MS101 Makerspace
2024-25/II Spring
Expt 5:Remote Control of BO Motors

Febrauary 11, 2025

Objectives:

- To control the speed of two BO motors remotely through Arduino Nano, Node MCU and ESP-WROOM32 microcontrollers.

List of components:

Arduino Nano, Node MCU (ESP8266 MOD), ESP-WROOM32, LM1117 Voltage Regulator,BO motors, L298 motor driver card, potentiometers.

Introduction

The Drone project requires the use of a gadget to send data remotely to the drone so as to control its various parameters, such as throttle, pitch, roll and yaw. The Drone used in the MS101 project has a Node MCU (ESP8266) which receives data over Wi-Fi for controlling four BLDC (brushless DC) motors through an Arduino Nano and motor driver cards.

In 2023-24/Spring semester a mobile app, viz. Remote XY was used to send drone parameters over Wi-Fi so as to emulate a joy stick. In the current semester students are required to build a ESP-WROOM32 based Joy Stick to control the drone parameters.

Experiment 5 has been designed to give a good background for remote speed control of BO motors. It is expected that after performing Expt 5, students will have the sufficient background to implement ESPWROOM32 based Joy Stick for the Drone project.

This experiment has three parts as detailed below:

- Part A: Interfacing two BO motors through Arduino Nano board and L298 driver card.
- Part B: Controlling the speed of two BO Motors through Arduino Nano with control from a Node MCU (ESP8266 MOD) board.
- Part C: Remote speed control of two BO motors using ESP-WROOM32, Node MCU (ESP8266 MOD) and Arduino Nano boards.

In Expt 4 one BO motor was run through Arduino Nano and the L298 driver card. In Part A we shall interface one more BO motor. These two motors will be run using the given Arduino sketch.

In Part B we shall extend the circuit of Part A so as to add a Node MCU (ESP 8266 MOD) card. Two potentiometers will be connected to the Node MCU, such that their voltages can be read using the inbuilt ADC and then mapped to obtain a PWM waveform to control the direction and speed of two BO motors. This is very similar to Expt 4A-4 – Analog Input, Serial Output, where the voltage from the potentiometer was used to control the intensity of an LED. In Expt 4 only the intensity of the LED was varied. Here we shall control both speed and direction. The mapped values of the Node MCU will be sent over SPI to the Arduino Nano. SPI (Serial Peripheral Interface) is a synchronous serial data protocol used by Microcontrollers for communicating with one or more peripheral devices quickly over short distances. Here SPI is used for communication between Node MCU and Arduino Nano, with Node

MCU as the controller and Arduino Nano as the peripheral.

In Part C we shall introduce one more microcontroller card, viz. ESP-WROOM32. The two potentiometers used in Part B will now be connected to ESP-WROOM32 so as to read their voltages using the inbuilt ADC. In this case the mapped values will be sent from ESP-WROOM32 over Wi-Fi to the Node MCU, which in turn will pass on these parameters to the Arduino Nano.

Part A - Interfacing two BO motors through Arduino Nano board and L298 driver card

A.1 Procedure

i) Place the Arduino Nano on the breadboard such that pins are seated on the opposite sides of the central dividing line of breadboard, ensuring that the gap prevents any short circuit between pins.

ii) Connect the pins of Arduino with the Motor driver as below:

Motor driver L298	Arduino Nano
ENA	D3
IN1	D4
IN2	D6
IN3	D5
IN4	D8
ENB	D9

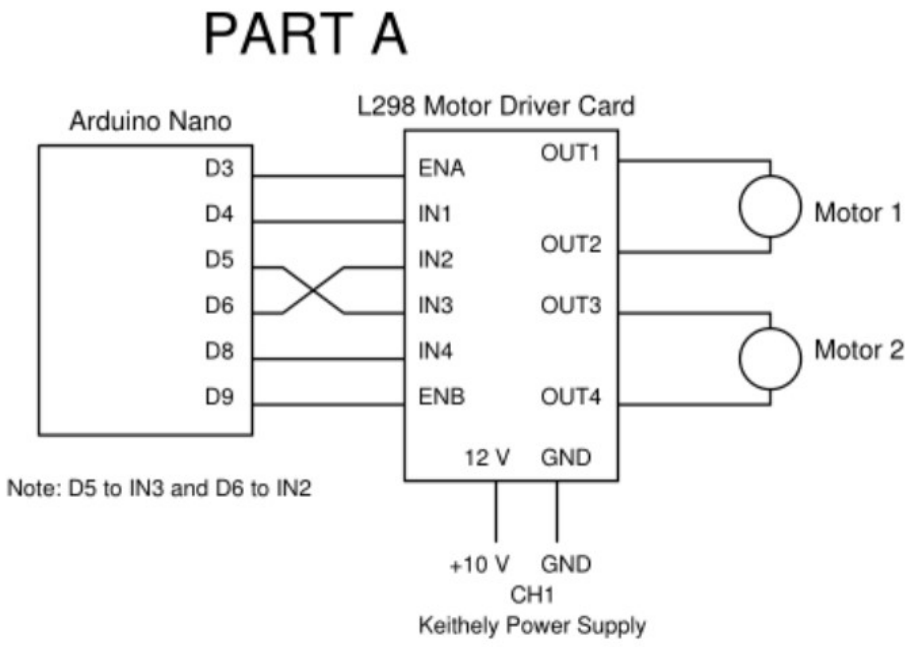


Figure 1: PartA- Circuit Diagram

iii) Verify that Keithley Power Supply CH1 is set to 10V. Ensure that the power supply outputs are disabled (or OFF). In the OFF mode, the green LED of the **'Output On/Off'** button will not glow.

iv) Connect the Powersupply CH1 output to the 12 V pin of Motor driver L298 and GND of CH1 to GND of Motor driver Card.

v) Connect **First Motor** to the **OUT1 & OUT2** and **Second Motor** to the **OUT3 & OUT4** pins of the Motor driver L298.

vi) Connect CH1 and CH2 DSO probes to the **D3 & D9** pins of Arduino Nano. Make sure that DSO GND pins of the CH1 & CH2 are connected to the common Ground of the Arduino Nano.

vii) Connect Arduino Nano to your PC and upload the **PartA_Arduino** from Arduino IDE. Observe PWM pulse signals of both channels in the DSO. Sketch the PWM Waveforms for one time instant.

viii) Power Supply output should be made ON only after your TA checks your Circuit.

Observations

1. After your circuit has been verified by your TA, put the Keithley Power Supply Output in the ON mode by pressing the 'Output On/Off' button. (In the ON mode, the LED will glow).
2. Now you should be able to see the motor rotating in one direction. After some time (depends on delay specified in the code) it will stop and will start rotating in the opposite direction.

Part B - Controlling the speed of two BO Motors through Arduino Nano with control from a Node MCU (ESP8266 MOD) board

B.1 Procedure

i) Turn OFF the Keithley Power Supply outputs.

ii) This experiment extends the work you completed in Part-A. Maintain the Connections between the Arduino Nano and L298 Motordriver that were established in Part-A.

iii) Place the Node MCU(ESP8266MOD) alongside Arduino Nano on the breadboard such that pins are seated on the opposite sides of the central dividing line of breadboard.

iv) Place the $10\,k\Omega$ Potentiometer on the breadboard.

v) Connect Keithley CH1 output to the 12V pin of Motor driver card and GND of CH1 to GND of Motor driver card. Set CH1 output to 10V.

vi) Connect CH2 output to Pin **VIN** of Arduino Nano and GND of CH2 to Pin **GND** of Arduino Nano and set the CH2 to 5V.

vii) Connect CH3 output to Pin **Vin** of ESP8266MOD and GND of CH3 to Pin **GND** of ESP8266MOD and set CH3 to 3.35V.

Note: The operating voltage of the ESP8266MOD module may range from 3.3V to 3.5V, depending on the specific hardware. If the motors fail to operate after the power supply is activated, please incrementally adjust the voltage, beginning at 3.35V.

viii) Connect the Pins of Arduino Nano with Pins of ESP8266MOD as below

Arduino Nano	ESP8266 MOD
D10	D8
D11	D7
D12	D6
D13	D5
GND	GND

ix) Connect the wiper (center pin) of the $10\,k\Omega$ Potentiometer to the analog input pin (Pin **A0**) of the ESP8266MOD. Connect one of the outer terminals of the potentiometer to the 3.3V (Pin **3V3)** supply pin and the other outer terminal to the ground (Pin **GND**) of the ESP8266MOD.

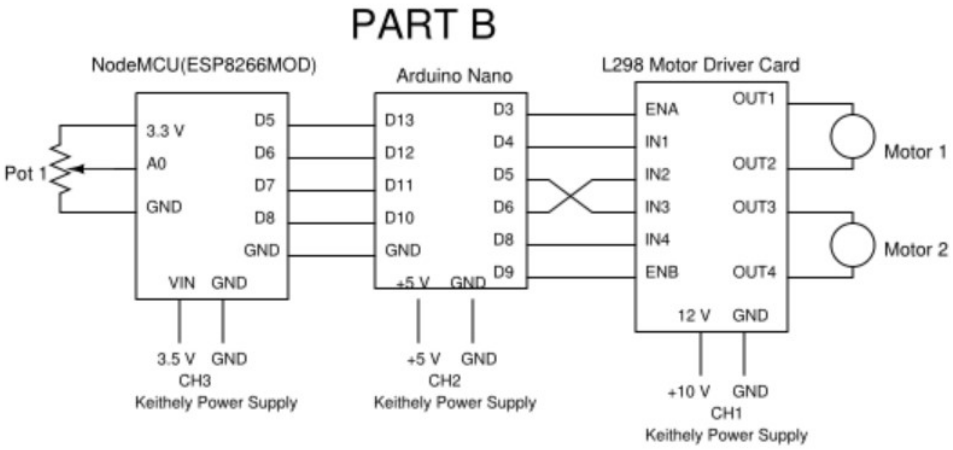


Figure 2: PartB- Circuit Diagram

x) Connect Arduino to your PC and Upload the **PartB_Arduino** Code to the Arduino from Arduino IDE. Ensure that you select "Arduino Nano" as the board type in the IDE.

xi) After uploading is done, connect your PC to ESP8266MOD and upload the **PartB_ESP8266MOD** code to the ESP8266MOD. Ensure that you select "Generic ESP8266" as the board type in the IDE and turn off the Power Supply while uploading.
Note: If the port is not detected after connecting to the ESP8266 module, please follow these steps: Click here to download and install the CP210x driver. Extract the downloaded folder. Navigate to the marked folder as shown in the Figure 4 and proceed with the installation.

xii) After your circuit has been verified by your TA, put the Keithley Power Supply Output in the ON mode by pressing the **'Output On/Off'** button.Ensure that the correct supply settings are configured for all three channels before turning the output on.

Observations

(a) After turning on the power supply, you should be able to observe the motors beginning to rotate in one direction. The speed and direction of the motors can be controlled by adjusting the potentiometer.

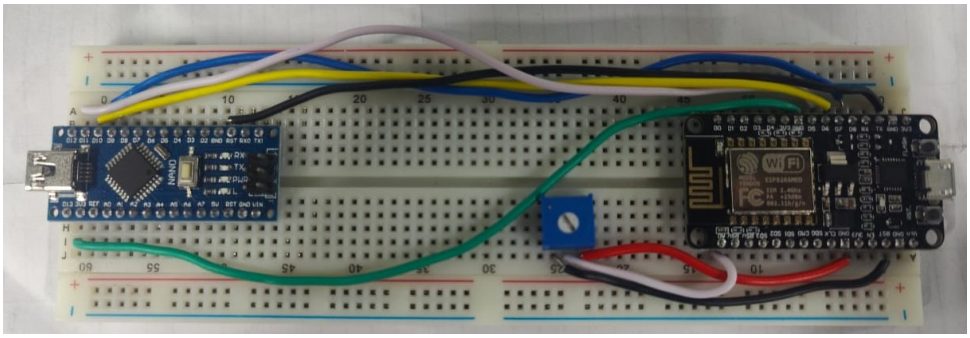


Figure 3: PartB-NodeMCU to Arduino Nano connections

Note:There may be a parallax error in the Figure 3 regarding the connections. This circuit image is provided for reference purposes only. Please follow the steps outlined in the procedure for correct pin connections.

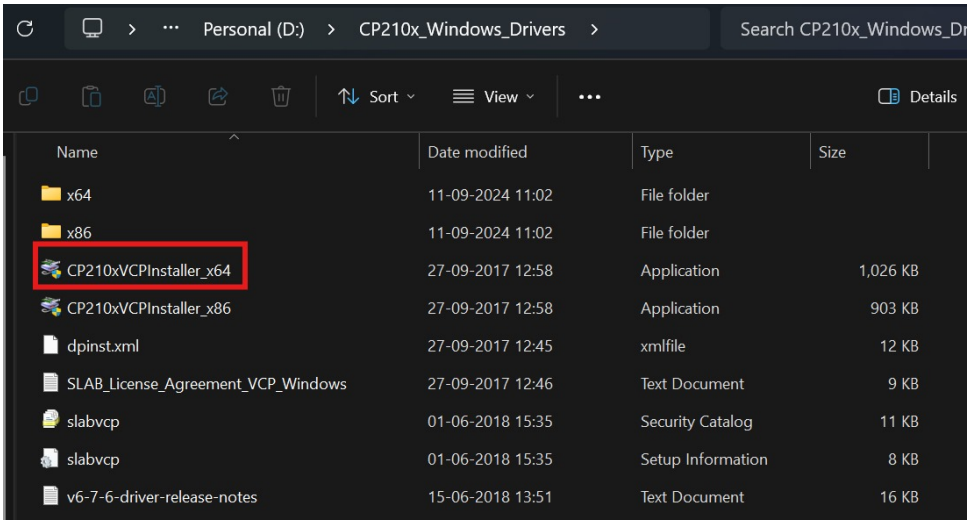


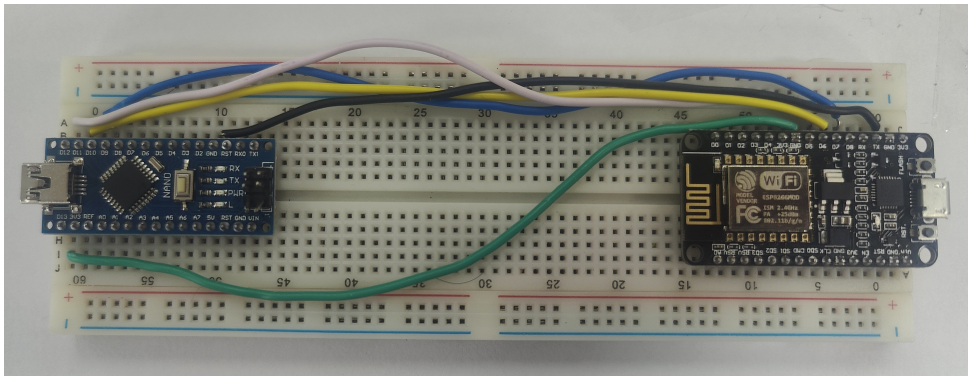
Figure 4: CP210XVCP installation

Part C-Remote speed control of two BO motors using ESP-WROOM32, Node MCU (ESP8266 MOD) and Arduino Nano boards

C.1Procedure

i) Turn OFF the Keithley Power Supply outputs.

ii) This experiment builds upon the work completed in Part B. Disconnect and remove the 10 kΩ potentiometer from the circuit assembled in Part B, while keeping the power supply settings unchanged from those used in Part B. The existing circuit configuration will be utilized for this section. However, do not remove the motor connections, as they will be used in this experiment.



iii) Position the ESP-WROOM32 on the new breadboard. Then, place the two 10 kΩ potentiometers on the same section of the breadboard as shown in Figure 6, where the ESP-WROOM32 is located.

iv) Connect the 5V output from the CH2 of Keithley power supply to the top horizontal row labeled as 5V in the figure3.

v) Connect the 5V from the top horizontal row to the INPUT pin of the LM1117 voltage regulator and to the VIN pin of the ESP-WROOM32. Connect the GND from the top horizontal row to the GND pins of both the LM1117 voltage regulator and the ESP-WROOM32.

vi) Connect the 3.3V output from the LM1117 voltage regulator to one terminal of each of the two potentiometers. Connect the ground (GND) to the remaining terminals of both potentiometers.

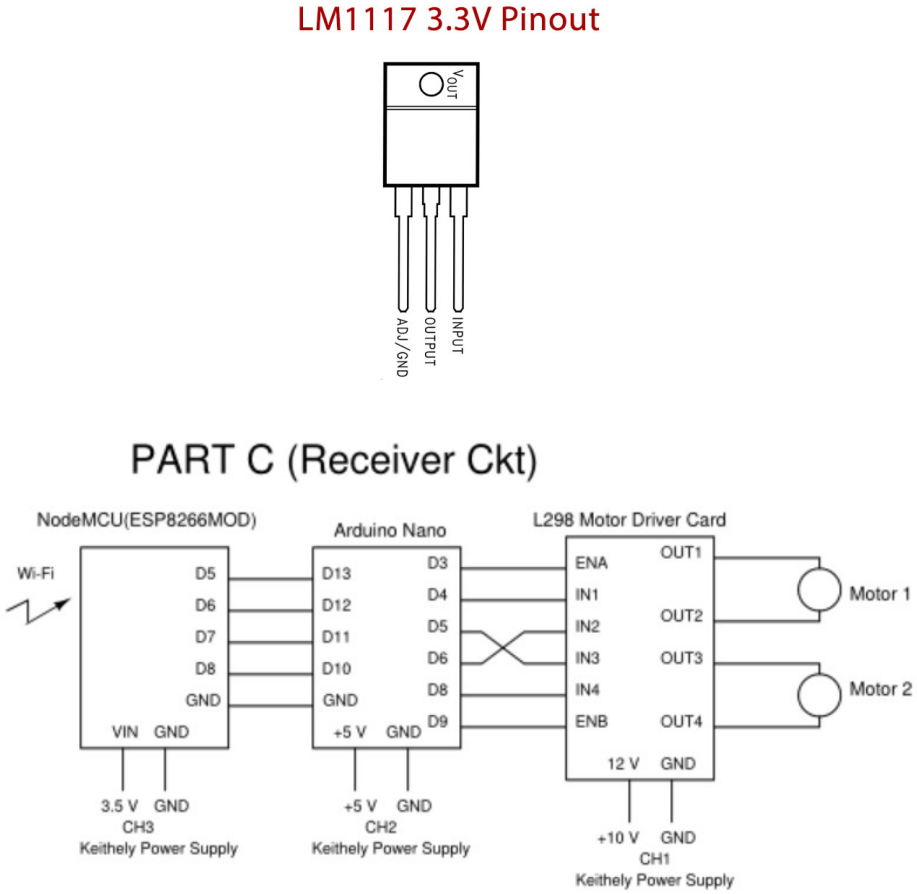


Figure 5: PartC- Receiver Circuit Diagram

vii) Connect the wiper (center pin) of the one of the potentiometer to Pin **D34** of the ESP-WROOM32.Connect the wiper (center pin) of the Other potentiometer to Pin **D32** of the ESP-WROOM32.

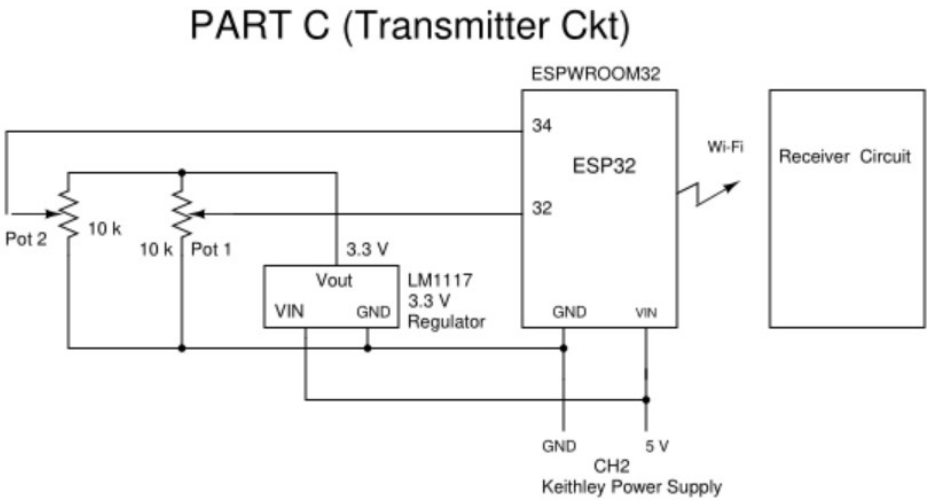
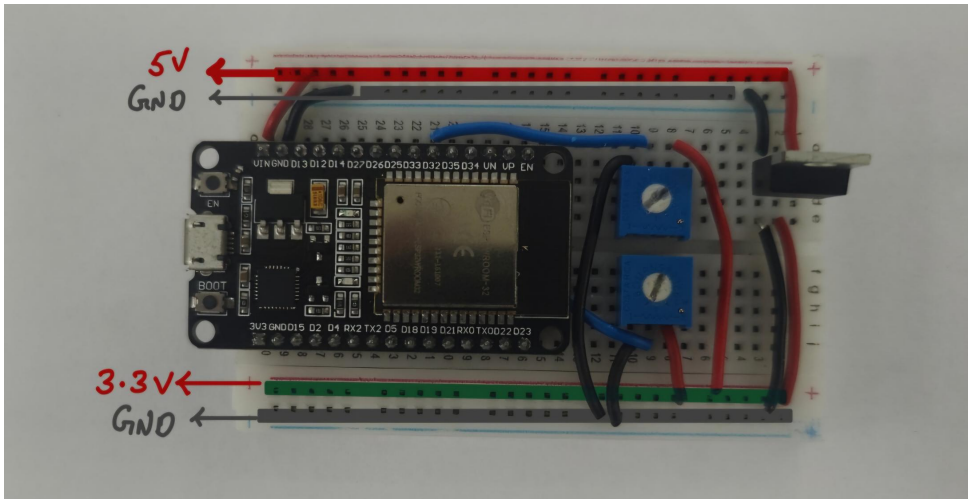


Figure 6: PartC- Transmitter Circuit Diagram

viii) Connect Arduino Nano to your PC and upload file named **PartC_Arduino** code to the



Arduino Nano.

ix) After the upload is completed, connect your PC to the ESP8266MOD(Node MCU) and open the file named **PartC_ESP8266MOD**. In this code, update the the SSID and serverAddress fields with the assigned values. Replace the **ZZ** in the SSID field with your Table No (Ex:D2,K3,...) and **Y** in the serverAddress field to your assigned value given by your TA. Upload this updated code to ESP8266MOD.

```
// Network credentials
// Your ssid and ip_address,both password of ESP32 and ESP8266 should match
const char* ssid = "Your_Group_ZZ."; //  //Change "ZZ" to your Table No
const char* password = "Your_Password ";// Password should be any 8 digit numerical valve

// Server address
const char* serverAddress = "http://192.168.4.Y"; // Change value of Y to your assigned value which is same IN ESP32
```

Figure 7: ESP8266MOD serverAddress and SSID

x) Upon successful completion of the upload to the ESP8266MOD, connect your PC to the Eand select the board **ESP32-WROOM-DA-Module** and open the **PartC_ESP32** code. Update the IP address and SSID fields with the assigned values. Replace the **Y** in the IP address field to your assigned value given by your TA and Replace **ZZ** in the SSID field with your Table No (Ex:D2,K3,...) and upload to ESP-WROOM32. Once the code has been successfully uploaded to all devices (Arduino Nano, ESP8266MOD, and ESP-WROOM32), disconnect your PC from all devices.

```
// Network credentials
// Your ssid and ip_address,both password of ESP32 and ESP8266 should match
const char* ssid = "Your_Group_ZZ"; //Change "ZZ" to your Table No
const char* password = "Your_password";// Change this to any 8 digit numerical value

// Static IP address configuration
IPAddress local_IP(192, 168, 4, Y); // Change value of Y to your assigned value
IPAddress gateway(192, 168, 4, 1);
IPAddress subnet(255, 255, 255, 0);
```

Figure 8: ESP-WROOM32 IPAddress and SSID

Note:

i. **Replace "Your_password" with any 8-digit numeric value, without including alphabetic characters or special symbols.Ensure that these values are consistently matched in both the ESP-WROOM32 and ESP8266 code configurations. Make sure that you select the correct board and port settings before proceeding with the upload.**

ii. **If the code fails to upload to the ESP32, press and hold the Boot button before initiating the upload process. Keep the button pressed until the upload begins successfully.**

iii. **If a system error occurs during the upload process, restart your system to resolve the error and then attempt the upload again.**

iv. **After successfully uploading the code to the ESP32, the blue LED should illuminate. If the LED does not light up, press the EN (Enable) button on the ESP32 to reset the device.**

xi) After your TA verifies your circuit turn on power supply.

Observations

i. After turning on the power supply, you should observe that the motors begin to rotate. To control the motors, use the potentiometers connected to the ESP-WROOM32. By adjusting these potentiometers, you should be able to control both the speed and direction of the motors.