000).

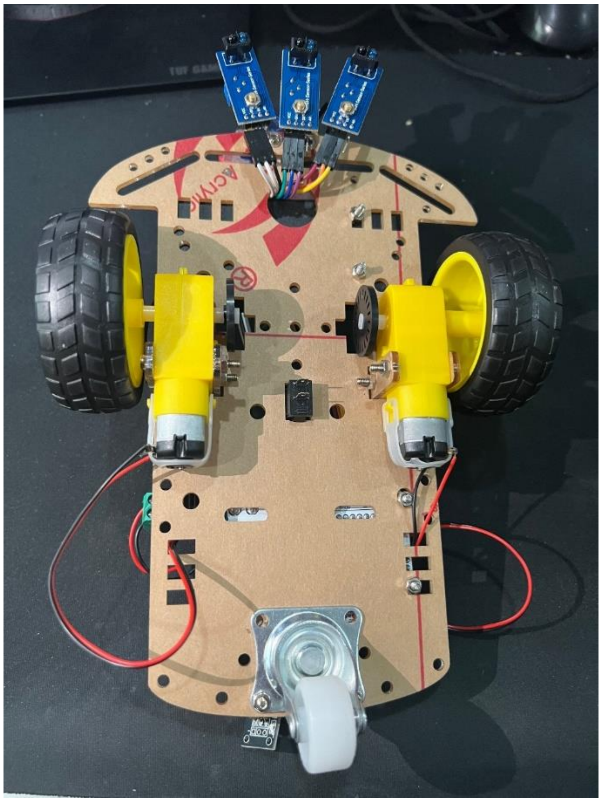


Figure 2: Under the car

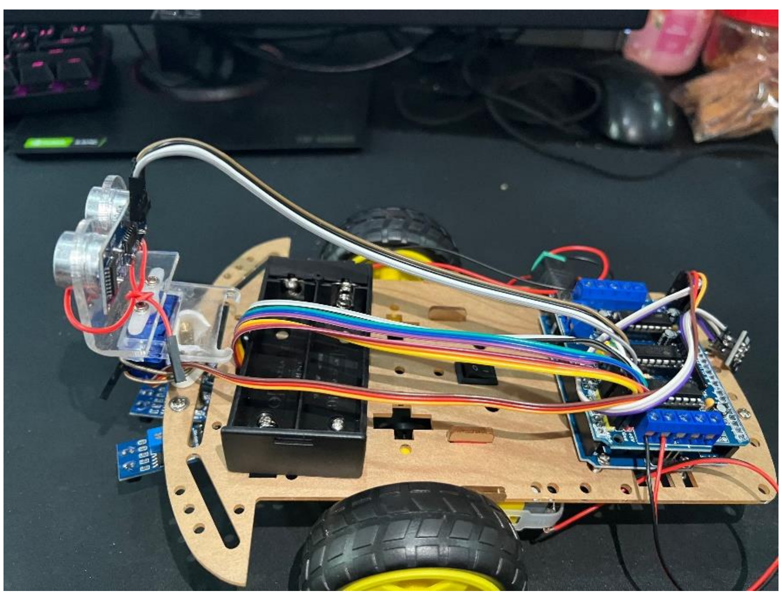


Figure 3: Left side of the car

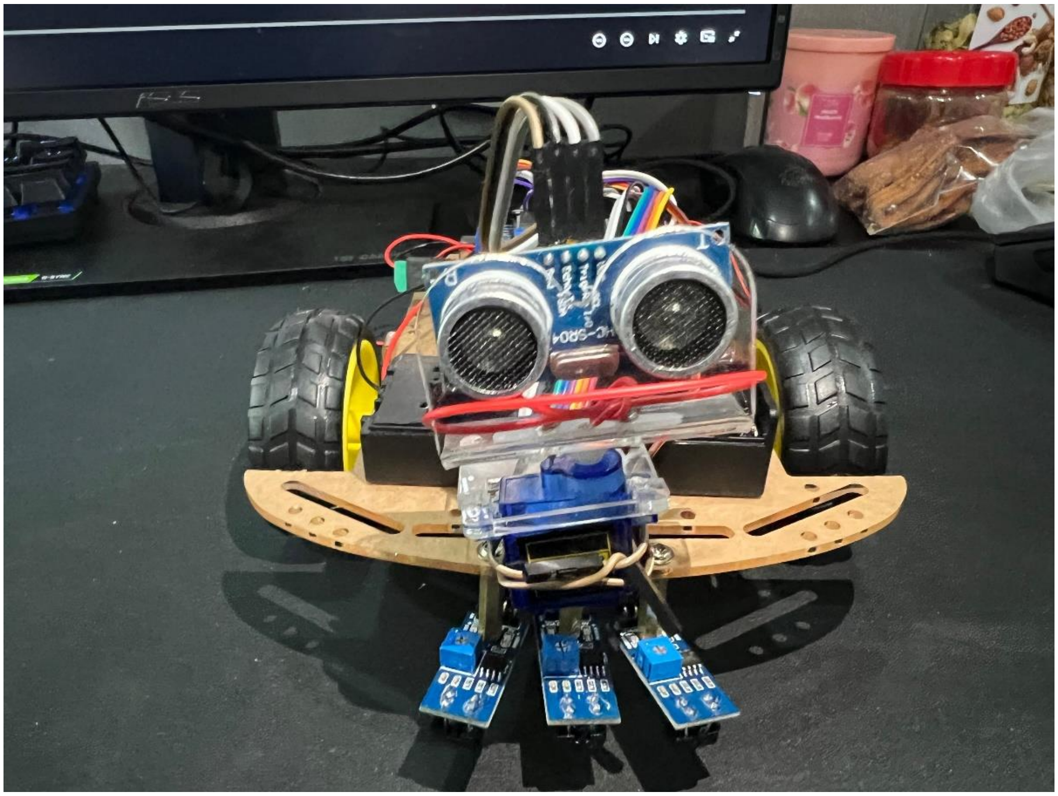
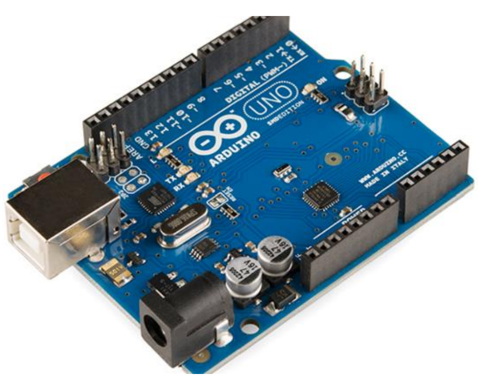
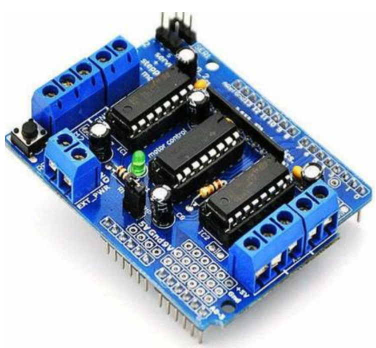
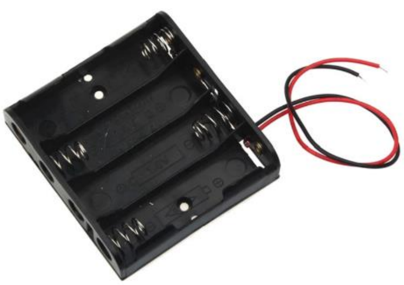
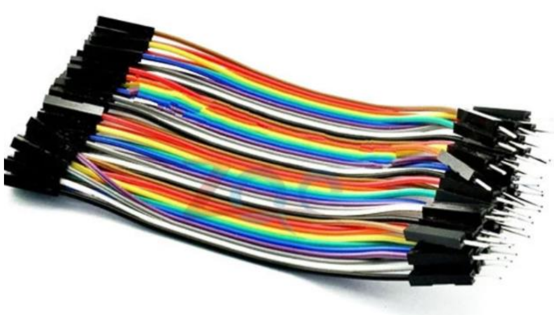


Figure 4: Above the car

Component	Quality	Price	Cost analysis
	1	6.18 USD =145.000 VND	The Arduino Uno is easy to use for beginning. It can run on multiple platform: Mac, Windows, and Linux. More sophisticated pre-assembled Arduino modules can be purchased and are modestly priced. The Arduino software is free and open source. The programming platform is based on the popular Wiring language. Arduino allows users a simple pathway to creating interactive objects that can take input from switches and sensors, and control physical

			outputs like lights, motors, or actuators
	1	3.2 USD =75000 VND	Two on-board L293Ds provide bidirectional control of 4 brush motors. Motor Driver Shield L293D provides up to 0.6A continuous output current, up to 1.2A maximite Driver Shield L293D supports 4.5V to 36V.2 ports connect an external power supply, to provide separate motor logic. Two 5V Servo motor terminals are connected to the high-resolution Arduino dedicated timer -no flicker. 4 motors can be driven DC brush or 2 stepper or 2 Servo motors. Up to 4 bidirectional DC motors with simple 8-bit speed selection. Pull resistance low to ensure motor is disabled during process. Power on and Arduino Restart Button
	1	2 USD =47000 VND	This is a plastic geared micro servo motor for RC, with a torque of 1.80kg.cm at 4.8V. It is a perfect solution for students' robotic projects like assembling robotic arms. The advantage of the RC servo motor over the DC brush motor is its ability to control the rotation angle. The Micro Servo is cheap and easy access. The Micro Servo can hold and move the ultrasonic to scan the obstacle.

	1	1.15 USD= 27000 VND	A device for measuring object distance at a low cost, this ultrasonic sensor HC-SR04 is a perfect choice.
	2	7.76 USD= 180000 VND	Geared DC Motors & Wheels are cheap and suitable for the Motor Driver Shield L293D. Geared DC Motors & Wheels can operate the car smoothly, flexible and safe power
	1	0.43 USD= 10000 VND	9-12V Battery Pack can provide enough power to operating for the entire system.
	1 pack M M 1 pack M F 1 pack F F 1 pack = 40 pcs 20 cm	16.5 USD = 387000 VND	Ribbon cable connect the entire patch together and transfer the signal from the Arduino, ultrasonic sensor

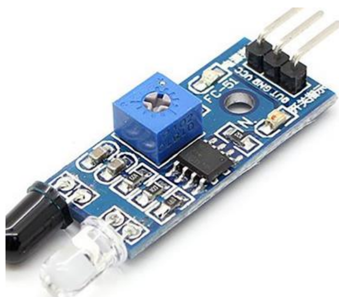
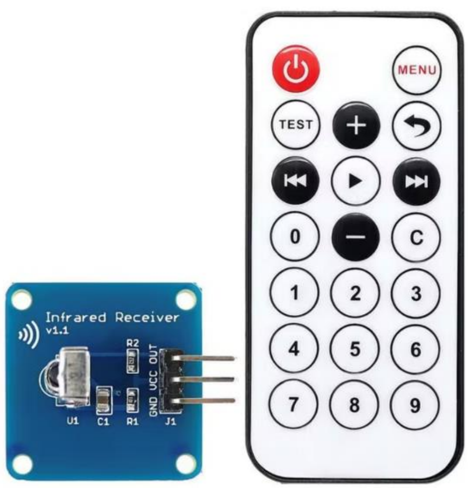
	3	1.28 USD= 30000 VND	The 4-pin IR sensor single line detector detect white and black lines. The circuit uses an infrared sensor TCRT5000 with a detection distance of 1~25mm, making it easy to install the module on the device. The sensor's sensitivity can be easily adjusted via the built-in rheostat on the board. The 4-pin line detection sensor circuit is suitable for devices that need to move along the line, the device detects white, black, ...
	1	3.38 USD = 76000 VND	IR remote transmitter and receiver can connect and control the car by Bluetooth. It cheap and flexible

Table 1: Component of car

4.Progress to completion

We have developed the best answers to each assignment in order to successfully finish tasks all the tasks on time. As of the present milestone of finishing tasks 1, 2 and 3, Team 10 has attained its ultimate objectives through trial-and-error techniques and systematic step-by-step testing.

A detailed explanation of the sorts of issues the team encountered during the robot design process is provided in the methodologies and practices that follow. The system is explained in the sections below.

4.1 Task: Object avoidance

1. Consideration

The obstacle identification in this exercise is the key challenge. The robot will respond and halt more effectively if the obstruction can be found as soon as feasible. This presents the scenario in which employees unintentionally cross the path of the robot and aims to guarantee worker safety by fostering a welcoming and secure work environment. We will make use of the Ultrasonic HC-SR04 for this work, as was mentioned in the proposal.

Figure 5: HC-SR04 sensor

Figure 6: Servo to support sensor rotate

4.2 Task: Following line

1.Consideration

In this task, the program needs to identify the line to follow. If the line moves to the left or right, the line sensor will send the signal to the code to turn the car to keep following the line. There are three light sensors in 3 directions: front, left, right. When the front sensor detects the presence of a black line by emitting infrared (IR) light and detecting the light levels that return to the sensor, the car will move forward with the limit speed. The car moves left and right when the sensor detects the black line. When both front and left sensor detect the black line, the car move in the same angle of the line, the same for front and right. If the light sensor cannot detect any black line, the car will move around to find the line to following.

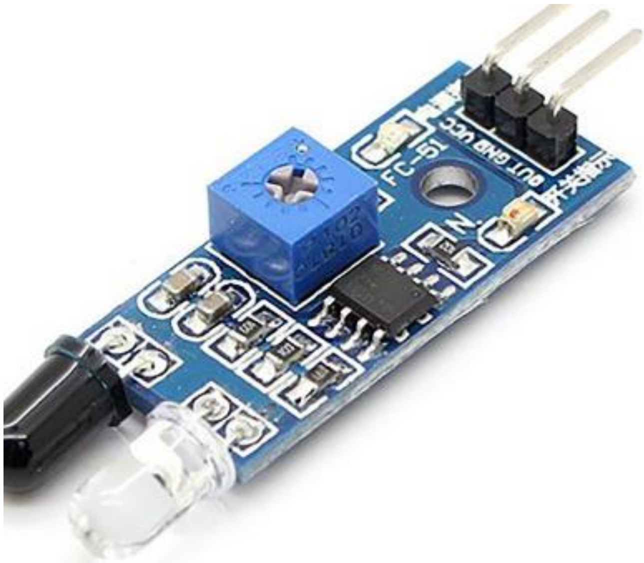


Figure 7: Line sensor

4.3 Task: Switching from automatic mode to manual mode and control the car by the remote

1.Consideration

This is the most complex task. The code not only combine task 1 and 2 but also change the car from manual mode to control mode. Using the module Bluetooth HC05 to control the car and change the car to manual the button function on the remote control. When the car turns on, the car will not run. The car will run task 1 when pressing button number 1. The same for line following and manual control.

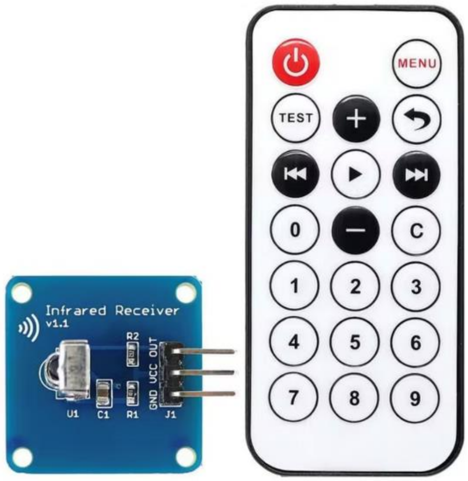


Figure 8: IR circuit and IR remote

5.Experiments, Testing, Finding

In this part of the project, we do 2 methods to shape a car using functions such as obstacle avoidance, line tracking and remote control. In order to create a frame that is strong and easy to repair, we have used plastic to save costs and easily change the shape.

In method 1, we use the technical toolkit of the Ministry of Education and Training in Vietnam to shape the car.

Figure 9: The technical toolkit of the Ministry of Education and Training in Vietnam

We made the shape of the car to use and attached 2 motors to the wheels. Everything was perfect until I found a place to mount the motor circuit. Next is the Arduino circuit and the power supply does not leave an area for mounting. Moreover, the balance of the vehicle is not stable. We had an additional dilemma that we couldn't find a way to attach the swivel wheel to the car, because the front long bar was not strong enough, it was not fixed by screws but only held by rubber.

Figure 10: Swivel wheel

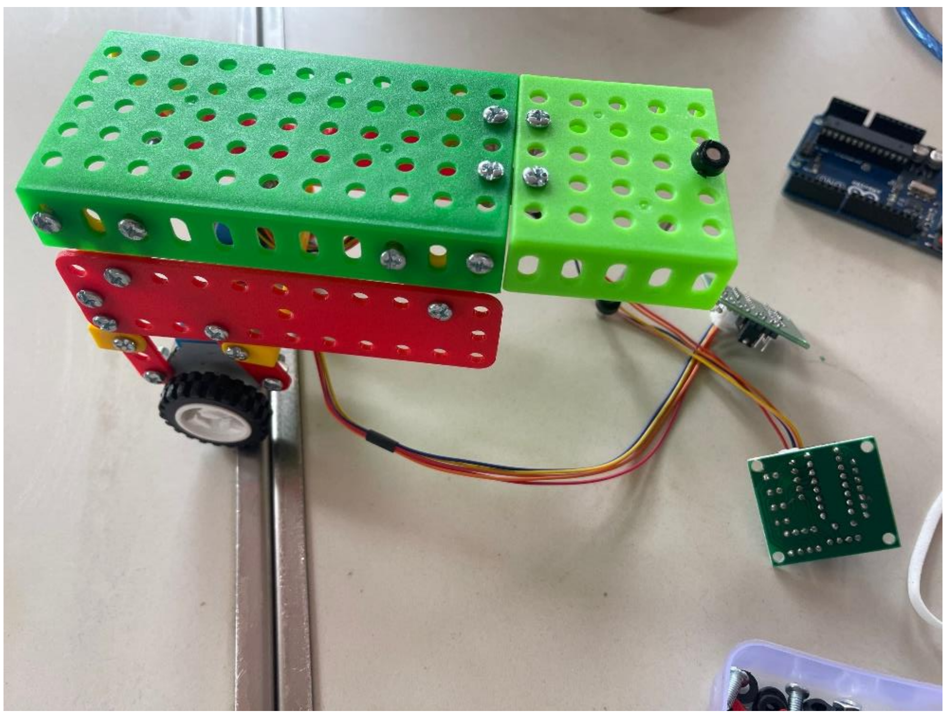


Figure 11: Car of the technical toolkit

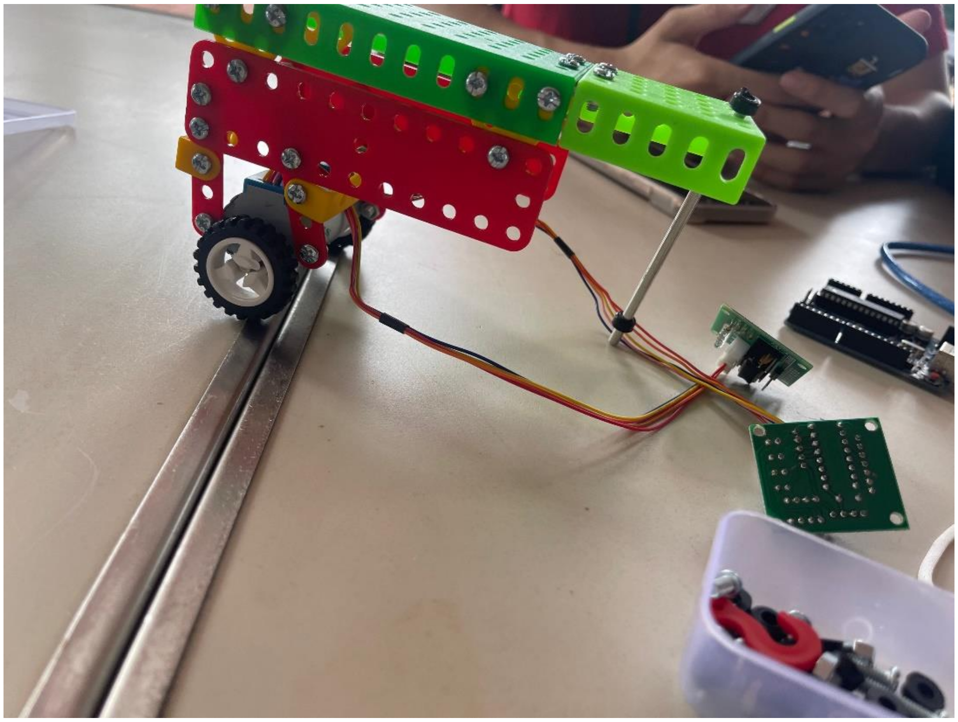


Figure 12: Car with 2 motor and without Arduino

Our option 2 is to use 3D printing with a larger area than option 1, the reason for using mica pads is to be able to choose the area you want to use, to be able to attach wires neatly, minimizing screws to connect. Then we swapped out the car engine with a higher capacity and a larger power source, to run the car with the circuitry.

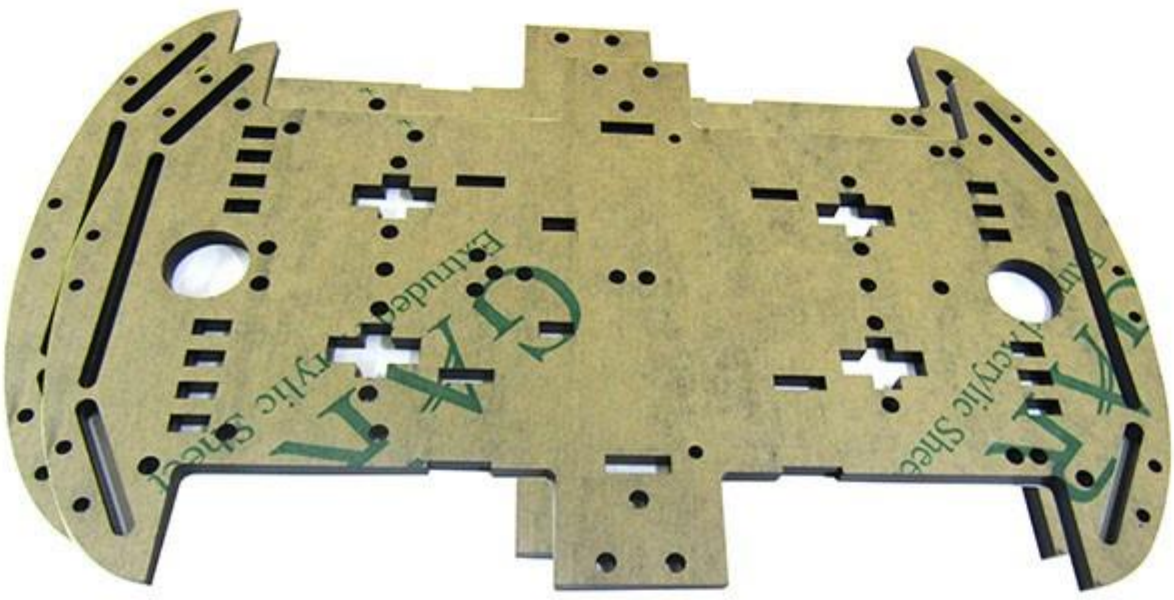


Figure 13: 3D printing (part of car – material: MICA)

After we used plexiglass to make the chassis, everything became easier. We then started mounting the circuit and mounting the motor. When everything was ready, we started to power the car with 2 high-capacity batteries, then soldered the wires to the car engine, fixed everything with screws. After wiring the circuit, the software department uses the code and Arduino to help the car operate according to the proposed function. That is to avoid obstacles.

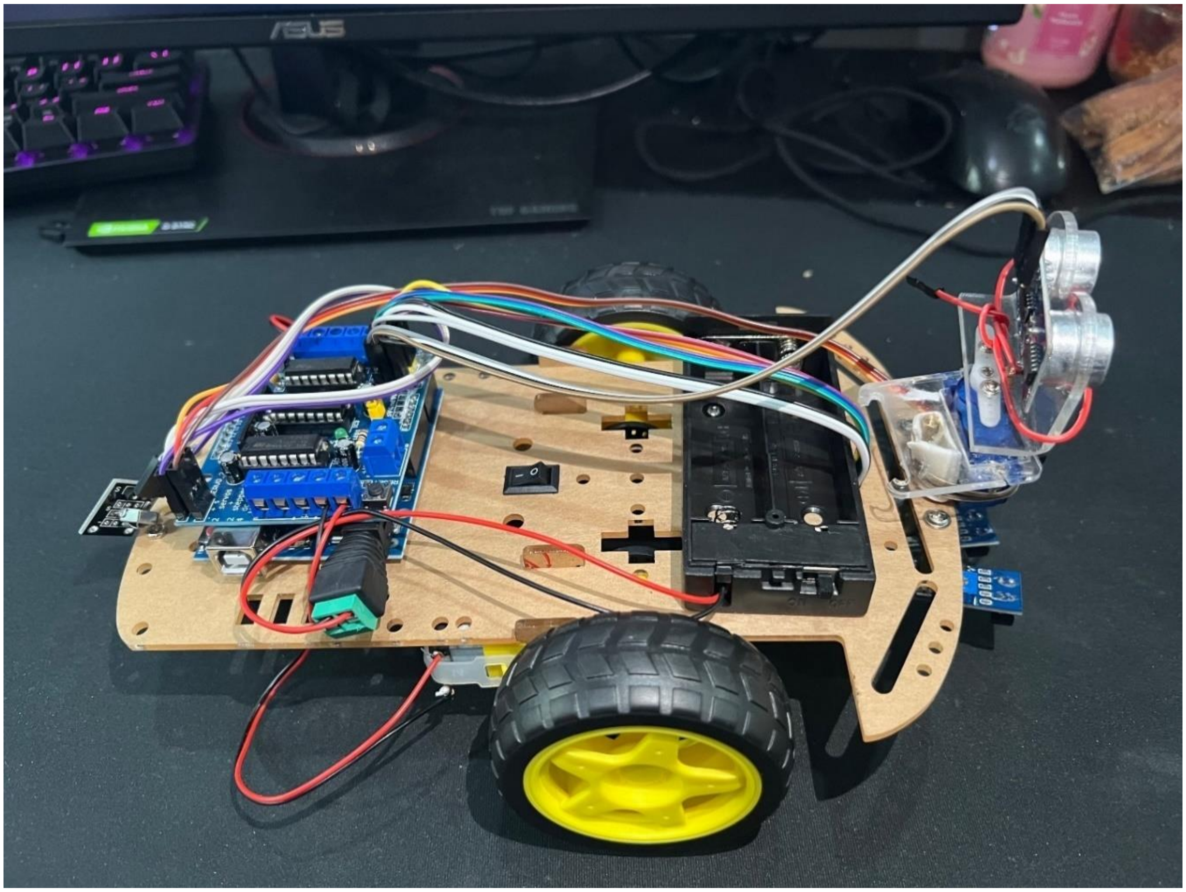


Figure 14: Final car with method 2

After trying 2 methods, we decided to use the 2 for later functions because the chassis is stable, making it easy for us to develop the functions for the car.

TESTING:

We have spent several days studying how the car works, and now we have produced proof of it.

The first:

This obstacle avoidance vehicle cannot run in rough environments because of the nature of the wheel at the rear. It shifts/displaces if bounced and it deflects the device, causing the vehicle to spin around and unable to continue the obstacle scanning system.

Figure 15:The first reason the car cannot work well

Second problem:

It is the engine of the wheel, when there is a faulty wheel, the car will go wrong. But for obstacle scanning, this error does not affect the results much. This error will affect the result of the following line function. Then there is a big influence on the result of that function.

Figure 16: The motor of the wheel is the second issue affecting the result

Third Problem:

Using the Bluetooth signal to control the car and switching between autonomic and manual mode, the code needs to define with button to change the mode. But using IR remote transmitter and receiver can be used in the short distance in five meters. Therefore, the IR remote needs to be that distance. If the IR remote transmitter wants to be used smoothly out of range, the transmitted need to boot up power four times. Also, the button in the remote need to define separately which not make any conflict inside the remote.

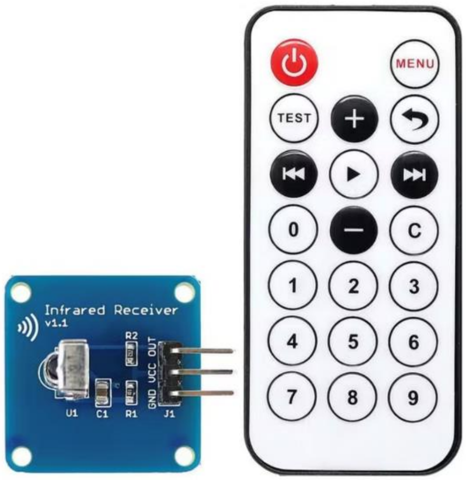


Figure 17: IR circuit and IR remote

Fourth problem:

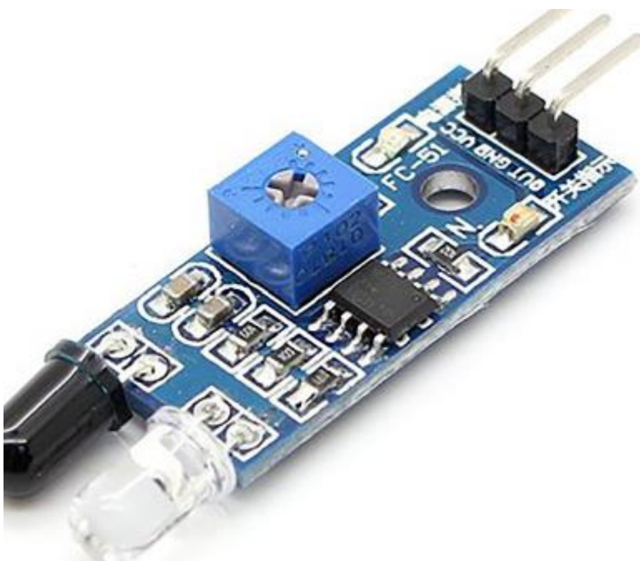


Figure 18: Line sensor

We implemented a 3-line sensor underneath the servo. The logical statement for line following is quite simple.

- If SL = HIGH && SM = HIGH && SR = HIGH
  - Car goes straight

- If SL = LOW && SM = HIGH && SR = HIGH
    - Car goes right
- If SL = HIGH && SM = HIGH && SR = LOW
    - Car goes left
- ELSE STOP

Where SL = Sensor Left ; SM = Sensor Middle; SR = Sensor Right

But the thing is that when our group completely set up all the codes and prepare for the run test, the problem of line following occurs. The sensors have detected the shadow as a black line themselves. Therefore, it always goes straight despite of we draw a black line. The reason why we realize this is because when we lift the car up (with small shadow), the car be able to detects the line to follow.

We try to fix this problem by placing a 3-line sensor higher. But the sensor cannot catch the light reflected.

Finding:

After finding out the above problems, we have an urgent meeting to find a quick and accurate solution.

For problem one, based on the car's design and performance. Some ideas are given to solve the problem that the car cannot run on rough roads, it is only suitable in flat environments. Because of the rotating nature of the equipment in the project.

The next problem, this may be a manufacturer error on the wheel's engine, the immediate fix is to change to another engine. If the car works normally after changing the engine, it proves that our solution is correct. If not, we need time to study the problem of the wheel engine and the chassis part

System Design Methodology/ Approach

Waterfall methodology is the development process that divides separate stages and sequential execution. The output of the first stage is the input of the next stage and no overlapping. Waterfall methodology will include 6 stages: requirement, design, implementation, verification or testing and maintenance. In this project, we are considering finishing at stage 5.

Final Testing and Evaluation

Phase 1: Requirement

Our project aims to make an autonomous car that avoids obstacles, line detector and remote control. Waterfall methodology is suitable for this project due to the non-change of the requirement throughout the project.

This project requires knowledge of software, electrical circuits, and object motion. Therefore, unit test will be applied to phase 1. To perform unit test, we will re-check each module separately. Then, module will be gradually integrated and thoroughly tested in order to ensure that the interface and interaction between the modules are smooth.

At this stage, the tests will be done on HC-06(Bluetooth transmission), SG90 Servo and Ultrasonic sensor via Arduino.

Test	Expected
Ultrasonic sensor interacts with SG90 Servo	While the autonomous car attaches obstacle, SG90 will force the ultrasonic sensor to move left and right to find the way to avoid the object. The data should be displayed on the terminal
IR Receiver	Successfully connection with Bluetooth on peripheral devices (phone, computer).
TCRT5000(Line detector)	Allow to detect the line color (Black and White) within 1~25mm (about 0.98 in). Data on the terminal should be displayed as 1 while catching the black line and 0 on the white line.
Motor	Motors with the wheels move in the same direction.

Table 2: Test Phase 1

Phase 2:

In this stage, the autonomous car is allowed to avoid obstacles. We test randomly place obstacles and expect the car to avoid smoothly.

Test	Expectation result	Reality
Avoidance Obstacle	Moving backward when the car roughly approaches the object(<24cm) and Servo forces ultrasonic sensor to move right and left to find the free to move.	The autonomous car is allowed to avoid the object.

Line Detector	TCRT5000 recognizes the black and white line	The TCRT5000 receives light color and displays on the terminal. The TCRT5000 and the Arduino are not matched.
Remote Control	The remote and IR receiver are interacting.	None

Table 3: Test Phase 2

6. Final System Design and Solution to Project aim

6.1 Object avoids

Because, using the same Arduino board for complete synchronization in the programming other components in the project. Therefore, Arduino IDE is the main platform to programming in our project. In this platform, C is the main language and logic for the project.

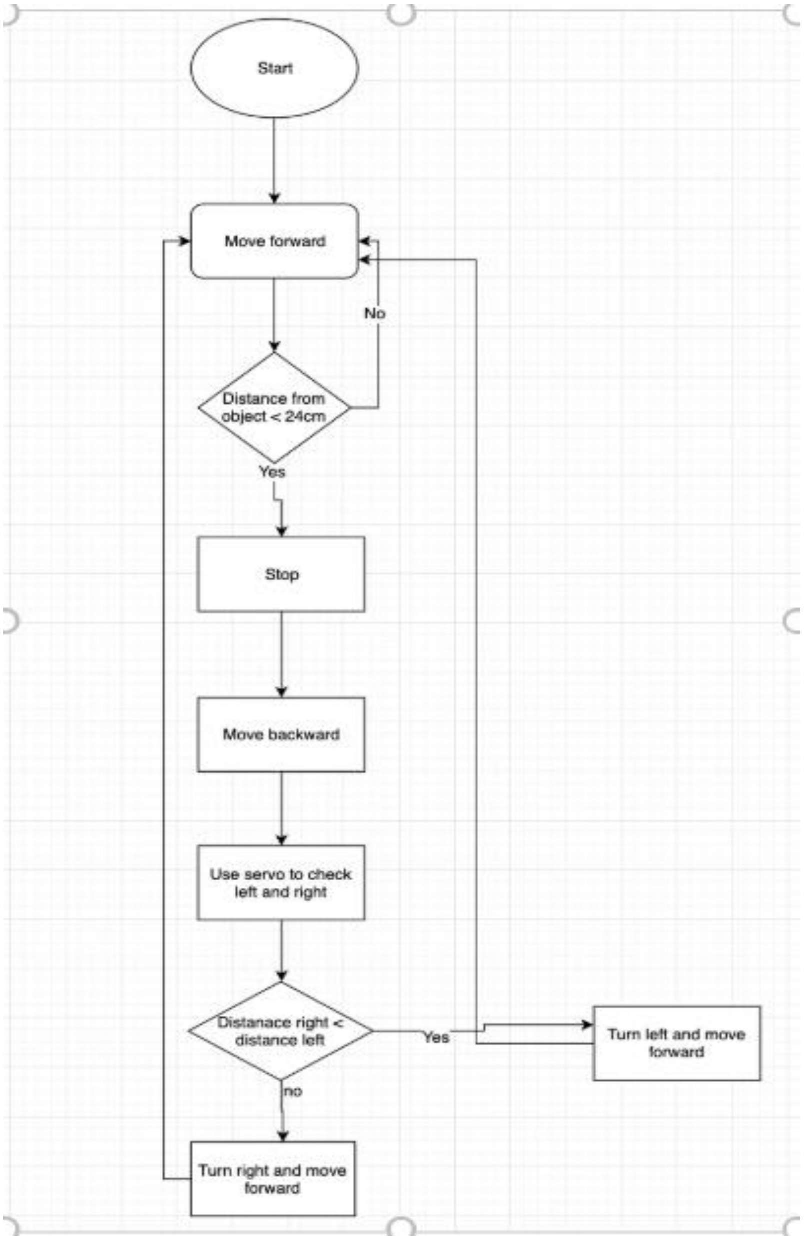
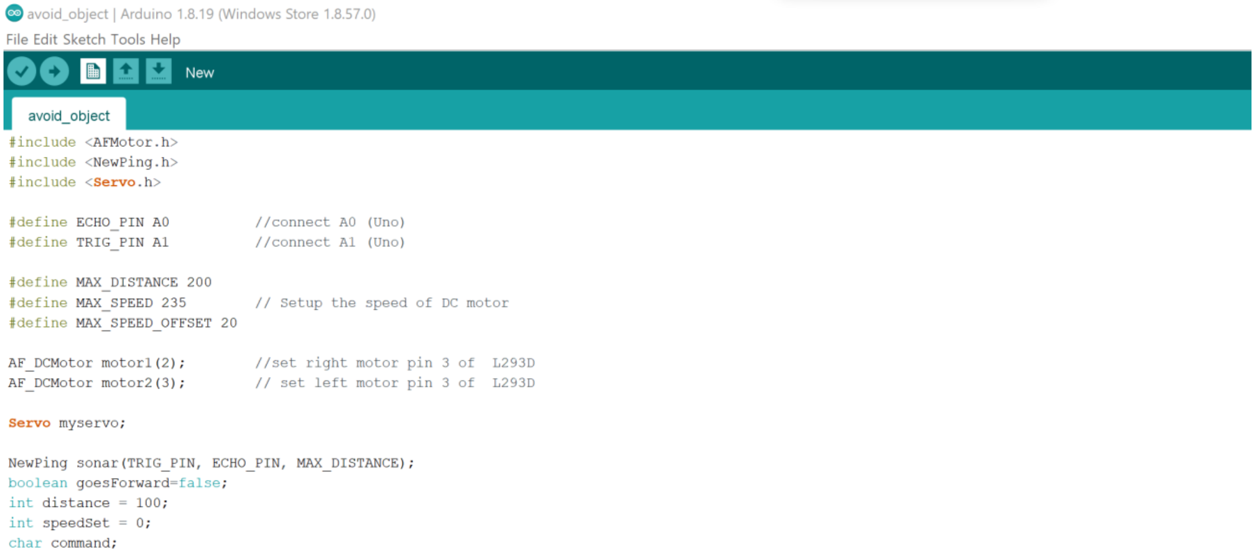


Figure 19: Flow chat of logic objective avoiding

Install AFMotor.h; NewPing.h;Servo.h libraries.

RMIT UNIVERSITY


avoid_object | Arduino 1.8.19 (Windows Store 1.8.57.0)

File Edit Sketch Tools Help

New

avoid_object

```
#include <AFMotor.h>
#include <NewPing.h>
#include <Servo.h>

#define ECHO_PIN A0          //connect A0 (Uno)
#define TRIG_PIN A1          //connect A1 (Uno)

#define MAX_DISTANCE 200
#define MAX_SPEED 235        // Setup the speed of DC motor
#define MAX_SPEED_OFFSET 20

AF_DCMotor motor1(2);        //set right motor pin 3 of  L293D
AF_DCMotor motor2(3);        // set left motor pin 3 of  L293D

Servo myservo;

NewPing sonar(TRIG_PIN, ECHO_PIN, MAX_DISTANCE);
boolean goesForward=false;
int distance = 100;
int speedSet = 0;
char command;
```

Figure 20: Code of objective avoiding [1] - Install AFMotor.h; NewPing.h;Servo.h libraries

In the first step of programing of project, the code needs to define all the pin of the device. In the code the ultrasonic have been defined in PIN A0 and the trigger the sonic in PIN A1. Then all valuable, which need to use in project, set into a valuable base on the calculate. In the code, the car speed has been set in 235. With this speed, the car can move fast but also the ultrasonic sensor still can detect the obstacles in front of the car.


avoid_object | Arduino 1.8.19 (Windows Store 1.8.57.0)

File Edit Sketch Tools Help

New

avoid_object

```
int dugme=1;
void setup()
{

    myservo.attach(9);
    myservo.write(95);         // Setting the angle of the SG90 :85,95,105,..so that the ultrasonic sensor is in line with the vehicle
    delay(1000)
    distance = readPing();
    delay(100);
    distance = readPing();
    delay(100);
    distance = readPing();
    delay(100);
    distance = readPing();
    delay(100);

    Serial.begin(9600);  //Set the baud rate to your Bluetooth module.
}
```

Figure 21: Code of objective avoiding (2)

To maintain the ultrasonic always point ahead to scan for the obstacles. The S90 program to rotate the sensor back to straight in any previous position.

avoid_object | Arduino 1.8.19 (Windows Store 1.8.57.0)

File Edit Sketch Tools Help

Save

avoid_object

```
    Serial.begin(9600);  //Set the baud rate to your Bluetooth module.
}
void loop(){
      long duration, distance;
           while(dugme==1)
           {
               int distanceR = 0;
               int distanceL =  0;
               delay(40);
               if(distance<=24)
               {
                    moveStop();
                    delay(100);
                    moveBackward();
                    delay(300);
                    moveStop();
                    delay(200);
                    distanceR = lookRight();
                    delay(200);
                    distanceL = lookLeft();
                    delay(200);
                    if(distanceR>=distanceL)
                    {
                      turnRight();
                      moveStop();
                    }else
                    {
                      turnLeft();
                      moveStop();
                    }
               }else
               {
                forward();
               }
               distance = readPing();
           }
```

Figure 22: Code of objective avoiding (3)

When there is an obstacle in front of the car at 24 cm (about 9.45 in), the ultrasonic sensor sends the signal to the board and the car immediately. Then the car will move backward 1 cm (about 0.39 in) and stop, the S90m start rotating right and left. If both right and left have obstacles, the car will choose to turn the way that the obstacles are far.

avoid_object | Arduino 1.8.19 (Windows Store 1.8.57.0)

File Edit Sketch Tools Help

avoid_object

```
                distance = readPing();
          }
}
void forward()
{
  motor1.setSpeed(255); //Define maximum velocity
  motor1.run(FORWARD); //rotate the motor clockwise
  motor2.setSpeed(255); //Define maximum velocity
  motor2.run(FORWARD); //rotate the motor clockwise
}
void back()
{
  motor1.setSpeed(255);
  motor1.run(BACKWARD);
  motor2.setSpeed(255);
  motor2.run(BACKWARD);
}
void left()
{
  motor1.setSpeed(255);
  motor1.run(FORWARD);
  motor2.setSpeed(0);
  motor2.run(RELEASE);
}
void right()
{
  motor1.setSpeed(0);
  motor1.run(RELEASE);
  motor2.setSpeed(255);
  motor2.run(FORWARD);
}
void aheadLeft()
{
  motor1.setSpeed(255);
  motor1.run(FORWARD);
  motor2.setSpeed(125);
  motor2.run(FORWARD);
}
void aheadRight()
{
  motor1.setSpeed(125);
  motor1.run(FORWARD);
  motor2.setSpeed(255);
  motor2.run(FORWARD);
```

Figure:23 Code of objective avoiding (4)

The last step is setting up the car to move forward, backward, turn right and left. Because the car just has two motors moving forward. One wheel stops when one wheel is moving to turn the car left or right. The code needs to set up a delay for the wheel stop in the brief period and run extract the moment the car has turn. Also, if both left and right of the ultrasonic have the object in range 24 cm (about 9.45 in), the car needs to turn around and start scanning again. The code will have a flag if that sensor

6.2 Following line

First thing first, the code needs to define which gate or pin of the device or sensor which will connect to the Arduino. In the line following program, there are three light sensors in it. Also, there are an loop to define with sensor detect the line.

```
#include <AFMotor.h>

/************************************************/
AF_DCMotor motor1(2);
AF_DCMotor motor2(3);
/*************************Define Line Track pins***********************************/
const int SensorLeft   = 10;      //Left sensor input (A1)
const int SensorMiddle = 9;       //Midd sensor input (A2)
const int SensorRight  = 8;       //Right sensor input(A3)
int SL;    //Status of Left line track sensor
int SM;    //Status of Midd line track sensor
int SR;    //Status of Righ line track sensor

unsigned char old_SL,old_SM,old_SR;

void setup()
{

      pinMode(SensorLeft,  INPUT);    //Init left sensor
      pinMode(SensorMiddle, INPUT);   //Init Middle sensor
      pinMode(SensorRight, INPUT);    //Init Right sensor
}

void loop()
{
      SL = digitalRead(SensorLeft);
      SM = digitalRead(SensorMiddle);
      SR = digitalRead(SensorRight);
```

Figure 24: Code of following line (1)

Then, the code needs to create all the condition of the car when it track the line. There are three sensors: middle, left, right. Base on the which sensor detect the light, the car will adapt the function to keep the car right on track of the line. For example, if the both the middle and left detect the line, the car need to be move in centai angle not just turn left. By using the delay function, the car can move in the small angle which don't need to turn left.

```
oid loop()

      SL = digitalRead(SensorLeft);
      SM = digitalRead(SensorMiddle);
      SR = digitalRead(SensorRight);

      //right fast 20 200
      if (SM == HIGH && SR == HIGH && SL == LOW)
      {
            goRight(160,40);delay(100);
      }
      if (SM == HIGH && SR == LOW && SL == HIGH)
      {
            goLeft(40,160);delay(100);

      }
      if (SM == HIGH && SR == LOW && SL == LOW)
      {
            forward(160,160);
      }
      if(SM == LOW && SR == LOW && SL == HIGH)
      {
            goLeft(200,80);delay(100);
      }
      if(SM == LOW && SR == HIGH && SL == LOW)
      {
            goRight(80,200);delay(100);
      }
      else
      {
```

Figure 25: Code of following line (2)

6.3 Switching from manual to control

The # button is the manual control mode selection button.

After switching to the remote-control mode, the car operates based on the buttons that have been pre-installed. According to the control panel, we include the forward button, the back button, the left and right button, and the OK button (understood to stop).

When we select the forward key, the car will move forward. But when we press the left turn button while the car is going straight, the car will only turn left (in a circle). And vice versa in other directions.

For the car to turn left about 30 degrees, we choose the straightforward button, then the left turn button and finally the straight run button (in an extremely brief time). An example is in the clip section.

Figure 26: IR Remote

In this mode, the signal part of the sensor is the most important thing, when the car is out of the control signal, we cannot control the car. It will stay at the last joystick until the signal is captured again. For example, they let the car go straight, the control range of the sensor is 20cm. If the car goes out 20cm, the car will still go straight even though we press stop, the car will still go straight.

Therefore, the signal of the remote control and the sensor mounted on our vehicle is 20cm away to be able to control. Moreover, car change the mode program, we will press the (*) button to perform line detection and (0) to avoid the vehicle automatically.

7. Conclusion

This project requires a skill of researching and understanding of both software and hardware. Besides that, plugging the correct pins into Arduino and writing the function that make the circuits work is a tough challenge. The final design project is expected to run smoothly while switching between manually and autonomously mode.

After this project, we hope to extend the knowledge more about the Embedded System, Drawing Design Prototype (Cad), develop coding skills.

8.References

[1]"Arduino UNO R3 SMD chip dán (kèm cáp)", *Nshopvn.com*, 2022. [Online]. Available: https://nshopvn.com/product/arduino-uno-r3-smd-chip-dan-kem-cap/. [Accessed: 18- Aug- 2022].

[2]"Đế module cảm biến siêu âm HC-SR04, HC-SR05", *Nshopvn.com*, 2022. [Online]. Available: https://nshopvn.com/product/de-module-cam-bien-sieu-am-hc-sr04-hc-sr05. [Accessed: 18- Aug- 2022].

[3]"Động cơ servo SG90 180 độ", *Nshopvn.com*, 2022. [Online]. Available: https://nshopvn.com/product/dong-co-servo-sg90-tower-pro. [Accessed: 18- Aug- 2022].

[4]"Dây Cắm Test Board Đực Cái 30cm (1 sợi)", *Nshopvn.com*, 2022. [Online]. Available: https://nshopvn.com/product/day-cam-test-board-duc-cai-30cm-1-soi. [Accessed: 18- Aug- 2022].

[5]"Hộp Pin 18650 2 cell có nắp và công tắc", *Nshopvn.com*, 2022. [Online]. Available: https://nshopvn.com/product/hop-pin-18650-2-cell-co-nap-va-cong-tac. [Accessed: 18- Aug- 2022].

[6]"Pin cell 18650 2000mAh", *Nshopvn.com*, 2022. [Online]. Available: https://nshopvn.com/product/pin-cell-18650-2000mah. [Accessed: 18- Aug- 2022].

[7]"Cảm Biến Siêu Âm HC-SR04", *Nshopvn.com*, 2022. [Online]. Available: https://nshopvn.com/product/cam-bien-sieu-am-hc-sr04. [Accessed: 18- Aug- 2022].

[8]"Jack DC đực có dây", *Nshopvn.com*, 2022. [Online]. Available: https://nshopvn.com/product/jack-dc-duc-co-day. [Accessed: 18- Aug- 2022].

[9]"NHẬN CẮT LẺ MICA THEO YÊU CẦU - MICA trong/trắng, độ dày 2mm, 3mm, 5mm - Tấm 21x30cm, 3mm,Trong | Totochan Decoration | Tiki", *tiki.vn*, 2022. [Online]. Available: https://tiki.vn/nhan-cat-le-mica-theo-yeu-cau-mica-trongtrang-do-day-2mm-3mm-5mm-tam-21x30cm-3mmtrong-

p141056780.html?spid=141056798&utm_source=google&utm_medium=cpc&utm_campaign=SEA_NBR_GGL_SMA_DTP_ALL_VN_ALL_UNK_UNK_C.ALL_X.10625984613_Y.106657549322_V.141056798_W.c_A.1687322216378_O.UNK&gclid=EAIaIQobChMIv5L0gvXP-QIVQqmWCh0a3AdREAYYASABEgK5WPD_BwE. [Accessed: 18- Aug- 2022].

[10]"TeamGantt", *Prod.teamgantt.com*, 2022. [Online]. Available: https://prod.teamgantt.com/gantt/schedule/?ids=3190168#&ids=3190168&user=&custom=&company=&hide_completed=false&date_filter=&color_filter=. [Accessed: 18- Aug- 2022].

9. Appendices.

Method	**Platform**	**Arrangement**	**Description**
Face-to-face	RMIT University RMIT UNIVERSITY	Location: Room B8.01.01 at RMIT University Saigon South Campus. Duration: 1.5 – 2.5 hour Regularity: 2 times per week Preparation: Online docs, Word App, Project Source. Note: Online docs	Face-to-face meeting is where we bring out our personal ideas, from which we select the best ideas to implement. Then, the team leader assigns each person to find data about the idea. This is the part to help each team member better understand the idea to be implemented. Finally, we gather in Online Docs, where my team takes notes and saves important information and sources. Information about the Face-to-face meeting: Attendance: Full members Agreement rate: 80% Looking for source Take notes. Solve all the confusing problems

Online	Microsoft Teams	Location: Flexible Duration: 1 – 1.5 hour Regularity: 2 times per week Preparation: Ideas Note: Online docs	Microsoft Teams is where we have online meetings, discussion conversations, and where we keep project information.
	Word		Word is the most important place where we do the project. Everything as important as information, images and project design is gathered in Word.
	Outlook		Outlook where we use to send invitations to join Microsoft Teams, Word and where we use to post. Moreover, this is the place to contact professors for questions.

Table 4: Communication of group

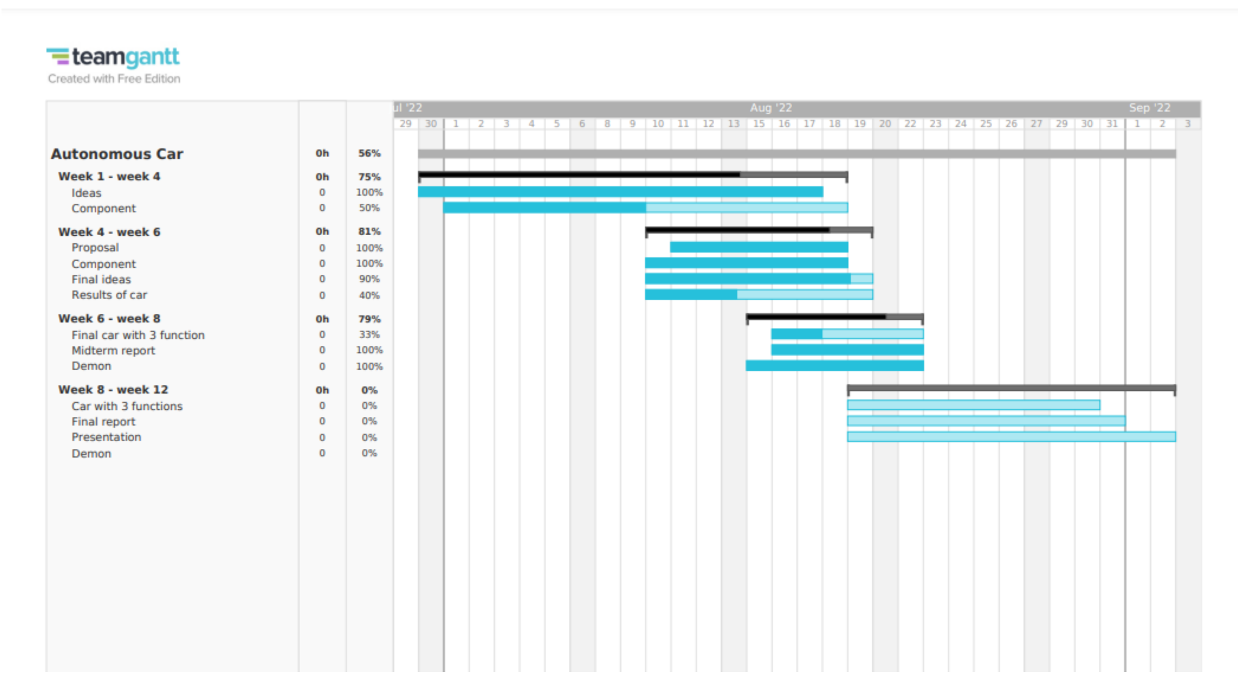


Figure 27: Gantt chart [10]

CODE:

```
#include <AFMotor.h>

#include <NewPing.h>
```

```
#include <Servo.h>


#define ECHO_PIN A0         //connect A0 (Uno)

#define TRIG_PIN A1         //connect A1 (Uno)


#define MAX_DISTANCE 200

#define MAX_SPEED 235       // Setup the speed of DC motor

#define MAX_SPEED_OFFSET 20


AF_DCMotor motor1(2);       //set right motor pin 3 of  L293D

AF_DCMotor motor2(3);       // set left motor pin 3 of  L293D


Servo myservo;


NewPing sonar(TRIG_PIN, ECHO_PIN, MAX_DISTANCE);

boolean goesForward=false;

int distance = 100;

int speedSet = 0;

char command;



int obstacles=1;

void setup()

{


  myservo.attach(9);
```

```
  myservo.write(95);        // Setting the angle of the SG90 :85,95,105,..so that the ultrasonic sensor is in line with the vehicle

  delay(1000)

  distance = readPing();

  delay(100);

  distance = readPing();

  delay(100);

  distance = readPing();

  delay(100);

  distance = readPing();

  delay(100);


  Serial.begin(9600);  //Set the baud rate to your Bluetooth module.

}

void loop(){

    long duration, distance;

        while(obstacles==1)

        {

            int distanceR = 0;

            int distanceL =  0;

            delay(40);

            if(distance<=24)

            {

                moveStop();

                delay(100);

                moveBackward();
```

```
            delay(300);
            moveStop();
            delay(200);
            distanceR = lookRight();
            delay(200);
            distanceL = lookLeft();
            delay(200);
            if(distanceR>=distanceL)
            {
              turnRight();
              moveStop();
            }else
            {
              turnLeft();
              moveStop();
            }
         }else
         {
          forward();
         }
         distance = readPing();
      }
}
void forward()
{
  motor1.setSpeed(255); //Define maximum velocity
```

```
  motor1.run(FORWARD); //rotate the motor clockwise
  motor2.setSpeed(255); //Define maximum velocity
  motor2.run(FORWARD); //rotate the motor clockwise
}
void back()
{
  motor1.setSpeed(255);
  motor1.run(BACKWARD);
  motor2.setSpeed(255);
  motor2.run(BACKWARD);
}
void left()
{
  motor1.setSpeed(255);
  motor1.run(FORWARD);
  motor2.setSpeed(0);
  motor2.run(RELEASE);
}
void right()
{
  motor1.setSpeed(0);
  motor1.run(RELEASE);
  motor2.setSpeed(255);
  motor2.run(FORWARD);
}
void aheadLeft()
```

```
{
  motor1.setSpeed(255);
  motor1.run(FORWARD);
  motor2.setSpeed(125);
  motor2.run(FORWARD);
}
void aheadRight()
{
  motor1.setSpeed(125);
  motor1.run(FORWARD);
  motor2.setSpeed(255);
  motor2.run(FORWARD);
}
void backRight()
{
  motor1.setSpeed(125);
  motor1.run(BACKWARD);
  motor2.setSpeed(255);
  motor2.run(BACKWARD);
}
void backLeft()
{
  motor1.setSpeed(255);
  motor1.run(BACKWARD);
  motor2.setSpeed(125);
  motor2.run(BACKWARD);
```

```
}
void Stop()
{
  motor1.setSpeed(0);
  motor2.run(RELEASE);
  motor2.setSpeed(0);
  motor2.run(RELEASE);
}
void stopAvoiding()
{
  obstacles=0;
  motor1.setSpeed(0);
  motor2.run(RELEASE);
  motor2.setSpeed(0);
  motor2.run(RELEASE);
}

int lookRight()
{
    myservo.write(50);
    delay(500);
    int distance = readPing();
    delay(100);
    myservo.write(105);
    return distance;
}
```

```
int lookLeft()
{
    myservo.write(160);
    delay(500);
    int distance = readPing();
    delay(100);
    myservo.write(105);
    return distance;
    delay(100);
}
int readPing() {
  delay(70);
  int cm = sonar.ping_cm();
  if(cm==0)
  {
    cm = 250;
  }
  return cm;
}
void moveStop() {
  motor1.run(RELEASE);
  motor2.run(RELEASE);
  }
void moveForward() {
if(!goesForward)
  {
```

```
    goesForward=true;
    motor1.run(FORWARD);
    motor2.run(FORWARD);
   for (speedSet = 0; speedSet < MAX_SPEED; speedSet +=2) // slowly bring the speed up to avoid loading down the batteries too quickly
   {
    motor1.setSpeed(speedSet);
    motor2.setSpeed(speedSet+MAX_SPEED_OFFSET);
    delay(5);
   }
  }
}
void moveBackward() {
    goesForward=false;
    motor1.run(BACKWARD);
    motor2.run(BACKWARD);
  for (speedSet = 0; speedSet < MAX_SPEED; speedSet +=2) // slowly bring the speed up to avoid loading down the batteries too quickly
  {
    motor1.setSpeed(speedSet);
    motor2.setSpeed(speedSet+MAX_SPEED_OFFSET);
    delay(5);
  }
}
void turnRight() {
  motor2.run(FORWARD);
  motor1.run(BACKWARD);
```

```
  delay(300);
  motor2.run(FORWARD);
  motor1.run(FORWARD);
}
void turnLeft() {
  motor2.run(BACKWARD);
  motor1.run(FORWARD);
  delay(300);
  motor2.run(FORWARD);
  motor1.run(FORWARD);
}
```