CS322: Computer Architecture
MINI-PROJECT

BLUETOOTH CONTROLLED ROBOT CAR

Done by: Harshita Meena(2001CS30),
Kushum(2001CS42)

Description:

In this project, we have made an arduino based bluetooth controlled robot car. This device is controlled by a bluetooth device like a mobile with an app, whose link is provided below, through a bluetooth module. We can control the direction and speed of the car using this app.

Components used:

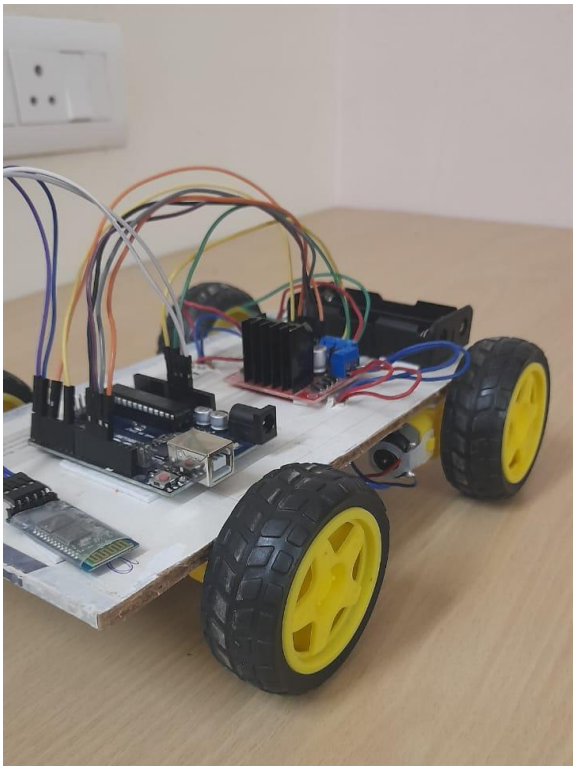
- Arduino UNO R3 module
- Motor Driver L298N module
- Bluetooth HC-05 module
- Gear motors - 4
- Wheels - 4
- 18650 Lithium ion batteries - 2
- Battery holder
- Jumper wires - male-to-female
- Connecting wires
- Soldering iron
- Wooden base

Making/Working:

First we have taken a wooden board (we have used a writing board) and attached the 4 gear motors to it using a double tape. After that we soldered the connecting wire to the terminals of the motor gears. Next, we attached the L298N module and the Arduino UNO R3 module to the board. We attached the connections of the gear motors to the L298N module. Then we made the other

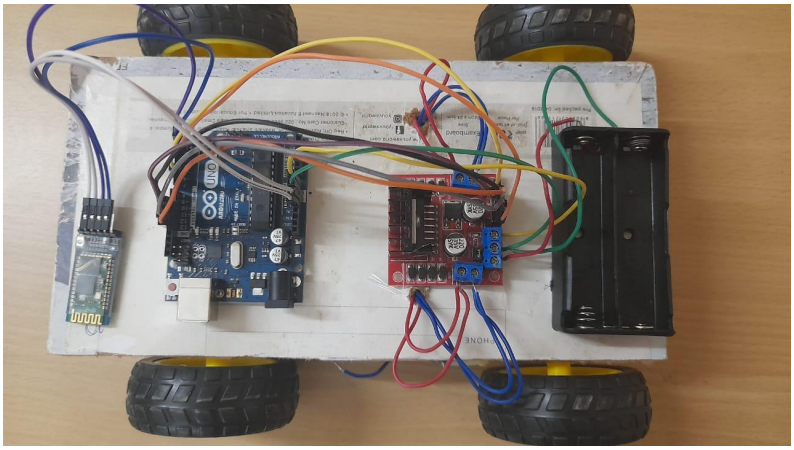
connections of the L298N module with the Arduino UNO R3 module. Next we attached the Bluetooth HC-05 module to the base and made its connections to the Arduino UNO R3 module. Then, we attached the battery holder to the base and made its connections with the L298N module. We attached the wheels to the gears. Then comes the code uploading part. While uploading the code we made sure that the bluetooth module is disconnected to the arduino and the batteries are also disconnected. After uploading the code, the connection with the bluetooth module is again made. Here, our robot car is ready. We now put the batteries in. Then we connect the car with the bluetooth RC controller app and give commands. The car moves according to the commands. The commands we have provided as per the bluetooth RC controller app. The commands are as follows:

Forward-> F
Backward-> B
Left-> L
Right-> R
Forward Left-> G
Forward Right-> I
Back Left-> H
Back Right-> J
Stop-> S



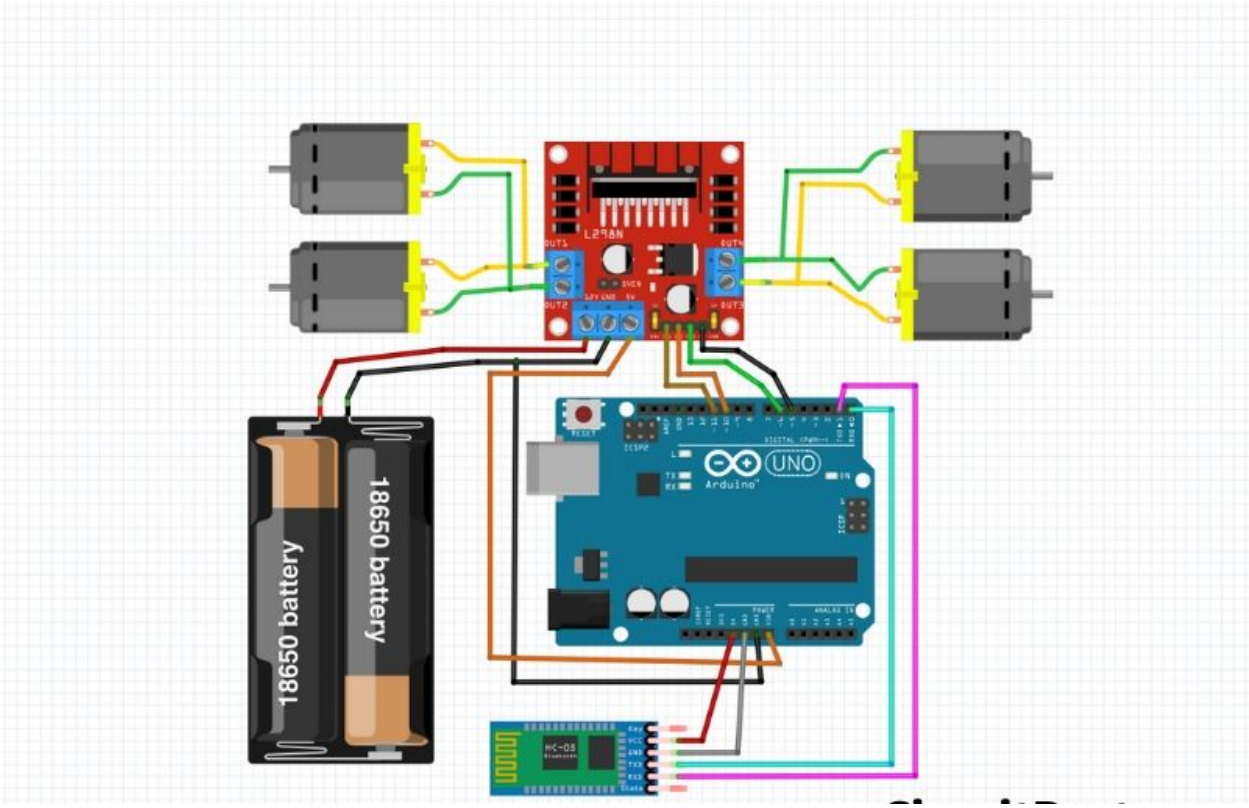
Connections:

Motor driver connections are attached to the L298N module's output 1, 2, 3 and 4. The ena and enb enable the PWM for motor A and motor B respectively. The ena, in1 and in2 controls the motor A whereas enb, in3 and in4 controls the motor B. ena is connected to the pin 3 of the UNO module whereas enb is connected to the pin 6 of the UNO module. The in1, in2, in3, and in4 are connected to pins 10, 11, 12, and 13 respectively. The ground of the L298N module is connected to the ground(power) of the UNO and the Vin is connected to the Vin(power) of the UNO module. The positive terminal of the battery holder is connected to the +12V of the L298N module and the negative terminal is connected to the ground of the L298N module. The bluetooth module's RX is connected to the TX(pin 1) of the UNO module, the TX is connected to the RX(pin 0) of the UNO module, the ground is connected to the ground(power) of the UNO module and the +5V is connected to the 5V(power) of the UNO module.



Circuit Diagram:

The circuit diagram is given below:



Code:

The code for the robot car working is as follows:

```
int in1 = 10 ;//L298n Motor Driver pins.
int in2 = 11 ;
int in3 = 12;
```

```
int in4 = 13;
int ena = 3;
int enb = 6;
int command; //Int to store app command state.
int Speed = 204; // 0 - 255.
int Speedsec;
int buttonState = 0;
int lastButtonState = 0;
int Turnradius = 0; //Set the radius of a turn, 0 - 255 Note:the robot
will malfunction if this is higher than int Speed.
int brakeTime = 45;
int brkonoff = 1; //1 for the electronic braking system, 0 for normal.
void setup() {
  pinMode(in1, OUTPUT);
  pinMode(in2, OUTPUT);
  pinMode(in3, OUTPUT);
  pinMode(in4, OUTPUT);
  Serial.begin(9600);  //Set the baud rate to your Bluetooth module.
}

void loop() {
  analogWrite(ena,150);
  analogWrite(enb,150);
  if (Serial.available() > 0) {
    command = Serial.read();
    Stop(); //Initialize with motors stopped.
    switch (command) {
      case 'F':
        forward();
        break;
      case 'B':
        back();
        break;
      case 'L':
        left();
        break;
      case 'R':
        right();
        break;
      case 'G':
```

```
      forwardleft();
      break;
    case 'I':
      forwardright();
      break;
    case 'H':
      backleft();
      break;
    case 'J':
      backright();
      break;
    case '0':
      Speed = 100;
      break;
    case '1':
      Speed = 140;
      break;
    case '2':
      Speed = 153;
      break;
    case '3':
      Speed = 165;
      break;
    case '4':
      Speed = 178;
      break;
    case '5':
      Speed = 191;
      break;
    case '6':
      Speed = 204;
      break;
    case '7':
      Speed = 216;
      break;
    case '8':
      Speed = 229;
      break;
    case '9':
      Speed = 242;
```

```
        break;
      case 'q':
        Speed = 255;
        break;
    }
    Speedsec = Turnradius;
    if (brkonoff == 1) {
      brakeOn();
    } else {
      brakeOff();
    }
  }
}

void forward() {
  analogWrite(in1, Speed);
  analogWrite(in3, Speed);
}

void back() {
  analogWrite(in2, Speed);
  analogWrite(in4, Speed);
}

void left() {
  analogWrite(in3, Speed);
  analogWrite(in2, Speed);
}

void right() {
  analogWrite(in4, Speed);
  analogWrite(in1, Speed);
}
void forwardleft() {
  analogWrite(in1, Speedsec);
  analogWrite(in3, Speed);
}
void forwardright() {
  analogWrite(in1, Speed);
  analogWrite(in3, Speedsec);
```

```
}
void backright() {
  analogWrite(in2, Speed);
  analogWrite(in4, Speedsec);
}
void backleft() {
  analogWrite(in2, Speedsec);
  analogWrite(in4, Speed);
}

void Stop() {
  analogWrite(in1, 0);
  analogWrite(in2, 0);
  analogWrite(in3, 0);
  analogWrite(in4, 0);
}

void brakeOn() {
  //Here's the future use: an electronic braking system!
  // read the pushbutton input pin:
  buttonState = command;
  // compare the buttonState to its previous state
  if (buttonState != lastButtonState) {
    // if the state has changed, increment the counter
    if (buttonState == 'S') {
      if (lastButtonState != buttonState) {
        digitalWrite(in1, HIGH);
        digitalWrite(in2, HIGH);
        digitalWrite(in3, HIGH);
        digitalWrite(in4, HIGH);
        delay(brakeTime);
        Stop();
      }
    }
    // save the current state as the last state,
    //for next time through the loop
    lastButtonState = buttonState;
  }
}
void brakeOff() {
```

```
}
```

Bluetooth RC controller app link:

https://play.google.com/store/apps/details?id=braulio.calle.bluetoothRCcontroller

Contributions:

- Code: Harshita Meena(2001CS30)
- Hardware and Connections: Kushum(2001CS42)