Microprocessors Project

Obstacle avoiding car using IR sensor

Mohamed Munib Mohamed Badr 20011737

Mohamed Ayman EL-Saeed 20011519

Mohanad Ali Mahmoud 20012030

Mohamed Mahmoud Abdelfattah 19016498

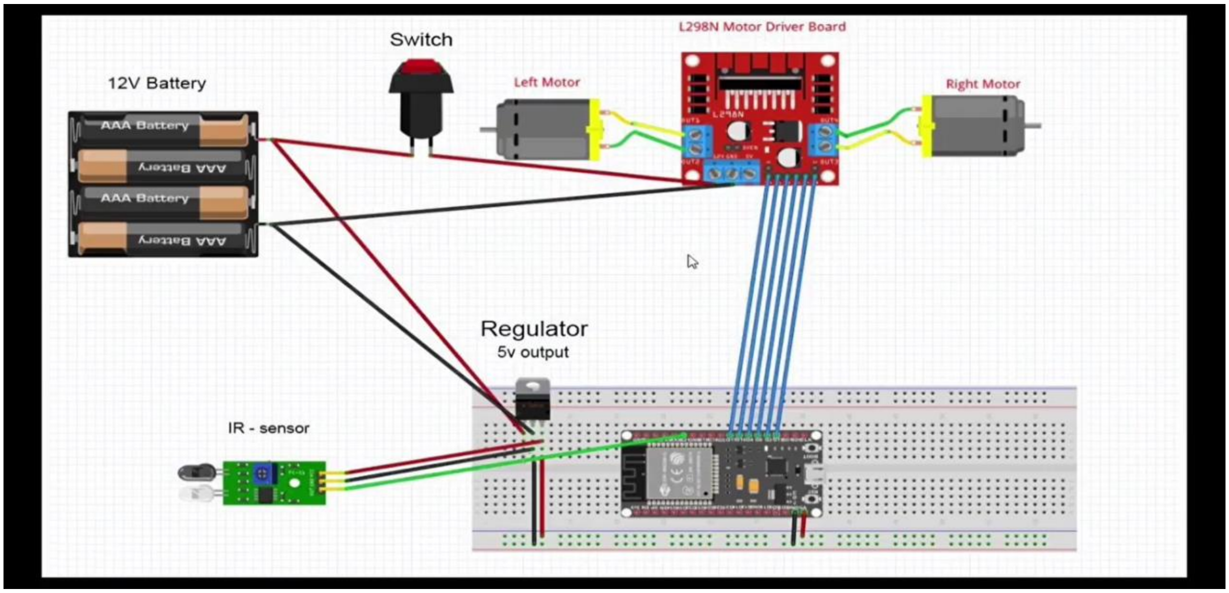
Aim of the project:

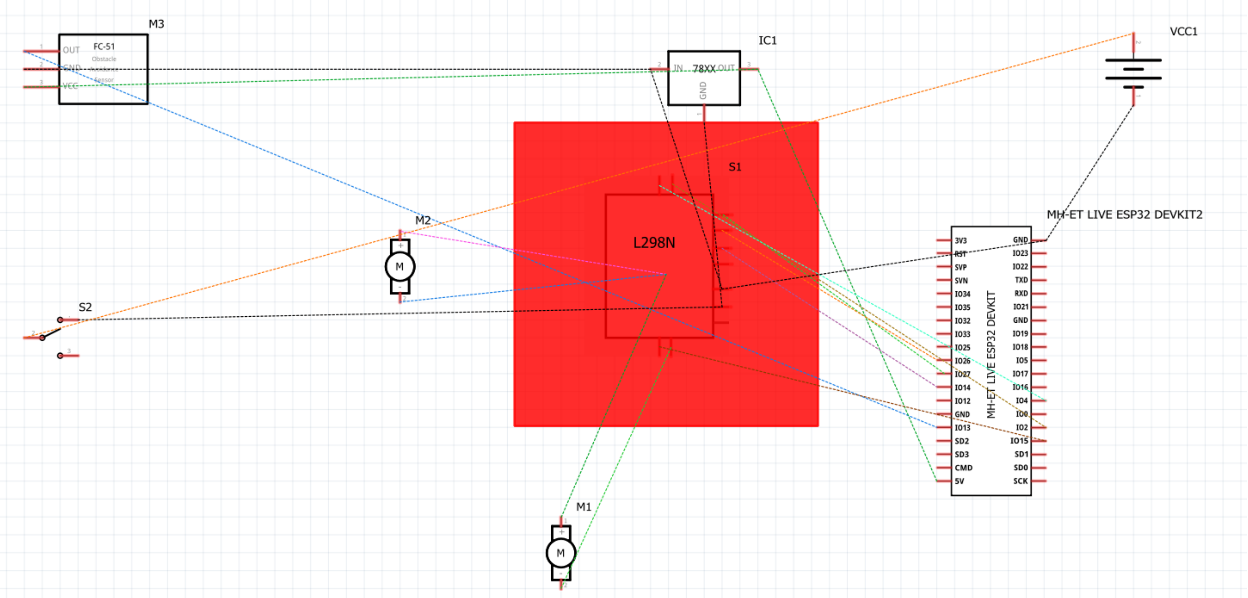
We want to simulate a car sensor, when the sensor read the car should stop and go back so it didn’t crash.

The used components:

- Chassis
- ESP32 Micro controller
- Motor driver
- Motors
- Wheels
- Battery
- IR sensor

Schematic of the circuit implemented:





Procedure to use this circuit:

1) Turn on the switch

2) Open the application and connect the Bluetooth (MY_CAR)

3) Then press “on” or speaker and say “start” to move it and press “off” or speaker and say “stop”.

Budget of the project:

- Chassis (65 L.E)
- Motors and Wheels (35*2+10*2 = 90 L.E)
- Motor driver (65 L.E)
- ESP32 (290 L.E)
- IR sensor (35 L.E)
- Jumpers and caste wheel nylon ball (20+20=40L.E)

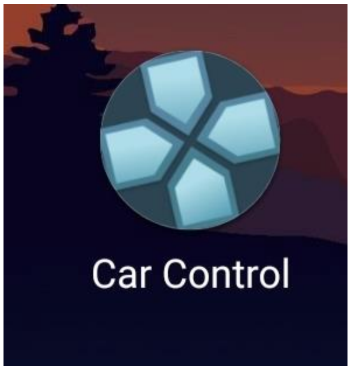
- Breadboard, regulator and switch (20+3.5+4=27.5 L.E)

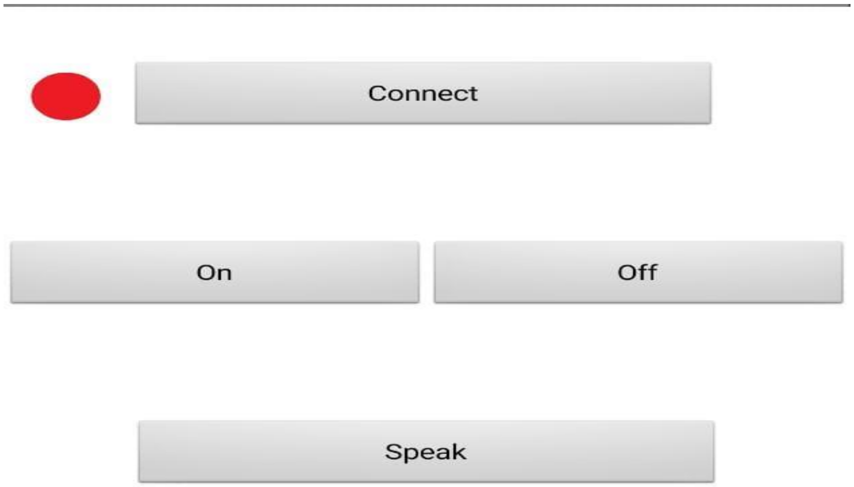
The Challenges

- One of the problems that we faced in the implementation of the project was something related to the sensor. The speed at the beginning did not allow the sensor to take enough time to recognize the presence of an obstacle in front of it and start making an action because the range of the IR sensor is narrow range reads only 2.5 cm but when we reduced the speed until we reached the speed that makes the sensor worked efficiently.

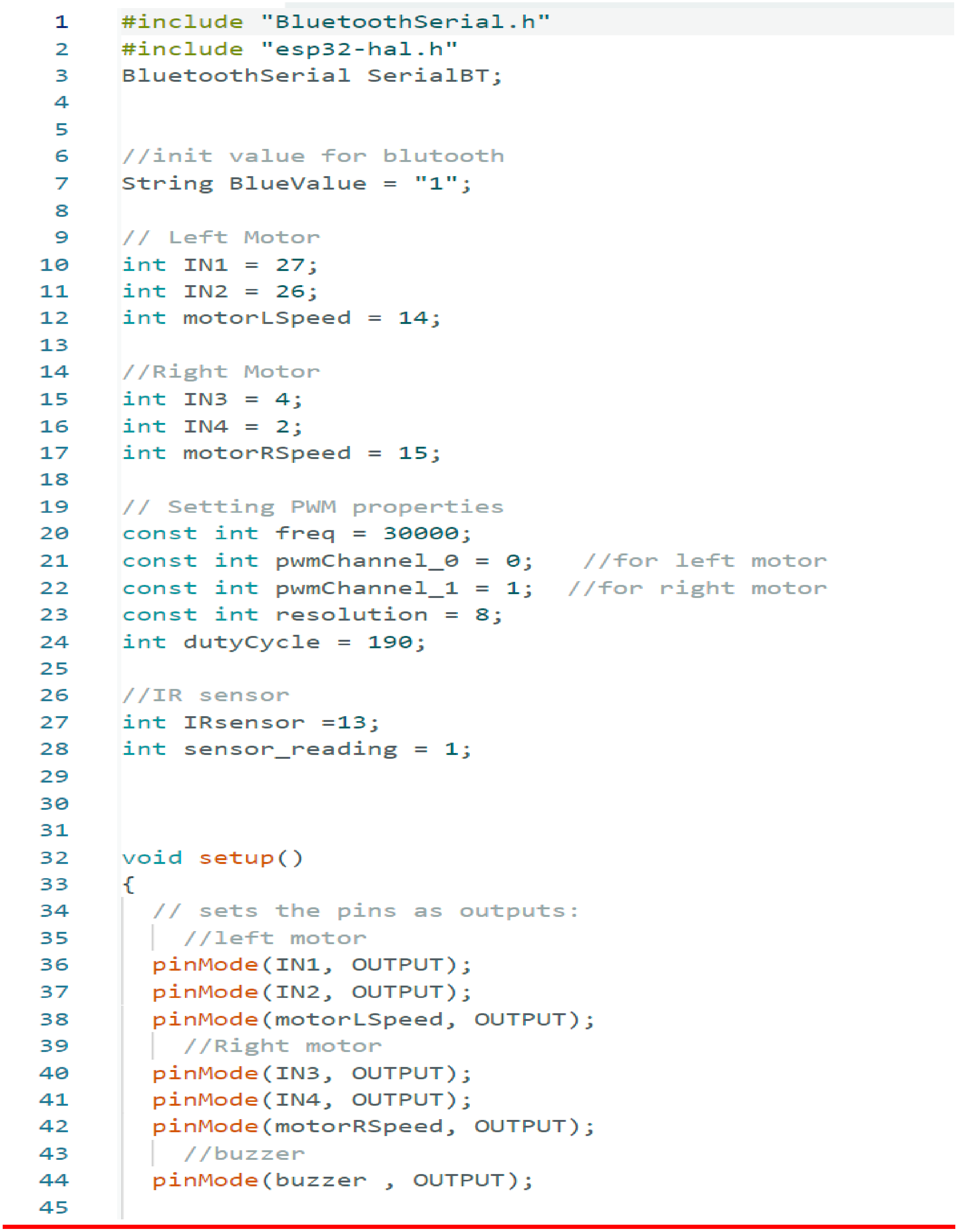
- Second thing that we faced was about the volt that we are going to use. We have 3 batteries each one is 4v so the total volt is 12v that value is not acceptable because it will cause damage to the components so we did a regulator circuit using breadboard to have an output volt equal to 5v.

The Bluetooth application:

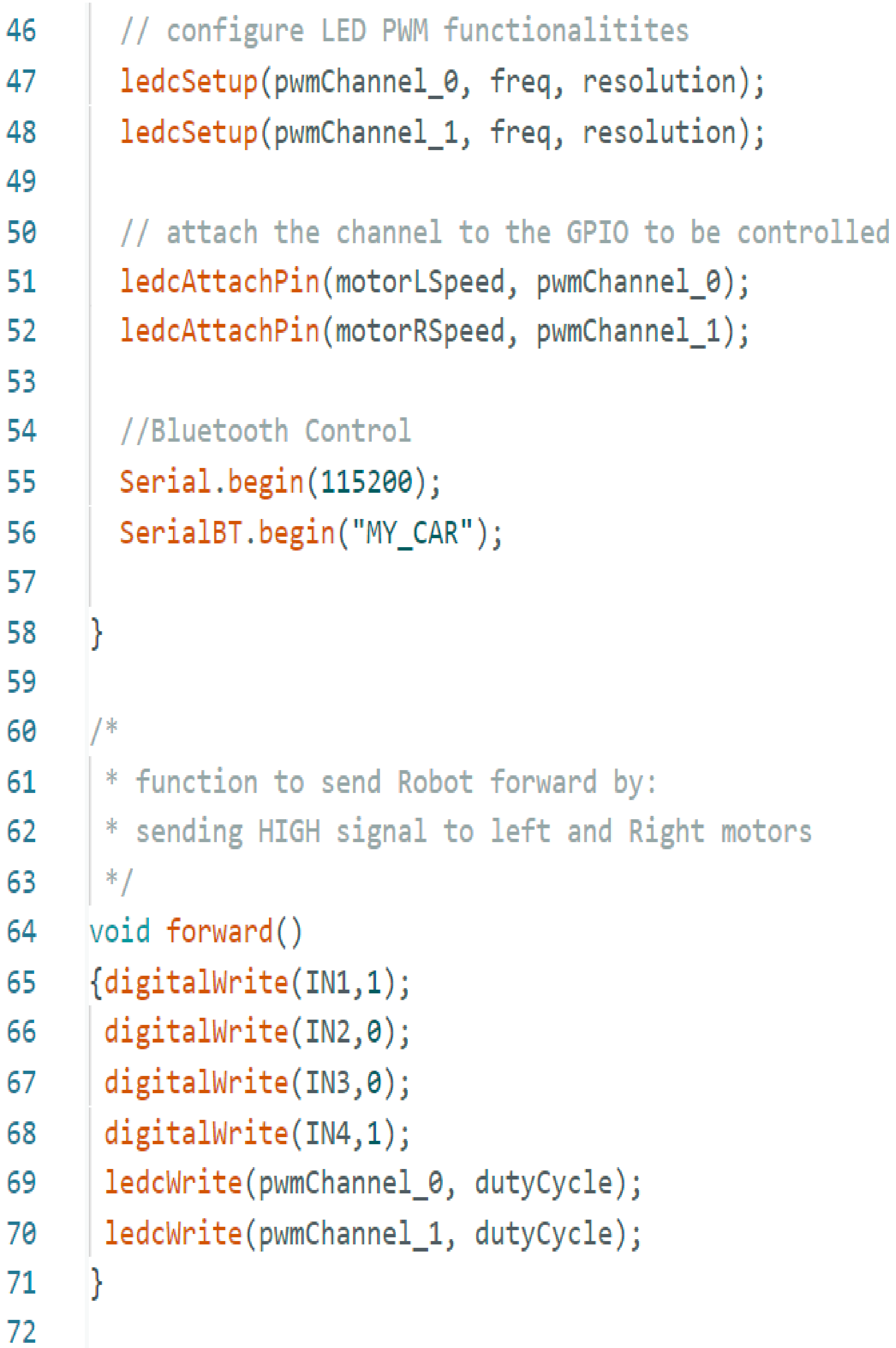




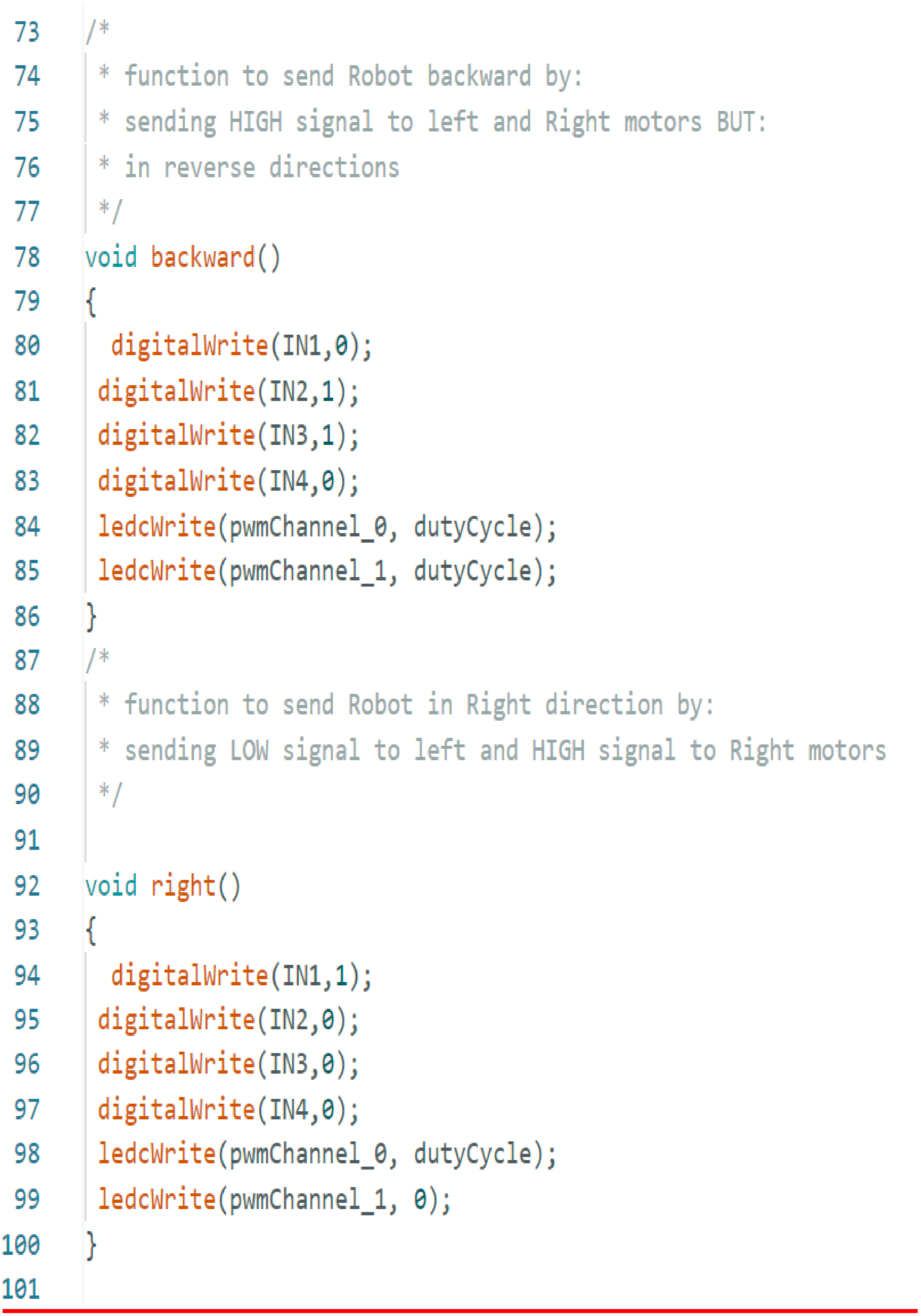
CODE



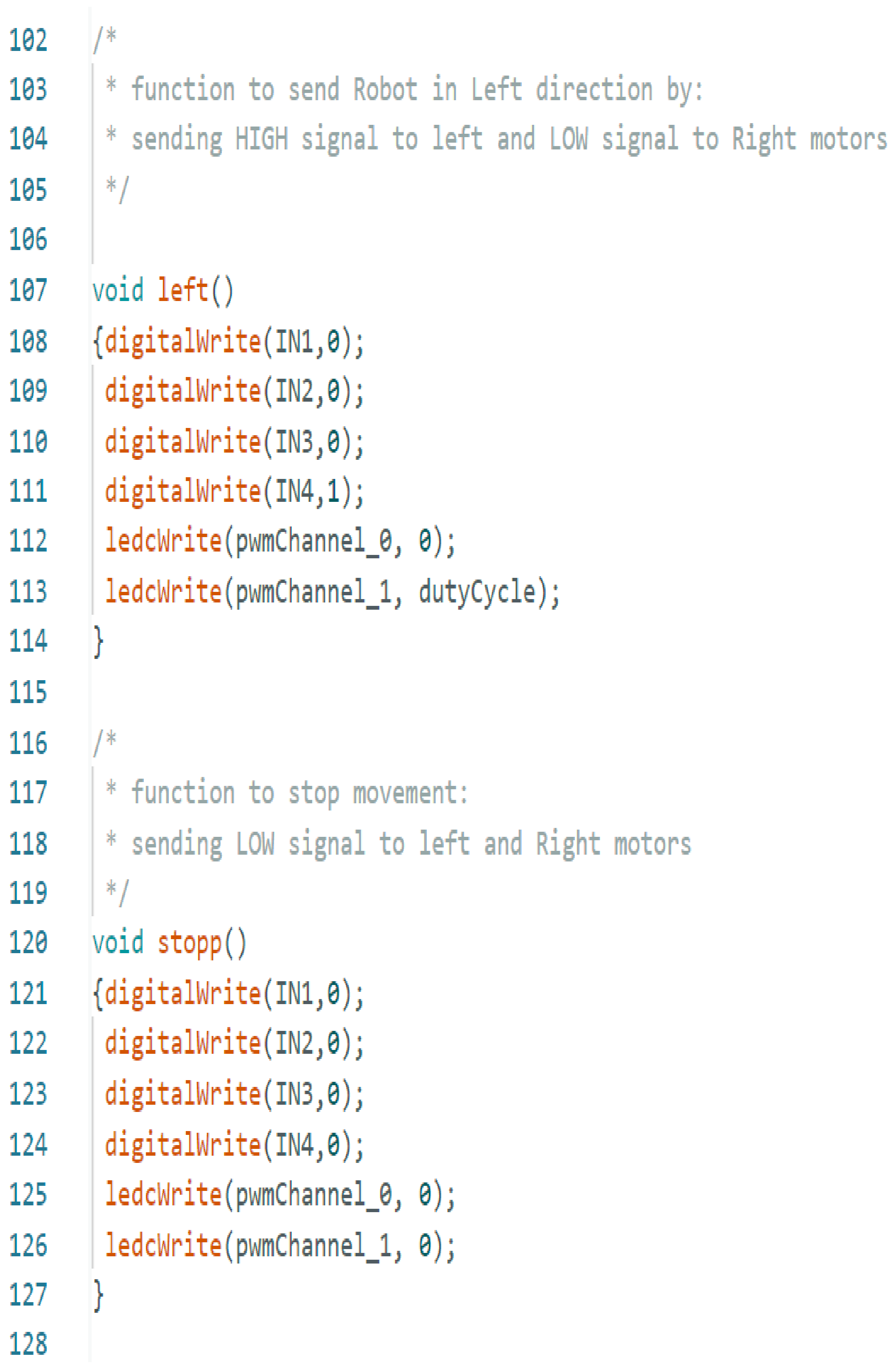
```
#include "BluetoothSerial.h"
#include "esp32-hal.h"
BluetoothSerial SerialBT;


//init value for blutooth
String BlueValue = "1";

// Left Motor
int IN1 = 27;
int IN2 = 26;
int motorLSpeed = 14;

//Right Motor
int IN3 = 4;
int IN4 = 2;
int motorRSpeed = 15;

// Setting PWM properties
const int freq = 30000;
const int pwmChannel_0 = 0;   //for left motor
const int pwmChannel_1 = 1;  //for right motor
const int resolution = 8;
int dutyCycle = 190;

//IR sensor
int IRsensor =13;
int sensor_reading = 1;



void setup()
{
  // sets the pins as outputs:
    //left motor
  pinMode(IN1, OUTPUT);
  pinMode(IN2, OUTPUT);
  pinMode(motorLSpeed, OUTPUT);
    //Right motor
  pinMode(IN3, OUTPUT);
  pinMode(IN4, OUTPUT);
  pinMode(motorRSpeed, OUTPUT);
    //buzzer
  pinMode(buzzer , OUTPUT);

```



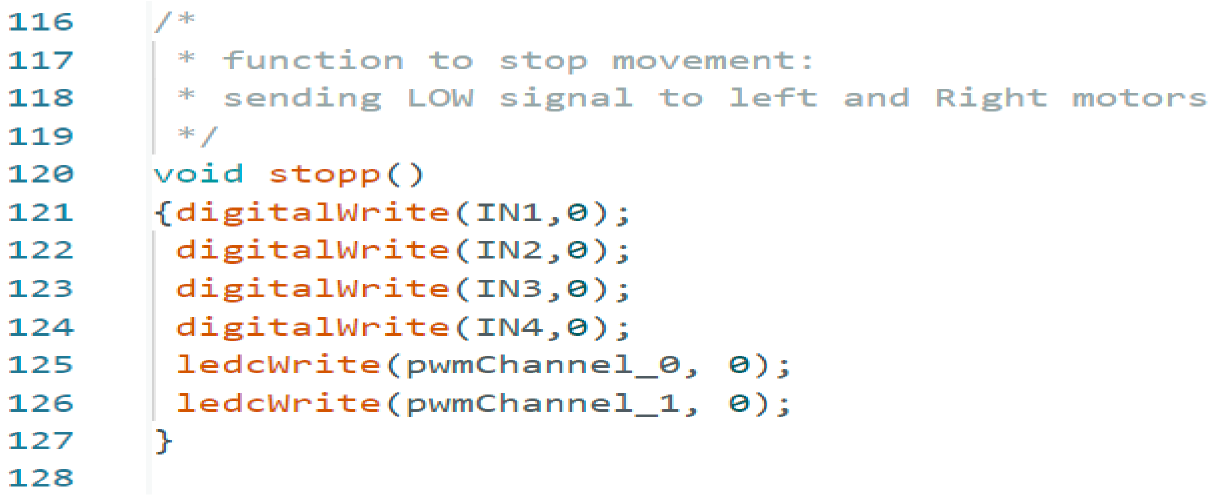
```
  // configure LED PWM functionalitites
  ledcSetup(pwmChannel_0, freq, resolution);
  ledcSetup(pwmChannel_1, freq, resolution);

  // attach the channel to the GPIO to be controlled
  ledcAttachPin(motorLSpeed, pwmChannel_0);
  ledcAttachPin(motorRSpeed, pwmChannel_1);

  //Bluetooth Control
  Serial.begin(115200);
  SerialBT.begin("MY_CAR");

}

/*
 * function to send Robot forward by:
 * sending HIGH signal to left and Right motors
 */
void forward()
{digitalWrite(IN1,1);
 digitalWrite(IN2,0);
 digitalWrite(IN3,0);
 digitalWrite(IN4,1);
 ledcWrite(pwmChannel_0, dutyCycle);
 ledcWrite(pwmChannel_1, dutyCycle);
}

```



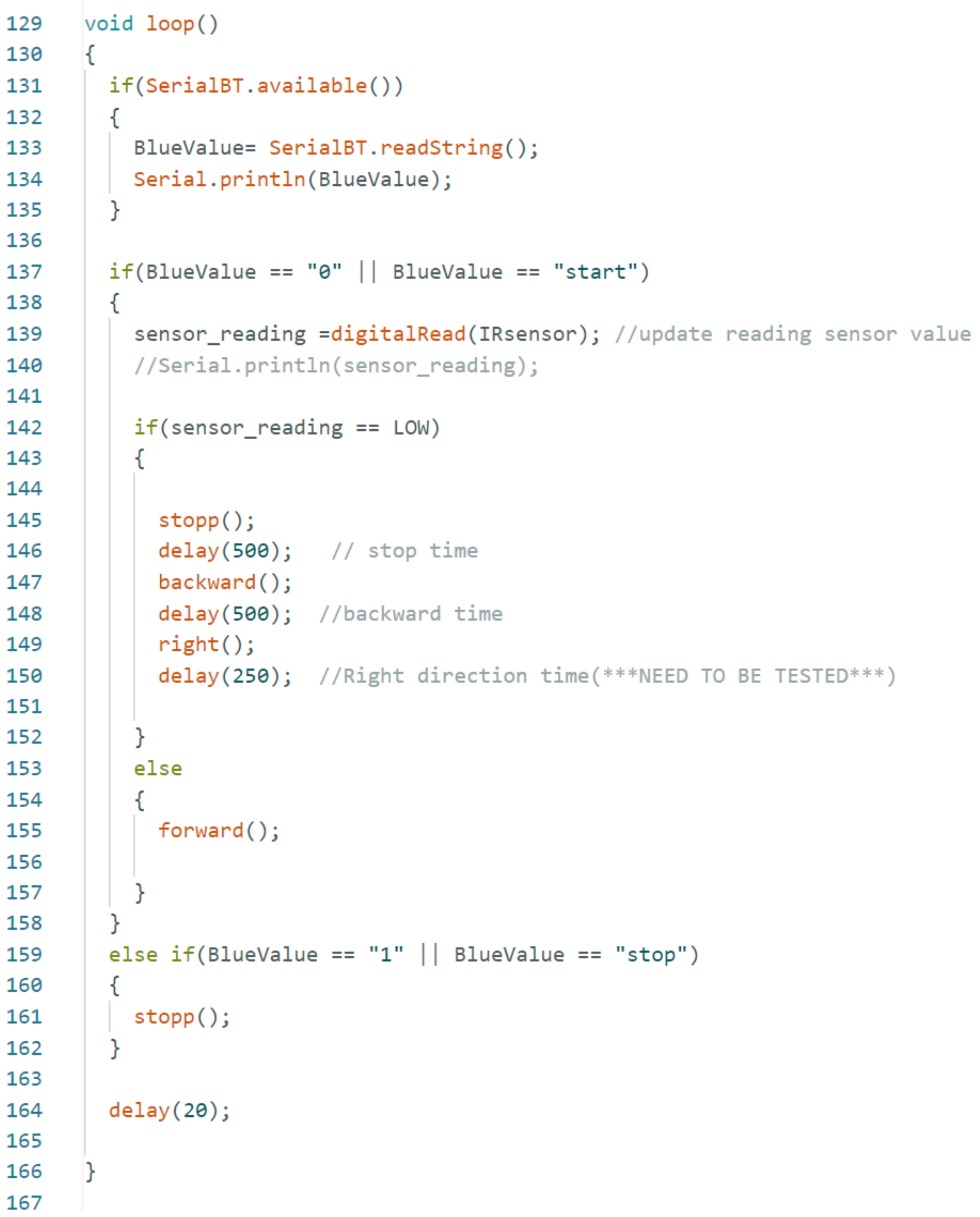
```
/*
 * function to send Robot backward by:
 * sending HIGH signal to left and Right motors BUT:
 * in reverse directions
 */
void backward()
{
  digitalWrite(IN1,0);
 digitalWrite(IN2,1);
 digitalWrite(IN3,1);
 digitalWrite(IN4,0);
 ledcWrite(pwmChannel_0, dutyCycle);
 ledcWrite(pwmChannel_1, dutyCycle);
}
/*
 * function to send Robot in Right direction by:
 * sending LOW signal to left and HIGH signal to Right motors
 */

void right()
{
  digitalWrite(IN1,1);
 digitalWrite(IN2,0);
 digitalWrite(IN3,0);
 digitalWrite(IN4,0);
 ledcWrite(pwmChannel_0, dutyCycle);
 ledcWrite(pwmChannel_1, 0);
}

```



```
/*
 * function to send Robot in Left direction by:
 * sending HIGH signal to left and LOW signal to Right motors
 */

void left()
{digitalWrite(IN1,0);
 digitalWrite(IN2,0);
 digitalWrite(IN3,0);
 digitalWrite(IN4,1);
 ledcWrite(pwmChannel_0, 0);
 ledcWrite(pwmChannel_1, dutyCycle);
}

/*
 * function to stop movement:
 * sending LOW signal to left and Right motors
 */
void stopp()
{digitalWrite(IN1,0);
 digitalWrite(IN2,0);
 digitalWrite(IN3,0);
 digitalWrite(IN4,0);
 ledcWrite(pwmChannel_0, 0);
 ledcWrite(pwmChannel_1, 0);
}

```



```
/*
 * function to stop movement:
 * sending LOW signal to left and Right motors
 */
void stopp()
{digitalWrite(IN1,0);
 digitalWrite(IN2,0);
 digitalWrite(IN3,0);
 digitalWrite(IN4,0);
 ledcWrite(pwmChannel_0, 0);
 ledcWrite(pwmChannel_1, 0);
}

```



```
void loop()
{
  if(SerialBT.available())
  {
    BlueValue= SerialBT.readString();
    Serial.println(BlueValue);
  }

  if(BlueValue == "0" || BlueValue == "start")
  {
    sensor_reading =digitalRead(IRsensor); //update reading sensor value
    //Serial.println(sensor_reading);

    if(sensor_reading == LOW)
    {

      stopp();
      delay(500);   // stop time
      backward();
      delay(500);  //backward time
      right();
      delay(250);  //Right direction time(***NEED TO BE TESTED***)

    }
    else
    {
      forward();

    }
  }
  else if(BlueValue == "1" || BlueValue == "stop")
  {
    stopp();
  }

  delay(20);

}

```