
A Bluetooth Controlled Car with ESP32

Spandan Guragain
scientiac@duck.com
Robotics Club, IOEPC

July 6, 2025

ABSTRACT

This is a simple documentation on how you can build a remote controlled bluetooth car using ESP32 and Arduino IDE.

Keywords ESP32 · ArduinoIDE

1 Introduction

This guide will walk you through creating a remote controlled Bluetooth car using an ESP32 microcontroller. You'll learn how to set up the development environment, wire the components, and program the ESP32 to receive Bluetooth commands for controlling the car's movement.

2 Development Environment Setup

Before starting with the hardware, we need to prepare your computer with the necessary software and drivers.

2.1 Arduino IDE Installation

First you need to install ArduinoIDE on your computer. <https://www.arduino.cc/en/software/>

2.2 USB to UART Bridge Drivers

You'll need VCP drivers to use USB to UART bridge to transfer data/code to your ESP32 from your computer. <https://www.silabs.com/developer-tools/usb-to-uart-bridge-vcp-drivers?tab=downloads>.

2.2.1 Platform-Specific Installation Instructions

- If you are on windows download CP210x VCP Windows.
- If you are on mac you should just install the driver using brew.
 - `brew install --cask silicon-labs-vcp-driver`
- If you are on linux, you should put yourself to the dialout group.
 - `sudo groupadd dialout`
 - `sudo usermod -aG dialout $USER`

After installing the drivers, you should reboot your computer.

2.3 ESP32 Board Package Installation

Then, you'll have to install ESP32 Libraries on your IDE. You can do that by following the steps below:

1. Open Arduino IDE
2. On the top right of your app, click on File and Preferences.
3. There is a field for Additional URL Board Manager URL.
4. Put https://raw.githubusercontent.com/espressif/arduino-esp32/gh-pages/package_esp32_index.json there.
 - If you already have a URL there, you can put it as a comma separated value.
5. Then click on OK to save.
6. Now, on the right sidebar, click on the icon which looks like an Arduino Board, it opens the Board Manager.
7. On the board manager, search for esp32 and Install esp32 by **Espressif Systems**

3 Hardware Components and Wiring

This section covers the physical components needed and how to connect them properly.

3.1 ESP32 Development Board

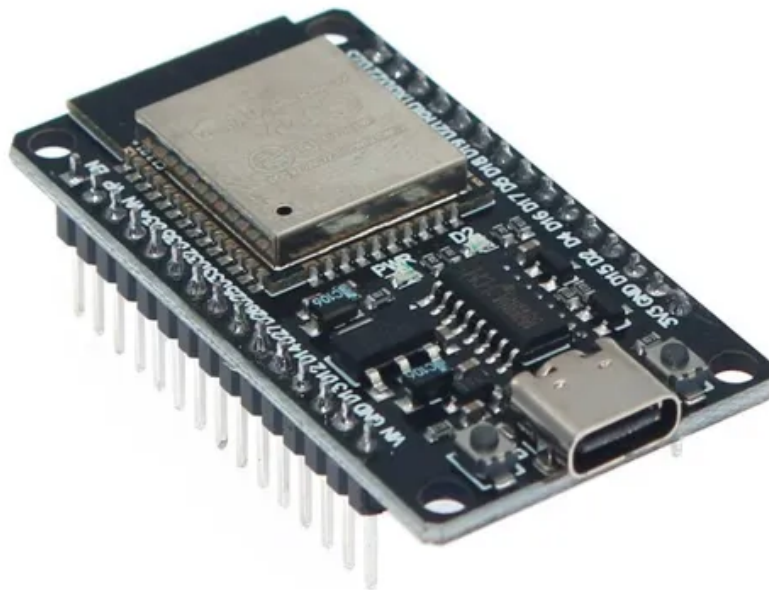


Figure 1: DOIT ESP32 DEV KIT V1

Now, connect your ESP32 on your computer with a USB cable and look for the the port to show up. Select the port and select the board you are using, it generally is **DOIT ESP32 DEV KIT V1**.

3.2 Circuit Assembly

Finally, we are ready to write something to the ESP32 that you have. But, lets first connect all the components to make the **Remote Controlled Bluetooth Car**.

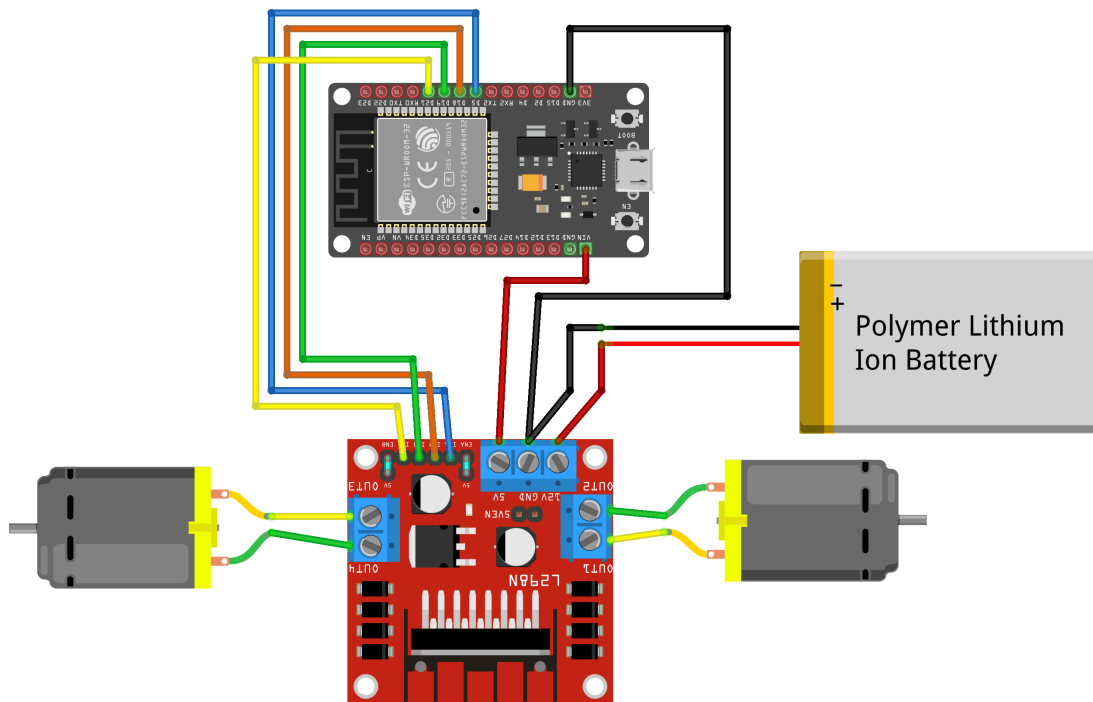


Figure 2: Component and Circuit Diagram to make the Bluetooth Controlled Car
The final version of the bot I made looks like this:

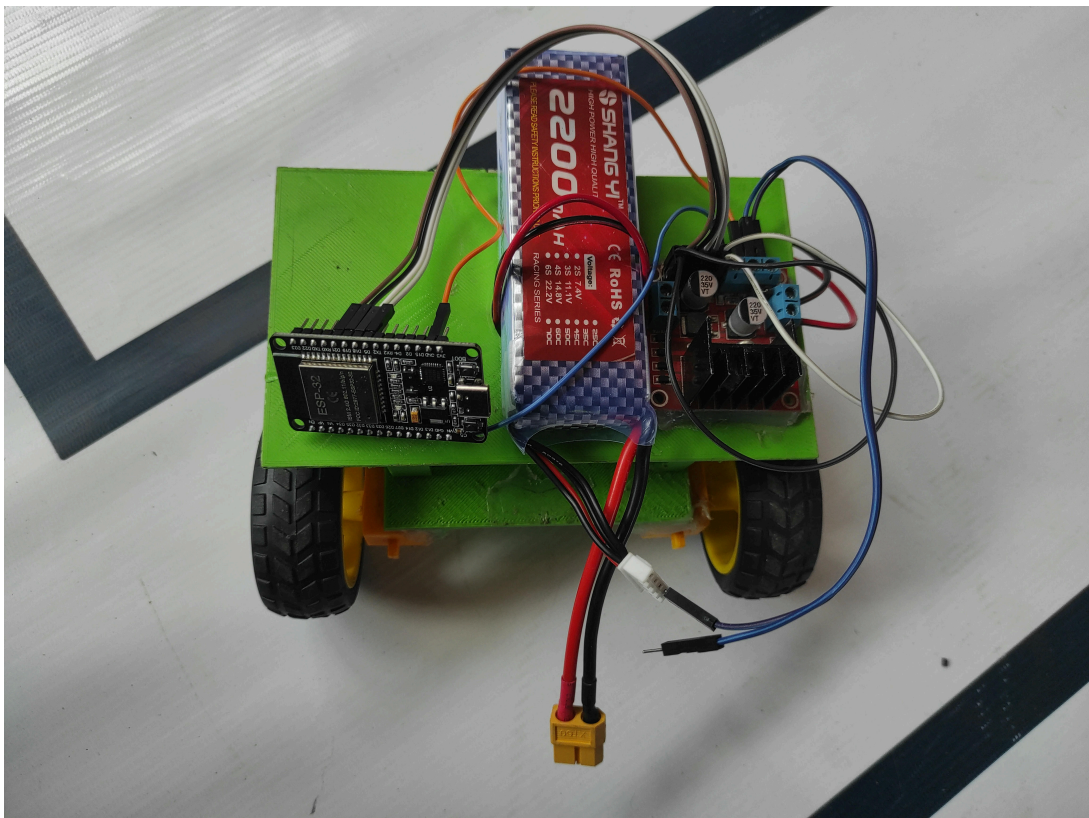


Figure 3: The Bluetooth Controlled Car

4 Programming the ESP32

This section contains the complete code for the Bluetooth controlled car functionality.

4.1 Code Upload Process

After connecting everything we now upload the code for the Bluetooth controlled car. Paste the following code on your IDE:

```
#include "BluetoothSerial.h"
BluetoothSerial SerialBT;

const int LEFT_CW    = 5;
const int LEFT_CCW   = 18;
const int RIGHT_CW   = 19;
const int RIGHT_CCW  = 21;
const int STATUS_LED = 2;

const bool ACTIVE_HIGH = true;

bool controlEnabled = false;

int ON  = ACTIVE_HIGH ? HIGH : LOW;
int OFF = ACTIVE_HIGH ? LOW  : HIGH;

void allStop() {
    digitalWrite(LEFT_CW, OFF);
    digitalWrite(LEFT_CCW, OFF);
    digitalWrite(RIGHT_CW, OFF);
    digitalWrite(RIGHT_CCW, OFF);
}

void forward() {
    digitalWrite(LEFT_CW, ON);
    digitalWrite(LEFT_CCW, OFF);
    digitalWrite(RIGHT_CW, ON);
    digitalWrite(RIGHT_CCW, OFF);
    delay(50);
    allStop();
}

void backward() {
    digitalWrite(LEFT_CW, OFF);
    digitalWrite(LEFT_CCW, ON);
    digitalWrite(RIGHT_CW, OFF);
    digitalWrite(RIGHT_CCW, ON);
    delay(50);
    allStop();
}

void pivotRight() {
    digitalWrite(LEFT_CW, OFF);
    digitalWrite(LEFT_CCW, ON);
    digitalWrite(RIGHT_CW, ON);
}
```

```

    digitalWrite(RIGHT_CCW, OFF);
    delay(50);
    allStop();
}

void pivotLeft() {
    digitalWrite(LEFT_CW, ON);
    digitalWrite(LEFT_CCW, OFF);
    digitalWrite(RIGHT_CW, OFF);
    digitalWrite(RIGHT_CCW, ON);
    delay(50);
    allStop();
}

void setup() {
    Serial.begin(9600);
    pinMode(LEFT_CW, OUTPUT);
    pinMode(LEFT_CCW, OUTPUT);
    pinMode(RIGHT_CW, OUTPUT);
    pinMode(RIGHT_CCW, OUTPUT);
    pinMode(STATUS_LED, OUTPUT);
    allStop();
    digitalWrite(STATUS_LED, OFF);
    SerialBT.begin("especial");
}

void loop() {
    if (Serial.available()) SerialBT.write(Serial.read());

    if (SerialBT.available()) {
        char c = SerialBT.read();

        if (c == 'l') {
            controlEnabled = true;
            digitalWrite(STATUS_LED, ON);
        } else if (c == '0') {
            controlEnabled = false;
            digitalWrite(STATUS_LED, OFF);
            allStop();
        } else if (controlEnabled) {
            switch (toupper(c)) {
                case 'F': forward();    break;
                case 'B': backward();   break;
                case 'L': pivotLeft();   break;
                case 'R': pivotRight();  break;
                case 'S': allStop();     break;
            }
        }
    }

    delay(5);
}

```

Click on the **compile** button, check for errors and then click on the **upload** button.

4.2 Troubleshooting Active Low Configuration

If your ESP is active low, then all the controls may be flipped. **ON** will work like **OFF** and vice-versa. You should change the **ACTIVE_HIGH** variable to **false**.

5 Testing and Operation

This section explains how to test your Bluetooth car using a mobile application.

5.1 Mobile App Setup

I have used **Serial Bluetooth Terminal** android app to test if it's working.

To use it:

1. Pair your ESP32 bluetooth with your phone, I have named it **especial** in the code, so I paired with **especial**.
2. Open the application and on the side panel select **Devices** and select **especial** or whatever you named your ESP32 Bluetooth.
3. Come back to the **Terminal**, connect to your ESP32 by clicking the **Connect Icon** on the top of the screen.

5.2 Control Commands

Send messages to the car to use it:

- 1 → Enable The Controls
- 0 → Disable The Controls
- L → Turn Left
- B → Go Back
- F → Go Front
- R → Turn Right
- S → Stop