NATIONAL INSTITUTE OF BUSINESS MANAGEMENT
BSc (Hons) Computing |Year 2
HIGHER DIPLOMA IN SOFTWARE ENGINEERING – 21.1P

Robotics Application Development

Course Work

Automated Robot Car
("CHIPS")

Group Members
Haajara Yakoub COHDSE211P-001
Istina Suresh COHDSE211P-004
NS Kottachchi COHDSE211P-010
Peshala Nilmini COHDSE211P-011

Table of Contents

Automated Robot Car – “CHIPS”

Abstract

This report describes the construction of a robot named “CHIPS” that has three main functionalities i.e., line following, obstacle detecting and wireless control. The line following robot can autonomously navigate along both straight and curved path using IR sensors positioned at the bottom of the car which are used to sense the black line on a white surface. The sensors send the signals to the Arduino Uno microcontroller and accordingly four simple DC motors attached to the four wheels are used to move the robot left, right, forward or backward. An ultra-sonic sensor has been used to detect and respond to obstacles on its path to avoid collision. The Ultrasonic sensor sends the signals to the micro controller which then commands the DC motors to halt and makes a distinct beep sound using a buzzer upon detecting an obstacle. Further, a Bluetooth Module is used to receive signals from a smart phone to navigate the Robot Car along a user preferred path.

Introduction

An autonomous car is a vehicle capable of sensing its environment and operating without human involvement. A human passenger is not required to take control of the vehicle at any time, nor is a human passenger required to be present in the vehicle at all. An autonomous car can go anywhere a traditional car goes and do everything that an experienced human driver does. The Society of Automotive Engineers (SAE) currently defines 6 levels of driving automation ranging from Level 0 (fully manual) to Level 5 (fully autonomous). These levels have been adopted by the U.S. Department of Transportation. Autonomous cars thus rely on sensors, actuators, complex algorithms, machine learning systems, and powerful processors to execute software.

In the field of robotics, electronics, or electrical engineering, line-following robots are well known. Depending on the devices that are used, the complexity of the robot varies. Among the devices that can be part of the development of the robot, are the infrared (IR) sensors, the driver for the motors, the ultrasonic sensor, and the Arduino development platform etc.

Sensor based line follower robots are one of the most basic robot cars. It follows black line on white background or vice versa. These may be used in various industrial and domestic applications such as to carry goods, floor cleaning, delivery services and transportation, military applications, human assistance purposes etc. A line follower robot is based on the phenomenon of light. The robot will use two infrared proximity sensors to detect the line from the colours black and white, and on the basis of input received from the sensors, the Arduino will direct the motors to move with the help of a motor controller.

When programmed to avoid obstacles, it possesses the ability to detect the presence of an obstacle in its path through an ultrasonic sensor. Upon sensing an obstacle, the Robot Car is caused to come to a halt by the motors stopping motion. The alert system may work by changing the colour of the LED and emitting sounds through a buzzer. This design allows the robot to navigate in an unknown environment by avoiding collisions, which is a primary requirement for any autonomous mobile robot. This is particularly useful in automatic vacuum cleaners and smart vehicles etc.

Adding wireless control to such Robot Car enhances its capabilities. The Robot is controlled using a mobile phone instead of any other method like buttons, gesture etc. Here the phone uses an application that controls various motions of the Robot, such as direction, Start/ Stop etc. by communicating signals through a Bluetooth Module connected to the Arduino Uno micro controller.

This report discusses the creation and assembly of a robot car that possesses the above capabilities.

Material and Methods

Material used to create the Robot Car are as follows:

1. **Arduino Uno Micro Controller**- is an ATmega 328p Microcontroller based prototyping board. It is an open-source electronic prototyping platform that can be used with various sensors and actuators.
2. **IR Proximity sensors** x 2 –
3. **Ultra-Sonic Sensor** - It is an Ultrasonic Range Finder Sensor. It is a non-contact-based distance measurement system and can measure distance. The ultrasonic sensor has 4 pins: Vcc, Trig, Echo and Gnd.
4. **DC Motors 12V** x 4
5. **Servo Motor** - The Tower Pro SG90 is a simple Servo Motor which can rotate 90 degrees in each direction (approximately 180 degrees in total). It is used to rotate the Ultrasonic Sensor to scan for obstacles. It has three pins namely Control, VCC and GND.
6. **L293D Motor Driver** - It is a motor driver which can provide bi-directional drive current for two motors.
7. **DC Motor Switch**
8. **Gear Motor** x 4
9. **Wheels** x 4
10. **Chassis** x 2
11. **TP 5100** (Charging module)
12. **18650 Li-on Battery** x 2
13. **Battery Holder**
14. **Jumper Cables**
15. **Bluetooth Module**
16. **Red LED**

Method

The Robot Car is assembled using the above components.
The path on which the robot navigates for demonstration of the line following function is as follows:
The width of a line is 2- 5 cm

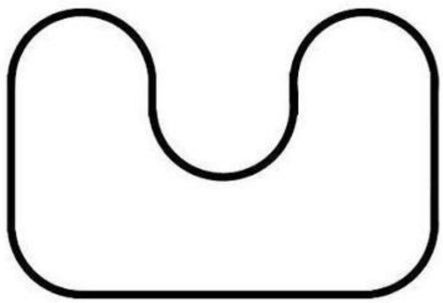
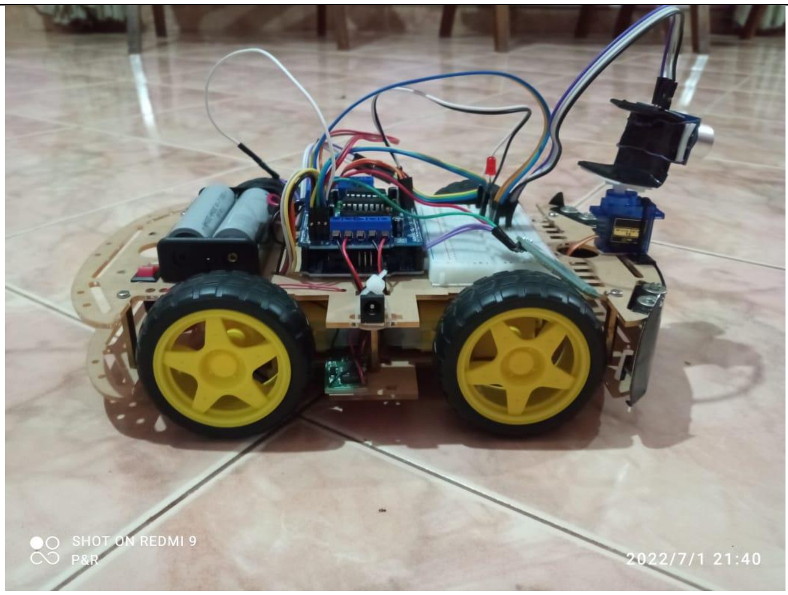
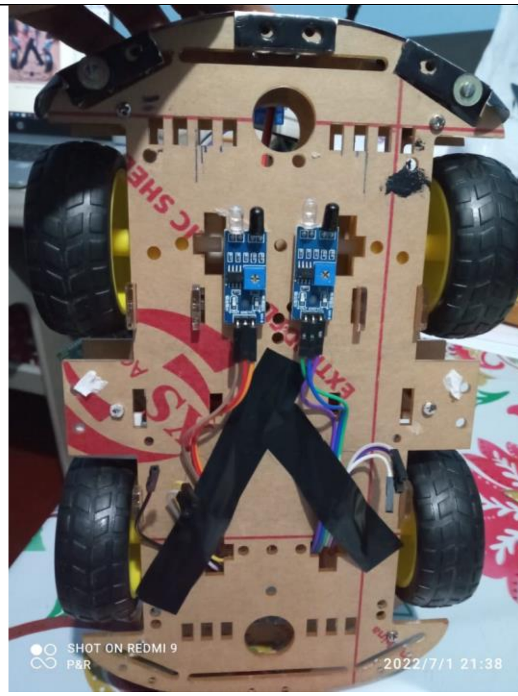
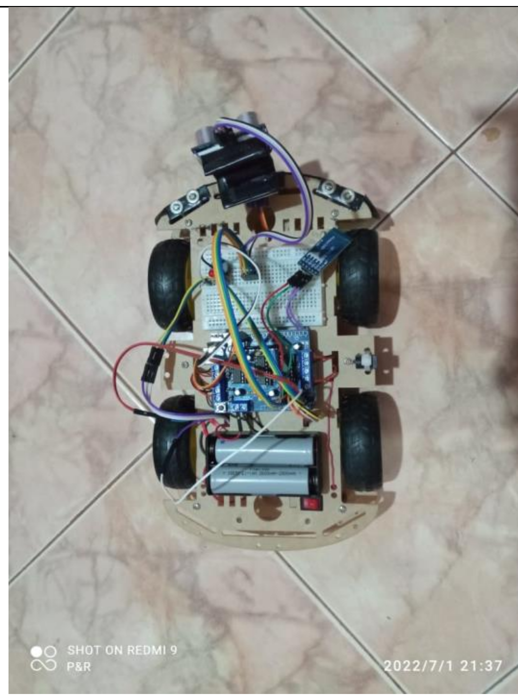


Figure 1

The project is constituted by a physical structure of two chassis that sandwich the electronic components that will process the signal to make the robot move according to the path it must follow. The following figures illustrate different views of the Robot upon assembly.

View	Photo
Front View of the Robot Car	 Figure 2
Side View of the Robot Car	 Figure 3

Bottom View of the Robot Car	 Figure 4
Top View of the Robot Car	 Figure 5

Results

Action	Photo
Straight Line following Results: i. Both bottom IR sensors sense black ii. The Motors cause the Robot Car to move forward	 **Figure 6**
Taking a right bend Results: i. The Robot Car slows down ii. The left IR sensor detects white while the right IR sensor detects black iii. The motors cause the Robot Car to turn right	 Figure 7

Taking a left bend Results: i. The Robot Car slows down ii. The left IR sensor detects black while the right IR sensor detects white iii. The motors cause the Robot Car to turn left	 Figure 8
Detecting an obstacle: Results: i. The Robot Car stops ii. The Red LED lights up iii. The Buzzer sounds	 Figure 9

Controlling navigation through the smartphone App using Bluetooth Results: i. The Robot moves according to the direction specified by the user through the App.	 Figure 10

Discussion

This robot uses a closed-loop system to follow the track accurately.

IR sensors are mounted on the lower part of the robot which tracks the line and moves towards its destination according to the received input. Line sensors are active IR sensors. It consists of a transmitter and a receiver. The transmitter is an IR source, which emits IR rays continuously and when these rays collide with an object, these rays return from there to the module. It always emits IR rays to the direction it is pointing to. When the IR rays hit a surface, some rays will be reflected back depending upon the color of the surface. Darker the color is, more IR will be absorbed by the surface and lesser IR rays will be reflected back. These reflected rays are sensed by the IR receiver and depending upon the received IR rays, the resistance of the receiver varies which will, in turn, varies the output voltage.

For the line follower sensor, we are using black tape and we have patched it onto the reflective surface. When the IR light falls on the black tape, the IR rays will be absorbed from the surface and if those rays fall on the reflective surface, then it will reflect back to the sensor and in this way, by calculating the reflected signals we will decide whether the robot is on the track or not.

The Robot car navigates on a straight line when both the IR Sensors at the bottom of the car detects the black line. When both sensors see white both wheels spins at same speed and robot goes straight. When “Left” sensor sees black and “Right” sensor sees white robot moves left. When “Right” sensor sees black and “Left” sees white robot moves right. The below figure illustrates this movement.

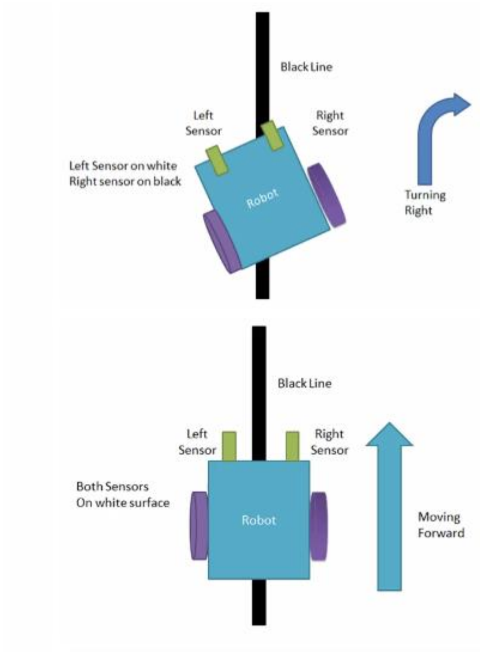


Figure 11

The basic principle behind the working of ultrasonic sensor is to note down the time taken by sensor to transmit ultrasonic beams and receiving the ultrasonic beams after hitting the surface by the receiver echo pin. Then further the distance is calculated using the formula. These signals are then delivered to the micro-controller which detects the presence of the object based on the instruction the programmer has stored in the program memory. This is used to avoid collision with obstacles.

This obstacle avoiding line following robot is driven by four gear motors powered by two lithium batteries and controlled by L293D Motor Shield and Arduino Uno with two IR Sensors and an Ultrasonic Sensor. If the ultra-sonic sensor detects an obstacle under 15 cm distance ahead while the vehicle moving along the black line, the vehicle stops and beeps while a Red LED lights up indicating the obstruction.

In this line track robot, it uses the HC-05 Bluetooth Module. This module receives input from the smartphone and functions according to commands received from the user. Accordingly, the smart phone will have an application which will enable the user to manipulate the robot by turning the robot left/ right or forward/ backward using icons on the mobile application. The signals transmitted from the phone will be received by the Bluetooth module which will then transmit to the Arduino Uno to control the DC motors to rotate accordingly to navigate the robot. This achieves wireless control of the robot through the smart phone.

Acknowledgement

Like every group project, working together brought us together in many ways and for that we thank each other for the friendship and compromises and not giving up on each other during these tough times.

The completion of this project could not have been possible without the participation and assistance of a lot of individuals contributing in numerous ways.

Our special thanks to Mr. Asanka Amarasinghe for sharing knowledge and experiences and the guidance and support from the start.

We also thank all our family members and friends who supported us in numerous ways to make this project a success.

References

How to Make Line Follower Robot Using Arduino, Aarav G, accessible on https://www.instructables.com/Line-Follower-Robot-Using-Arduino-2/

Synopsys, What is an Autonomous Car?, accessible on https://www.synopsys.com/automotive/what-is-autonomous-car.html

Bluetooth controlled car. (n.d.). Arduino Project Hub. Retrieved July 1, 2022, from https://create.arduino.cc/projecthub/samanfern/bluetooth-controlled-car-d5d9ca

Line follower robot. (n.d.). Arduino Project Hub. Retrieved July 1, 2022, from https://create.arduino.cc/projecthub/saher-iqbal/line-follower-robot-36516b

Obstacle avoiding car. (n.d.). Arduino Project Hub. Retrieved July 1, 2022, from https://create.arduino.cc/projecthub/adam/obstacle-avoiding-car-a192d9

Appendices

Appendix 1: Simple line following mode

```
#include <AFMotor.h>

#define irPinML A4
#define irPinMR A2
AF_DCMotor FrontL(1);
AF_DCMotor BackL(2);
AF_DCMotor BackR(3);
AF_DCMotor FrontR(4);

void setup(){
 Serial.begin(9600);
 pinMode(irPinML ,INPUT);
 pinMode(irPinMR ,INPUT);


 FrontL.setSpeed(150);

 FrontR.setSpeed(150);
 BackL.setSpeed(150);
 BackR.setSpeed(150);
}

void moveFront()
{
 FrontL.run(FORWARD);
 FrontR.run(FORWARD);
 BackL.run(FORWARD);
 BackR.run(FORWARD);
 }
void turnLeft()
{
 FrontL.run(BACKWARD);
 FrontR.run(FORWARD);
 BackL.run(BACKWARD);
 BackR.run(FORWARD);
 }
```

```
void turnRight()
{
 FrontL.run(FORWARD);
 FrontR.run(BACKWARD);
 BackL.run(FORWARD);
 BackR.run(BACKWARD);
 }
 void moveStop()
{
 FrontL.run(RELEASE);
 FrontR.run(RELEASE);
 BackL.run(RELEASE);
 BackR.run(RELEASE);

 }
void loop()
{
 int irML= digitalRead(irPinML);
 int irMR= digitalRead(irPinMR);
 Serial.print("left  is -");
 Serial.println(irML);
 Serial.print("Right is -");
 Serial.println(irMR);

  if(irML==HIGH && irMR==HIGH){
  moveFront();delay(2);
  }

  if(irML==HIGH && irMR==LOW)
{
  //adjust to left
  turnLeft();delay(75);
  moveFront();delay(3);
  }

 if(irML==LOW && irMR==HIGH){
 //adjust to right
 turnRight();delay(75);
 moveFront();delay(3);
  }
   if(irML==LOW && irMR==LOW){
```

```
 //stop the car
 moveStop();delay(2);
  }
}
```

Appendix 2: An Obstacle detected

```
#include <AFMotor.h>

#define irPinML A4
#define irPinMR A2
#define echoPin A1
#define trigPin A0
#define bzzPin 9


long duration;
int distance;

AF_DCMotor FrontL(1);
AF_DCMotor BackL(2);
AF_DCMotor BackR(3);
AF_DCMotor FrontR(4);

void setup(){
  pinMode(bzzPin,OUTPUT);
  pinMode(trigPin, OUTPUT);
  pinMode(echoPin, INPUT);
  Serial.begin(9600);
  pinMode(irPinML ,INPUT);
 pinMode(irPinMR ,INPUT);


 FrontL.setSpeed(150);

 FrontR.setSpeed(150);
 BackL.setSpeed(150);
 BackR.setSpeed(150);
}

void moveFront(){
 FrontL.run(FORWARD);
 FrontR.run(FORWARD);
 BackL.run(FORWARD);
 BackR.run(FORWARD);
```

```
}
void moveBack(){
FrontL.run(BACKWARD);
FrontR.run(BACKWARD);
BackL.run(BACKWARD);
BackR.run(BACKWARD);
}
void turnLeft(){
FrontL.run(BACKWARD);
FrontR.run(FORWARD);
BackL.run(BACKWARD);
BackR.run(FORWARD);
}
void turnRight(){
FrontL.run(FORWARD);
FrontR.run(BACKWARD);
BackL.run(FORWARD);
BackR.run(BACKWARD);
}
void moveStop(){
FrontL.run(RELEASE);
FrontR.run(RELEASE);
BackL.run(RELEASE);
BackR.run(RELEASE);

}

void ultraSonic()
{
 digitalWrite(trigPin, LOW);
 delayMicroseconds(2);

 digitalWrite(trigPin, HIGH);
 delayMicroseconds(10);
 digitalWrite(trigPin, LOW);
 duration = pulseIn(echoPin, HIGH);
 distance = duration * 0.034 / 2;
 Serial.print("Distance: ");
 Serial.print(distance);
 Serial.println(" cm");
```

```
 }
void ringBuzzer()
{
 digitalWrite(bzzPin,HIGH);
 delay(100);
 digitalWrite(bzzPin, LOW);
 delay(100);
 digitalWrite(bzzPin,HIGH);
 delay(100);
 digitalWrite(bzzPin, LOW);
 delay(100);
 }
 void loop(){
  ultraSonic();
  if (distance<=10){
   moveStop();
   ringBuzzer();
   delay(100);
  }else{
   moveFront();
   delay(10);
  }
    int irML= digitalRead(irPinML);
int irMR= digitalRead(irPinMR);
Serial.print("left  is -");
Serial.println(irML);
Serial.print("Right is -");
Serial.println(irMR);

 if(irML==HIGH && irMR==HIGH){
 moveFront();delay(2);
 }

 if(irML==HIGH && irMR==LOW){
 //adjust to left
 turnLeft();delay(75);
 moveFront();delay(3);
 }

if(irML==LOW && irMR==HIGH){
//adjust to right
```

```
turnRight();delay(75);
moveFront();delay(3);
 }
  if(irML==LOW && irMR==LOW){
//stop the car
moveStop();delay(2);
 }

   }
```

Appendix 3: Bluetooth Mode

```
#include <AFMotor.h>

#define irPinML A4
#define irPinMR A2
AF_DCMotor FrontL(1);
AF_DCMotor BackL(2);
AF_DCMotor BackR(3);
AF_DCMotor FrontR(4);
char val;  // for Bluetooth

void setup()
{
 Serial.begin(9600);
 pinMode(irPinML ,INPUT);
 pinMode(irPinMR ,INPUT);


 FrontL.setSpeed(150);
 FrontR.setSpeed(150);
 BackL.setSpeed(150);
 BackR.setSpeed(150);
}

void moveFront()
{
 FrontL.run(FORWARD);
 FrontR.run(FORWARD);
 BackL.run(FORWARD);
 BackR.run(FORWARD);
 }
 void moveBack()
{
 FrontL.run(BACKWARD);
 FrontR.run(BACKWARD);
 BackL.run(BACKWARD);
 BackR.run(BACKWARD);
 }
void turnLeft()
```

```
{
 FrontL.run(BACKWARD);
 FrontR.run(FORWARD);
 BackL.run(BACKWARD);
 BackR.run(FORWARD);
 }
void turnRight()
{
 FrontL.run(FORWARD);
 FrontR.run(BACKWARD);
 BackL.run(FORWARD);
 BackR.run(BACKWARD);
 }
 void moveStop()
{
 FrontL.run(RELEASE);
 FrontR.run(RELEASE);
 BackL.run(RELEASE);
 BackR.run(RELEASE);

 }
void loop()
{
if(Serial.available())
{
  val=Serial.read();
  Serial.print(val);
  }
  switch(val)
{
    case 'F':
      moveFront();
      delay(1000);
      moveStop();

      break;
    case 'B':
      moveBack();
      delay(800);
      moveStop();
      break;
```

```
  case 'R':
   turnRight();
   delay(500);
   moveStop();
   break;
  case 'L':
   turnLeft();
   delay(500);
   moveStop();
   break;
  default:
   moveStop();
  }
  val='S';
}
```