CMP3010 - Embedded Systems Programming

Lab Term Project

Topic: Bluetooth Controlled Robot

Ugur Taysi (2101101)

Atif Ghani (2019751)

Istanbul, June 2023

Introduction

In the modern era we live in, Robotics has gained a significant share in the field of technology. One important aspect of Robotics is the ability to control them remotely. This project aims to provide an efficient solution by creating a Bluetooth controlled robot.

We chose to construct a Remote Control (RC) car using Bluetooth technology, allowing us to control it from an Android device via a Bluetooth connection. The concept of embedded systems programming was employed in the project to make the interaction between hardware (like motors, batteries, Bluetooth modules) and software (the program written in Arduino programming language).

Task Distribution

The tasks for the project were divided as follows:

Task / Person	Uğur Tayşi	Atif Ghani
Planning which components are needed		X
Planning initial circuit design	X	
Buying components & gear	X	X
Bargaining at Yazıcıoğlu Electronics Center.	X	
Assembling the robot	X	X
Programming	X	X
Circuit modeling	X	
Producing report		X
Producing presentation slides		X

Figure 1: Responsibility Matrix

Materials and Components

The components used in the project and their respective purposes are as follows:

- **DC Motor (4x):** These are used to rotate the wheels of the robot, thereby enabling its movement.
- **Wheel (4x):** These are attached to the motors, allowing the robot to move.
- **20x12cm Wooden Board:** Used as the chassis for the robot, providing a base to attach other components.
- **LED (2x Red, 2x Yellow):** Used for illumination and the stress signal.
- **Arduino Uno:** The main controller unit that connects each part and allows programming.
- **Buzzer:** Used for the horn and stress signal.
- **Breadboard:** We used a breadboard for making easier additional circuit connections.
- **Li-Po Battery (2x in series):** Used to power the system.
- **TP4056 Charging Module:** Charges the batteries via a micro-USB cable.
- **HC06 Bluetooth Module:** Allows Bluetooth connection which is the main method for connecting to the remote.
- **L298N Controller Module:** A controller between each motor and the Arduino. Allows us to program accordingly.
- **220Ω Resistor (4x):** Required for each LED to control the current flow.
- **Jumper Cables:** Connects each component together.

Robot Functions

The functions that the robot can perform include:

1. Motor control for moving the robot in various directions:
    - Forward
    - Backward
    - Left
    - Right
    - Forward Right
    - Forward Left
    - Backward Left
    - Backward Right
2. Connecting to the Android Remote App via Bluetooth.
3. Turning headlights on and off.
4. Turning taillights on and off.
5. Activating the horn.
6. Activating the stress signal.
7. Charging via Micro-USB Cable.

Circuit Design

The circuit was designed with a clear understanding of how each component interacts with each other. The Arduino Uno served as the primary control unit, with the L298N module acting as an intermediary between the Arduino and the DC motors. The Bluetooth module was connected to the Arduino, facilitating wireless control. LEDs were connected through resistors to prevent excess current flow, and the buzzer was directly connected to a digital I/O pin on the Arduino. All the components were powered by the Li-Po battery, and the TP4056 module was used for charging these batteries. A schematic diagram of the full circuit is provided in the project document.

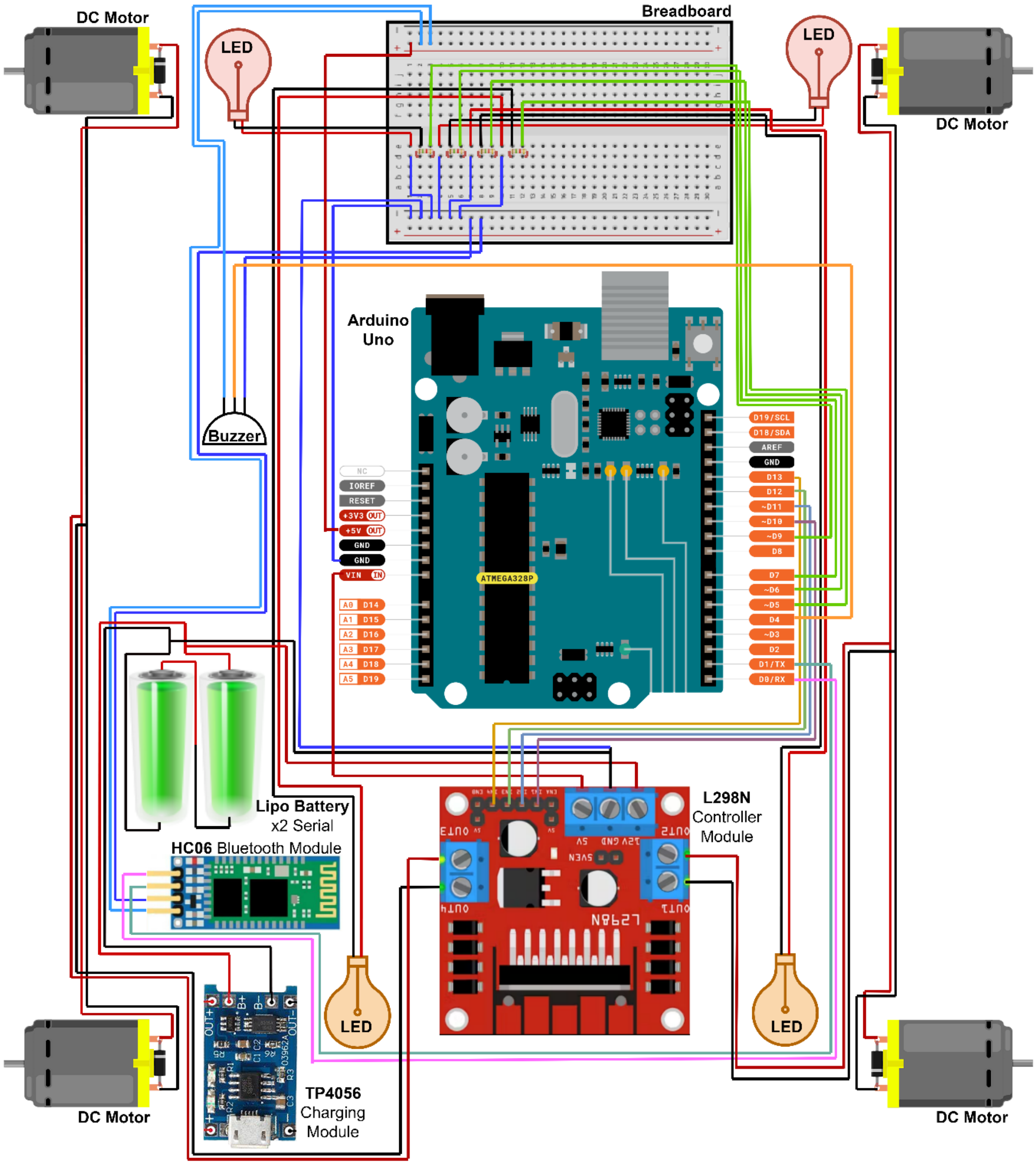

Figure 2: Circuit Diagram of the Robot

Arduino Code Explanation

The Arduino code primarily involved using the Serial library for communication with the Bluetooth module. The Bluetooth state was read from the Serial buffer and used to determine the robot's actions.

The robot's movements were controlled by changing the speed of the motors. Depending on the received Bluetooth state, different combinations of motor speeds were used to make the robot move forward, backward, turn left or right, or stop.

Additional robot features like headlights, taillights, horn, and stress signals were controlled by turning respective pins HIGH or LOW as per the received Bluetooth state.

The full code is included in Appendix A, and on GitHub (https://github.com/utaysi/Arduino-Bluetooth-RC-Car).

Conclusion

The Bluetooth Controlled Robot project provided hands-on experience in embedded systems programming, hardware interfacing, and practical application of Bluetooth technology. It required careful task management, system design, and problem-solving skills to successfully complete.

Overall, the project was a success in that it met all its initial objectives, providing a functioning Bluetooth-controlled robot demonstrating the power and versatility of embedded systems.

Appendix A

```
//L298N Connection

  const int motorA1  = 5;  // L298N IN3 Input

  const int motorA2  = 6;  // L298N IN1 Input

  const int motorB1  = 10; // L298N IN2 Input

  const int motorB2  = 9;  // L298N IN4 Input

  const int buzzer  = 7;  // L298N IN4 Input

  #define LED_PIN1 11

  #define LED_PIN2 13

  #define LED_PIN3 12

  #define LED_PIN4 8



  #define POTENTIOMETER_PIN A1


  int i=0;

  int j=0;

  int state; // Bluetooth State

  int vSpeed=255; // Standard speed, set between 0-255


void setup() {

    // Set pins

    pinMode(LED_PIN1, OUTPUT); // back left

    pinMode(LED_PIN2, OUTPUT); // back right

    pinMode(LED_PIN3, OUTPUT); // front left?

    pinMode(LED_PIN4, OUTPUT); // front right?

    pinMode(motorA1, OUTPUT);

    pinMode(motorA2, OUTPUT);

    pinMode(motorB1, OUTPUT);

    pinMode(motorB2, OUTPUT);

    pinMode(buzzer, OUTPUT);

    // Open serial port in 9600 baud speed

    Serial.begin(9600);

}


void loop() {

  //Save incoming data to state variable.

    if(Serial.available() > 0){

      state = Serial.read();

    }


  // 4 speed levels from the app. Must be set between 0-255.
```

```
  if (state == '0'){
    vSpeed=0;}
  else if (state == '1'){
    vSpeed=100;}
  else if (state == '2'){
    vSpeed=180;}
  else if (state == '3'){
    vSpeed=200;}
  else if (state == '4'){
    vSpeed=255;}

/***********************Forward****************************/
  if (state == 'F') {
    analogWrite(motorA1, vSpeed);
    analogWrite(motorA2, vSpeed);
    analogWrite(motorB1, 0);
    analogWrite(motorB2, 0);
  }
/**********************Forward Left************************/
  else if (state == 'G') {
    analogWrite(motorA1,vSpeed );
    analogWrite(motorA2, 100);
    analogWrite(motorB1, 0);
    analogWrite(motorB2, 0);
  }
  /**********************Forward Right************************/
  else if (state == 'I') {
      analogWrite(motorA1, 100);
      analogWrite(motorA2, vSpeed);
      analogWrite(motorB1, 0);
      analogWrite(motorB2, 0);
  }

/***********************Backward****************************/
  else if (state == 'B') {
    analogWrite(motorA1, 0);
    analogWrite(motorA2, 0);
    analogWrite(motorB1, vSpeed);
    analogWrite(motorB2, vSpeed);
  }
/**********************Backward Left************************/
  else if (state == 'H') {
```

```
    analogWrite(motorA1, 0);
    analogWrite(motorA2, 0);
    analogWrite(motorB1, 100);
    analogWrite(motorB2, vSpeed);
  }
/**********************Backward Right************************/
  else if (state == 'J') {
    analogWrite(motorA1, 0);
    analogWrite(motorA2, 0);
    analogWrite(motorB1, vSpeed);
    analogWrite(motorB2, 100);
  }

/***************************Left*****************************/
  else if (state == 'L') {
    analogWrite(motorA1, vSpeed);
    analogWrite(motorA2, 0);
    analogWrite(motorB1, vSpeed);
    analogWrite(motorB2, 0);
  }
/***************************Right*****************************/
  else if (state == 'R') {
    analogWrite(motorA1, 0);
    analogWrite(motorA2, vSpeed);
    analogWrite(motorB1, 0);
    analogWrite(motorB2, vSpeed);
  }

/************************Stop*****************************/
  else if (state == 'S'){
      analogWrite(motorA1, 0);
      analogWrite(motorA2, 0);
      analogWrite(motorB1, 0);
      analogWrite(motorB2, 0);
  }
  /************************Front Lights On*****************************/
  else if (state == 'W'){
      digitalWrite(LED_PIN3, HIGH);
      digitalWrite(LED_PIN4, HIGH);
  }
  /************************Back Lights On*****************************/
  else if (state == 'U'){
```

```
    digitalWrite(LED_PIN1, HIGH);
    digitalWrite(LED_PIN2, HIGH);
}
/************************Front Lights Off****************************/
else if (state == 'w'){
    digitalWrite(LED_PIN3, LOW);
    digitalWrite(LED_PIN4, LOW);
}
/************************Back Lights Off****************************/
else if (state == 'u'){
    digitalWrite(LED_PIN1, LOW);
    digitalWrite(LED_PIN2, LOW);
}
/************************Horn On****************************/
else if (state == 'V'){
  int counter = 0;
    while(counter <50){
    digitalWrite(buzzer, HIGH);
    delay(10);
    digitalWrite(buzzer, LOW);
    delay(10);
    counter += 1;
  }
}

/************************Horn Off****************************/
else if (state == 'v'){
    digitalWrite(buzzer, LOW);
}
/************************Stress Signal On****************************/
else if (state == 'X'){
  int counter = 0;
    while (counter < 10){
    digitalWrite(LED_PIN1, HIGH);
    delay(50);
    digitalWrite(buzzer, HIGH);
    digitalWrite(LED_PIN2, HIGH);
    delay(50);
    digitalWrite(buzzer, LOW);
    digitalWrite(LED_PIN3, HIGH);
    delay(50);
    digitalWrite(buzzer, HIGH);
```

```
        digitalWrite(LED_PIN4, HIGH);

        delay(50);

        digitalWrite(buzzer, LOW);

        digitalWrite(LED_PIN1, LOW);

        delay(50);

        digitalWrite(buzzer, HIGH);

        digitalWrite(LED_PIN2, LOW);

        delay(50);

        digitalWrite(buzzer, LOW);

        digitalWrite(LED_PIN3, LOW);

        delay(50);

        digitalWrite(buzzer, HIGH);

        digitalWrite(LED_PIN4, LOW);

        delay(50);

        digitalWrite(buzzer, LOW);

        counter += 1;

        }

    }
    /************************Stress Signal Off*****************************/
    else if (state == 'x'){


      digitalWrite(LED_PIN1, LOW);

      digitalWrite(LED_PIN2, LOW);

      digitalWrite(LED_PIN3, LOW);

      digitalWrite(LED_PIN4, LOW);

      digitalWrite(buzzer, LOW);

    }

}
```