Ahsanullah University of Science & Technology

Department of Computer Science & Engineering

TUMBLER

[Android Bluetooth Car]

CSE 3216

Microcontroller Based System

Design Lab

Submitted By:

Mahmudur Rahman	16.01.04.120
Chowdhury Abida Anjum	16.01.04.126

Introduction:

This is a simple project of Android Bluetooth Car with Bluetooth control. To control the car used Android-device with a built-in accelerometer. Press forward - car goes forward, press to the left - car turns to the left, press back - car goes back.

Required Equipment:

Hardware:

- Arduino Uno:

  Arduino has been used in thousands of different projects and applications. The Arduino software is easy-to-use for beginners, yet flexible enough for advanced users. It runs on Mac, Windows, and Linux. Teachers and students use it to build low cost scientific instruments, to prove chemistry and physics principles, or to get started with programming and robotics. Designers and architects build interactive prototypes, musicians and artists use it for installations and to experiment with new musical instruments.

- Motor Driver:

  A motor driver is a small Current Amplifier whose function is to take a low-current control signal and then turn it into a higher-current signal that can drive a motor. The L293D is a typical Motor Driver which can drive 2 DC motors simultaneously. Motor Driver ICs are primarily used in autonomous robotics only.

- Bluetooth Module :

  The Bluetooth module HC-05 is a MASTER/SLAVE module. By default the factory setting is SLAVE. The Role of the module (Master or Slave) can be configured only by AT COMMANDS. The slave modules cannot initiate a connection to another Bluetooth device, but can accept connections. Master module can initiate a connection to other devices.

- Gear Motor:

  DC Gear motor or BO motor can be used to get the fast speed of the robot. If you know the PWM (Pulse Width Modulation) technique to control the speed of robot than high rpm motor can be used to get the fast speed of Vision LFR.

- Sonar Sensor :
  An Ultrasonic sensor is a device that can measure the distance to an object by using sound waves. It measures distance by sending out a sound wave at a specific frequency and listening for that sound wave to bounce back. By recording the elapsed time between the sound wave being generated and the sound wave bouncing back, it is possible to calculate the distance between the sonar sensor and the object.

- Wheels
- Jumper wire
- Breadboard
- Battery
- Bluetooth enabled android device

Software/Applications:

- Arduino Bluetooth controller
- Fritzing
- Arduino IDE

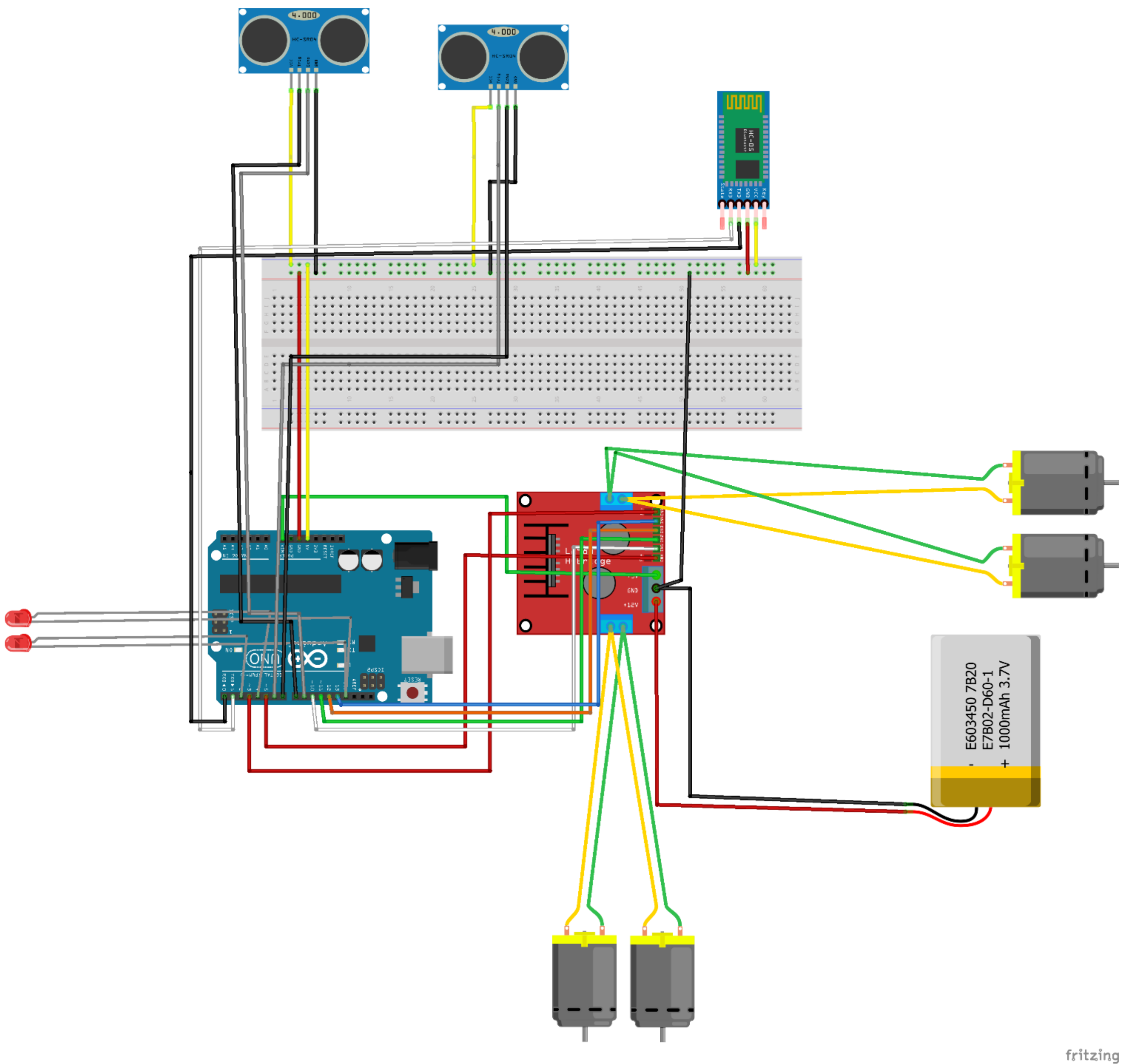
Features:

- Detect Obstacles
- Capable of taking various degrees of turns
- Controlled by ‘Android device’
- Long term battery life

Working principle:

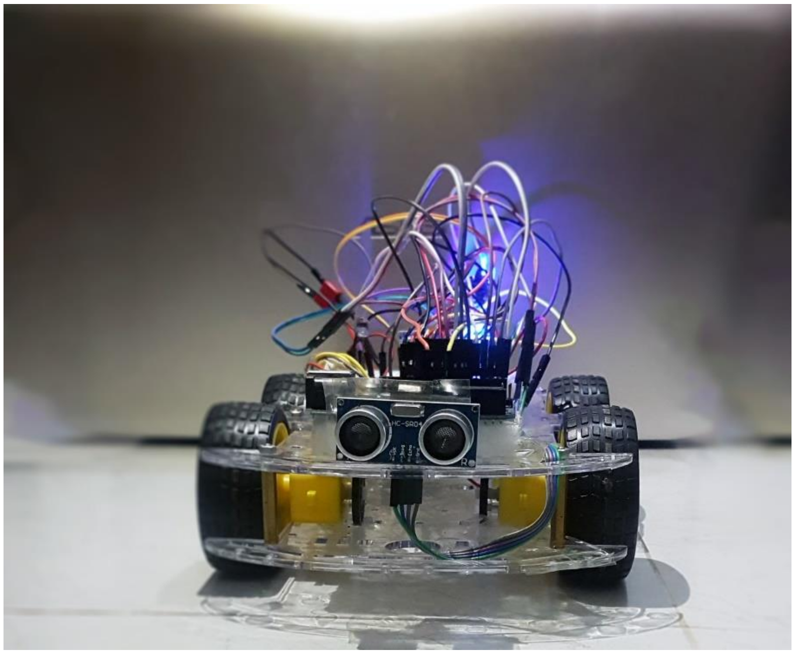
To control the car used Android-device with a built-in accelerometer. Press forward - car goes forward, press to the left - car turns to the left, press back - car goes back. If it finds an obstacle it stops its motion and goes in the direction where the android device directed.

Schematic diagram:

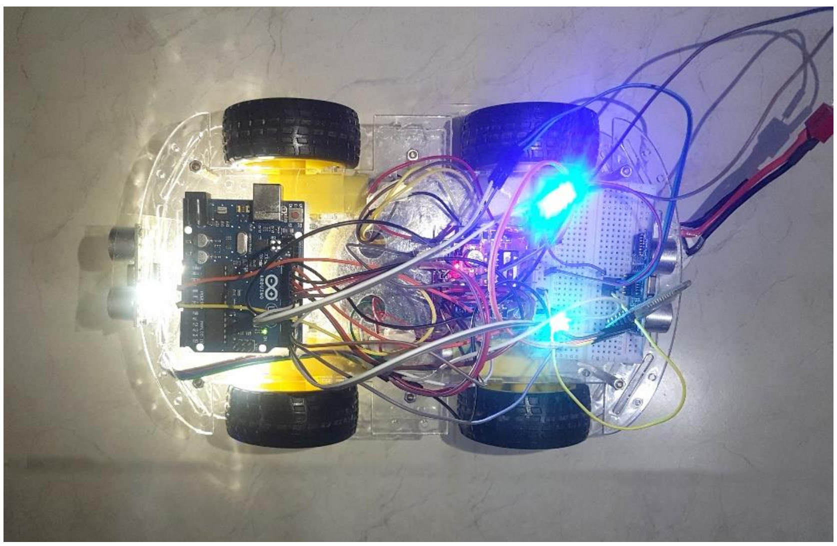


Figures of the project:

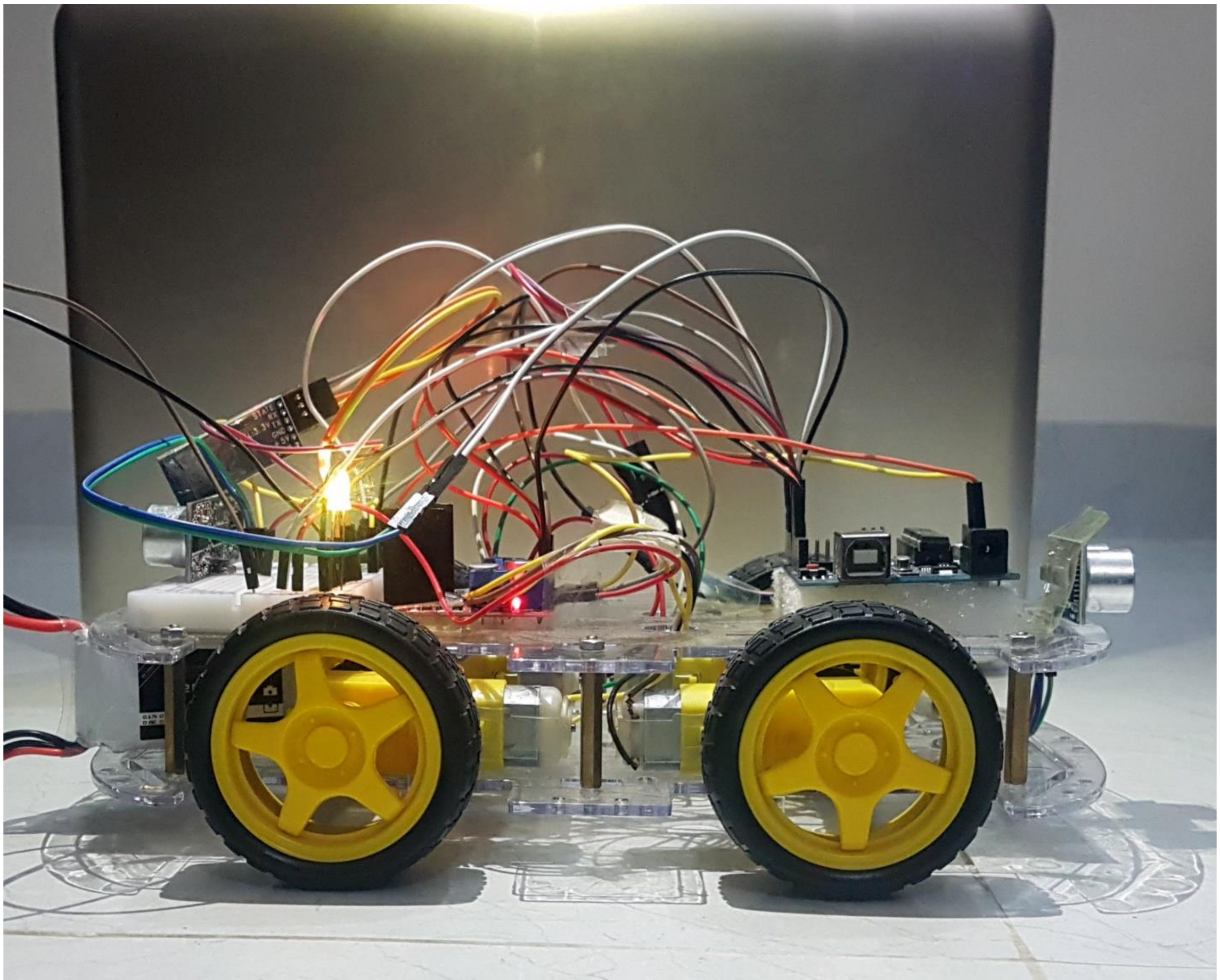
Front view



Top View



Side View



Constraints:

We proposed that our bluetooth control car will convert in remote control car which is not possible right now. We also want to set a camera with or car to do cinematography but connecting camera with Arduino is little bit difficult.

Dos' and DON'Ts':

Proposal	Yes	No
Make tight turns	√	
Bluetooth control	√	
Obstacle detection with Sonar sensor	√	
Indicator LEDs	√	

Conclusion:

The car will be developed as field efficient by using the structure with optimization in design and working principal.

Appendix:

```
char t;
const int ftrigPin = 8;
const int fechoPin = 9;
const int trigPin = 6;
const int echoPin = 7;
int ledL = 2;
int ledR = 4;
int enA = 5;
int enB = 3;
long duration;
int distance;
long fduration;
int fdistance;
void setup() {
  pinMode(13, OUTPUT);  //left motors forward
  pinMode(12, OUTPUT);  //left motors reverse
  pinMode(11, OUTPUT);  //right motors forward
  pinMode(10, OUTPUT);  //right motors reverse
  pinMode(enA, OUTPUT);
  pinMode(enB, OUTPUT);
  pinMode(ledL, OUTPUT);
  pinMode(ledR, OUTPUT);
  pinMode(trigPin, OUTPUT);
  pinMode(echoPin, INPUT);
  pinMode(ftrigPin, OUTPUT);
  pinMode(fechoPin, INPUT);
  Serial.begin(9600);
  Serial.setTimeout(10);
}
void fsonar()
{
  digitalWrite(ftrigPin, LOW);
  delayMicroseconds(2);

  digitalWrite(ftrigPin, HIGH);
  delayMicroseconds(10);
  digitalWrite(ftrigPin, LOW);
```

```
  fduration = pulseIn(fechoPin, HIGH);
  fdistance = fduration * 0.034 / 2;
  Serial.print("FDistance: ");
  Serial.println(fdistance);
}
void sonar()
{
  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.println(distance);
}
void stop()
{
    digitalWrite(13, LOW);
    digitalWrite(12, LOW);
    digitalWrite(11, LOW);
    digitalWrite(10, LOW);
}
void loop() {
sonar();
 fsonar();
  analogWrite(enA, 100);
  analogWrite(enB, 100);
  if (Serial.available()) {
    t = Serial.read();
    Serial.println(t);
  }
  //if((t  ==  'F'  ||  t  ==  'B')&&     (distance<30  ||
fdistance<30)) stop();
  if(t == 'F' && fdistance<30) stop();
```

```
else if(t == 'B' && distance<30) stop();

else if (t == 'F' && fdistance>=30) {
  digitalWrite(13, HIGH);
  digitalWrite(11, HIGH);
  digitalWrite(ledL, LOW);
  digitalWrite(ledR, LOW);
  analogWrite(enA, 100);
  analogWrite(enB, 100);
}
else if (t == 'B' && distance>=30) {
  digitalWrite(12, HIGH);
  digitalWrite(10, HIGH);
  digitalWrite(ledL, LOW);
  digitalWrite(ledR, LOW);
  analogWrite(enA, 100);
  analogWrite(enB, 100);
}
else if (t == 'L') {
  digitalWrite(11, HIGH);
  digitalWrite(13, LOW);
  digitalWrite(12, LOW);
  digitalWrite(ledL, HIGH);
  digitalWrite(ledR, LOW);
  analogWrite(enA, 85);
  analogWrite(enB, 85);

}

else if (t == 'R') {
  digitalWrite(13, HIGH);
  digitalWrite(11, LOW);
  digitalWrite(10, LOW);
  digitalWrite(ledR, HIGH);
  digitalWrite(ledL, LOW);
  analogWrite(enA, 85);
  analogWrite(enB, 85);
}
else if (t == 'W') {
```

```
    digitalWrite(9, HIGH);
  }
  else if (t == 'w') {
    digitalWrite(9, LOW);
  }
  else if (t == 'S') {
    digitalWrite(13, LOW);
    digitalWrite(12, LOW);
    digitalWrite(11, LOW);
    digitalWrite(10, LOW);
    digitalWrite(ledL, LOW);
    digitalWrite(ledR, LOW);
  }
  //delay(100);
}
```