Some Notes

On

DAY-6 ESP32: Inatallation and Usage

By

PRASAMSHA ADHIKARI
PUR080BCT060

To

ROBOTICS CLUB-Purwanchal Campus

Tribhuwan University

Institute of Engineering

Robotics Club

Purwanchal Campus

Dharan, Nepal

2025/07/05

1 Session in Summary

1.1 Introduction

The sixth day of 8-day interactive learning event organized by Robotics club of Purwanchal Campus —> TECHNOMORPH got more interesting and even more productive like other days. We learned with a lot of excitement and efficiency.

2 What We Learned

We got to use ESP32 as our main tool. Installation of cp210x drivers, ESP32 support on our PC to support the environment in which we worked on. We used Arduino IDE to run our code and did real-time simulation.

Connecting the device to Bluetooth and WiFi respectively, and turning on and off the LED accordingly using a mobile phone was such a wholesome experience. The session was great, and we're hoping it ends with more concepts and information.

3 Set-Ups Made

1. Arduino ide was pre-installed(at the beginning of the session)
2. CP210X driver was installed for windows(UART) version
3. In IDE, >file >preferences >additional board manager URIs was changed to "https://dll.espressif.com/dll/package/dll/package_esp32_index.json,http://arduino.esp8266.com/stable/package_e"
4. In IDE, >Tools >Preferences >Boards >Boards Manager >Install Arduino ESP32 boards by espressif
5. Mobile App-Serial Bluetooth Terminal was installed

Additionally,

1. Type C data cable was connected to ESP32 board and LED was connected to the board.

3.1 Code for Bluetooth

```
#include "BluetoothSerial.h"

BluetoothSerial SerialBT;

const int ledPin = 5;              // LED connected to GPIO 5
const bool LED_ACTIVE_LOW = true;  // Flip logic if LED is active-low

String device_name = "poco";
String incomingMessage = "";

void setup() {
  Serial.begin(115200);
  pinMode(ledPin, OUTPUT);
  digitalWrite(ledPin, LED_ACTIVE_LOW ? HIGH : LOW); // Start with LED OFF

  SerialBT.begin(device_name);
  Serial.printf("Bluetooth device \"%s\" started.\nPair and send 'on' or
'off'.\n", device_name.c_str());
}

void loop() {
  // Relay from serial monitor to Bluetooth
  if (Serial.available()) {
    SerialBT.write(Serial.read());
  }

  // Read from Bluetooth
  if (SerialBT.available()) {
    char c = SerialBT.read();

    // Collect characters into a message
    if (c == '\n' || c == '\r') {
      incomingMessage.trim();
      Serial.printf("Received via Bluetooth: %s\n", incomingMessage.c_str());

      if (incomingMessage.equalsIgnoreCase("on")) {
        digitalWrite(ledPin, LED_ACTIVE_LOW ? LOW : HIGH);
        Serial.println("LED turned ON");
      } else if (incomingMessage.equalsIgnoreCase("off")) {
        digitalWrite(ledPin, LED_ACTIVE_LOW ? HIGH : LOW);
        Serial.println("LED turned OFF");
      }

      incomingMessage = ""; // clear for next command
    } else {
      incomingMessage += c;
    }
  }

  delay(10);
}
```

3.2 Code for WiFi

```
#include <WiFi.h>

const char* ssid = "apple";
const char* password = "appleball";

WiFiServer server(80);

const int ledPin = 5;

void setup() {
  Serial.begin(115200);
  pinMode(ledPin, OUTPUT);
  digitalWrite(ledPin, HIGH); // active-low: HIGH = OFF

  WiFi.begin(ssid, password);
  Serial.print("Connecting to WiFi");
  while (WiFi.status() != WL_CONNECTED) {
    delay(500);
    Serial.print(".");
  }

  Serial.println("\nWiFi connected. IP address:");
  Serial.println(WiFi.localIP());

  server.begin();
}

void loop() {
  WiFiClient client = server.available();
  if (!client) return;

  Serial.println("New Client.");
  String currentLine = "";

  while (client.connected()) {
    if (client.available()) {
      char c = client.read();
      currentLine += c;
      Serial.write(c);

      if (c == '\n') {
        if (currentLine.indexOf("GET /H") >= 0) {
          digitalWrite(ledPin, LOW);  // ON (active-low)
        } else if (currentLine.indexOf("GET /L") >= 0) {
          digitalWrite(ledPin, HIGH); // OFF (active-low)
        }

        client.println("HTTP/1.1 200 OK");
        client.println("Content-Type: text/html");
        client.println();
        client.println(R"rawliteral(
          <!DOCTYPE html>
          <html>
          <head>
            <meta name="viewport" content="width=device-width, initial-
scale=1">
            <title>ESP32 LED Control</title>
          </head>
          <body>
            <h1>ESP32 LED Control</h1>
            <l><a href="/H">Turn LED ON</a></l></br>
            <l><a href="/L">Turn LED OFF</a></l>
          </body>
          </html>
        )rawliteral");
        break;
      }
    }
  }

  delay(1);
  client.stop();
  Serial.println("Client Disconnected.");
}
```

3.3 Code working

For WiFi(hostspot):

We run the code is IDE and in serial monitor the IP address was displayed, on opening that address, there were two options LED turn ON and LED turn OFF, since the signal was active low, when we clicked turn ON, LED was off and vice-versa.

For Bluetooth:

App installed at the beginning was opened, bluetooth was connected to the mobile phone and LED was tested to turn on and off and code above worked efficiently in same active-low mode.

4 Devices used:

1. ESP32
2. Jumper Wires
3. Type-C data cable
4. LED

Figure 1: Logo of Tribhuvan University.