Tinkering Project

Group: Wed#1

Janhvi Soni	2022MEB1317
Mili Prajapati	2022MEB1325
Gandhe Harshitha	2022MEB1308
Booma Haripriya	2022MEB1304

Aim

To implement IOT Controlled Boat by using Blynk/radio transmitter.

Materials Required

ESP32 Development Board

Motor Driver (L298N)

DC Motor

Hull/Frame

Wi-Fi Antenna

LED, Resistor, Wires, Switch, Tape.

Breadboard

Propeller

Procedure

The project involves creating a wireless boat controlled via the Blynk app, utilizing an ESP32 development board. The boat's propulsion system, comprising two DC motors for the propellers, is managed by an L298N motor driver. Power is supplied and controlled by a rechargeable battery and switch integrated into the system.

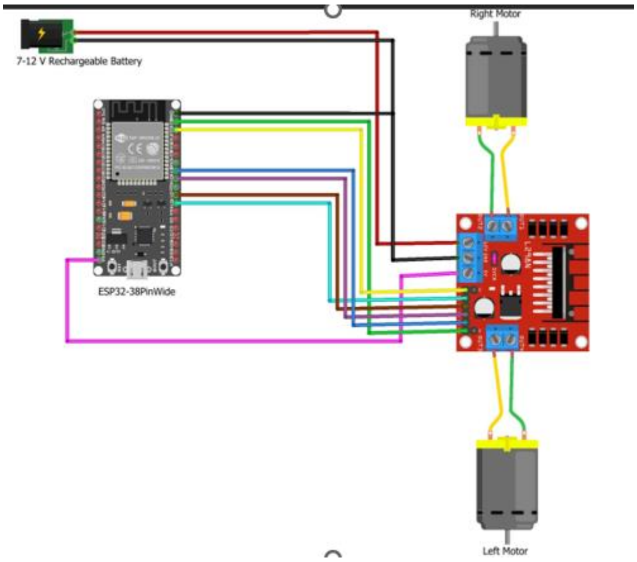

Fig. 1 - Illustrates the detailed circuit wiring used in the setup.

Following the circuit assembly on the boat frame, a C++ program is developed in the Arduino IDE and subsequently uploaded to the ESP32 development board. This program enables the boat to establish a Wi-Fi connection for communication with the Blynk app. All operational commands are then transmitted seamlessly through the Blynk app interface, facilitating remote control and monitoring of the boat's movements and functions.

Blynk Setup

On the Blynk console, a new device is created that provides a token ID, essentially the address of the Blynk cloud server. This token ID is added to the Arduino IDE code. Four virtual pin DataStream are created in the console, named forward, backward, left, and right

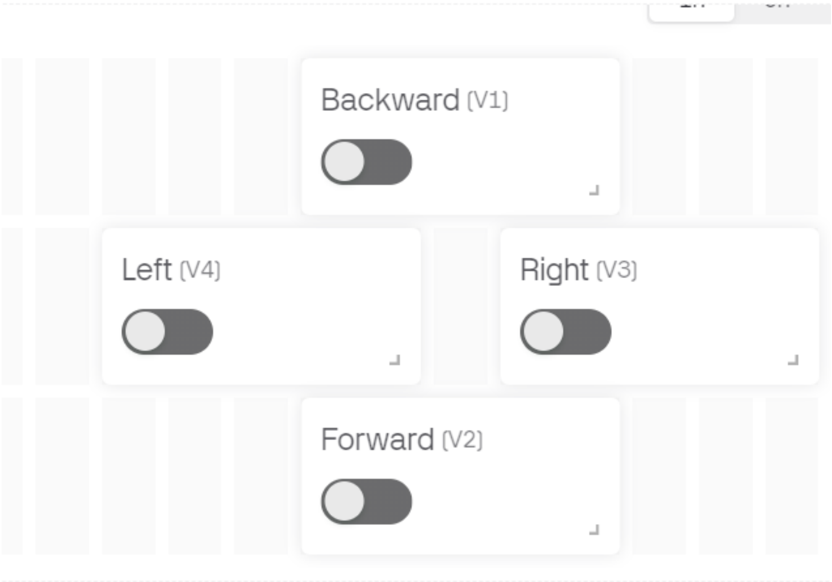

Fig.2 – Buttons on the blynk app

<u>Arduino IDE setup :</u>

The following code is written in the ide and is uploaded in the ESP 32.

```
#define BLYNK_TEMPLATE_ID "TMPL3S7TnIsXA"
#define BLYNK_TEMPLATE_NAME "RC Boat"
#define BLYNK_AUTH_TOKEN "68LHUnBbJSfMVQt4ITRgX81PGPSE2ndm"
  // token provided by the blynk

#include <WiFi.h>
#include <WiFiClient.h>
#include <BlynkSimpleEsp32.h>

// Motor control pins
const int motor1A = 4; //IN1
const int motor1B = 5; //IN2
const int motor1E = 0;  // ENA Pin

const int motor2A = 12; //IN4
const int motor2B = 13; //IN3
const int motor2E = 2;  // ENB Pin

// Enter your Auth token
char auth[] = BLYNK_AUTH_TOKEN;               //credentials of blynk cloud server

// Enter your WIFI SSID and password
char ssid[] = "vivo Y20G";
char pass[] = "19212807";

void setup() {
  // Debug console
  Serial.begin(9600);

  // Initialize motor control pins as outputs
  pinMode(motor1A, OUTPUT);
  pinMode(motor1B, OUTPUT);
  pinMode(motor1E, OUTPUT);

  pinMode(motor2A, OUTPUT);
  pinMode(motor2B, OUTPUT);
  pinMode(motor2E, OUTPUT);
  digitalWrite(motor1A,LOW); //intially all pin zero
  digitalWrite(motor1B,LOW);
  digitalWrite(motor2A,LOW);
  digitalWrite(motor2B,LOW);

  // Connect to Wi-Fi
  WiFi.begin(ssid, pass);
  while (WiFi.status() != WL_CONNECTED) {
    delay(1000);
    Serial.println("Connecting to WiFi...");
  }
  Serial.println("Connected to WiFi");

  // Connect to Blynk
  Blynk.begin(auth, ssid, pass, "blynk.cloud", 80);
}

void loop() {
  Blynk.run();
```

```
}

// Blynk Virtual Pin handler for button states
BLYNK_WRITE(V1) {

  int buttonState = param.asInt(); // Get value from Blynk app (0 for OFF, 1 for ON)

  if (buttonState == 1) {
    // Forward
    digitalWrite(motor1A, HIGH);//IN1 high(motor clockwise)
    digitalWrite(motor1B, LOW);//IN2 low(anticlockwise = 0)
    analogWrite(motor1E, 255);  // Full speed

    digitalWrite(motor2A, HIGH); //IN4 high (motor clockwise)
    digitalWrite(motor2B, LOW); //IN3 Low (anticlockwise)
    analogWrite(motor2E, 255);  // Full speed

    //"Backwards" // was forward before, (backward after)
  } else {
    // Stop
    digitalWrite(motor1A, LOW);
    digitalWrite(motor1B, LOW);
    analogWrite(motor1E, 0);  // Stop

    digitalWrite(motor2A, LOW);
    digitalWrite(motor2B, LOW);
    analogWrite(motor2E, 0);  // Stop

    //"Stop  "
  }
}

BLYNK_WRITE(V2) {
  int buttonState = param.asInt();

  if (buttonState == 1) {
    // Backward
    digitalWrite(motor1A, LOW);
    digitalWrite(motor1B, HIGH); // IN2 high (anticlockwise)
    analogWrite(motor1E, 255);  // Full speed

    digitalWrite(motor2A, LOW);
    digitalWrite(motor2B, HIGH);// IN3 high (anticlockwise)
    analogWrite(motor2E, 255);  // Full speed

    //"Forward" //was backward first,changed to forwad
  } else {
    // Stop
    digitalWrite(motor1A, LOW);
    digitalWrite(motor1B, LOW);
    analogWrite(motor1E, 0);  // Stop

    digitalWrite(motor2A, LOW);
    digitalWrite(motor2B, LOW);
    analogWrite(motor2E, 0);  // Stop

    //"Stop"
  }
```

```
}

BLYNK_WRITE(V3) {
  int buttonState = param.asInt();

  if (buttonState == 1) {
    // Right
    digitalWrite(motor1A, HIGH); //IN1 high (clockwise)
    digitalWrite(motor1B, LOW);
    analogWrite(motor1E, 255);  // Full speed

    digitalWrite(motor2A, LOW);
    digitalWrite(motor2B, HIGH); //IN3 high (anticlockwise)
    analogWrite(motor2E, 255);  // Full speed

    //"Right"
  } else {
    // Stop
    digitalWrite(motor1A, LOW);
    digitalWrite(motor1B, LOW);
    analogWrite(motor1E, 0);  // Stop

    digitalWrite(motor2A, LOW);
    digitalWrite(motor2B, LOW);
    analogWrite(motor2E, 0);  // Stop

    //"Stop"
  }
}
```

```
BLYNK_WRITE(V4) {
  int buttonState = param.asInt();

  if (buttonState == 1) {
    // Left
    digitalWrite(motor1A, LOW);
    digitalWrite(motor1B, HIGH); //IN2 high (anticlockwise)
    analogWrite(motor1E, 255);  // Full speed

    digitalWrite(motor2A, HIGH); // IN4 high (clockwise)
    digitalWrite(motor2B, LOW);
    analogWrite(motor2E, 255);  // Full speed

    //"Left"
  } else {
    // Stop
    digitalWrite(motor1A, LOW);
    digitalWrite(motor1B, LOW);
    analogWrite(motor1E, 0);  // Stop

    digitalWrite(motor2A, LOW);
    digitalWrite(motor2B, LOW);
    analogWrite(motor2E, 0);  // Stop

    //"Stop")
  }
}
```

Final Boat