

Embedded Systems Laboratory

- Lap10:
- มีความรู้ความเข้าใจในรายละเอียดของ Wifi ของ ESP32
 - การโปรแกรมเพื่อควบคุมการทำงานโดยใช้ Wifi
 - การโปรแกรมประยุกต์ในการใช้งาน Wifi ของ ESP32

10.1 ข้อมูลเบื้องต้น Wifi ของ ESP32

ให้นักศึกษาร่วมกันหาข้อมูลเพื่อนำมาตอบคำถามข้างล่างดังนี้

คำถาม	คำตอบ
ESP32 รองรับมาตรฐานการทำงานของระบบเครือข่ายไร้สาย IEEE ไດบ้าง	802.11 b/g/n
Wifi ของ ESP32 รองรับคลื่นความถี่ใด? และ ความเร็วสูงสุดเท่าใด?	2.4 GHz / 150 Mbps
การใช้งาน Wifi ใน ESP32 สามารถใช้งานได้กี่โหมด? อะไรบ้าง?	3 mode 1.AP 2.STA 3.AP+STA
ถ้าต้องการให้ ESP32 เชื่อมต่อ Internet จาก Access Point ของ KUWIN ต้องใช้โหมดใด? เพราะอะไร?	STA เพราะเป็น mode ที่ให้ esp32 ไปเชื่อมต่อเข้ากับอุปกรณ์อื่น ๆ
ถ้าต้องการให้ ESP32 มี Web Server แล้วให้มือถือมาเชื่อมต่อเพื่อควบคุมอุปกรณ์ต่างๆ ต้องใช้โหมดใด? เพราะอะไร?	AP เพราะเป็น mode ที่ให้อุปกรณ์ใดๆไปเชื่อมต่อเข้ากับ esp32

จงอธิบายการทำงาน Function Wifi ใน ArduinoIDE

Function	คำตอบ
ต้องนำเข้า Lib ไต - ยกตัวอย่างการนำไปใช้งาน	#include <WiFi.h>
void mode(.....); - ยกตัวอย่างการนำไปใช้งาน	WiFi.mode(WIFI_AP);// AP mode
bool softAP(.....); - ยกตัวอย่างการนำไปใช้งาน	WiFi.softAP(ssid, password);

bool softAPConfig(.....); - ยกตัวอย่างการนำไปใช้งาน	WiFi.softAPConfig(local_ip, gateway, subnet);
bool softAPdisconnect(); - ยกตัวอย่างการนำไปใช้งาน	WiFi.softAPdisconnect();
void begin(....); - ยกตัวอย่างการนำไปใช้งาน	WiFi.begin(ssid, password);
bool reconnect(); - ยกตัวอย่างการนำไปใช้งาน	WiFi.reconnect();
bool disconnect(); - ยกตัวอย่างการนำไปใช้งาน	WiFi.disconnect();

10.2 ESP32 STA Mode

ให้นิสิต ศึกษาการใช้งาน Wifi ของ ESP32 ในโหมด STA

ทดลองเขียนโค้ดเพื่อทดสอบการทำงานของ ESP32

```
#include <WiFi.h>
const char* ssid      = "YOUR_SSID";
const char* password  = "YOUR_PASS";
void setup()
{
    Serial.begin(115200);
    //connect to WiFi
    Serial.printf("Connecting to %s ", ssid);
    WiFi.begin(ssid, password);
    while (WiFi.status() != WL_CONNECTED) {
        delay(500);
        Serial.print(".");
    }
    Serial.println(" CONNECTED");
}

void loop()
{
}
```

จากโค้ดด้านบน ให้ตอบคำถามต่อไปนี้ 1. นิสิตคิดว่าตัว Firmware ต้องการทำอะไร 2. ทำอย่างไรให้ ESP32 เชื่อมต่อ Internet ได้ 3. ผลลัพธ์ที่ได้จาก Firmware คืออะไร

1. เชื่อมต่อ wifi

2. กำหนดชื่อ wifi และรหัสผ่านที่จะต่อที่จะต่อ แล้วใช้คำสั่ง wifi.begin เพื่อเริ่มเชื่อมต่อ

3. Connecting to AndroidAP kit CONNECTED

10.3 ESP32 AP Mode

ให้นักศึกษาการใช้งาน Wifi ของ ESP32 ในโหมด AP

ทดลองเขียนโค้ดเพื่อทดสอบการทำงานของ ESP32

```
#include<WiFi.h>

#define WIFI_AP_NAME "YOUR_SSID"
#define WIFI_AP_PASS "YOUR_PASS"

void setup() {
  Serial.begin(115200);

  WiFi.mode(WIFI_AP);
  WiFi.softAP(WIFI_AP_NAME, WIFI_AP_PASS);

  Serial.print("IP address: ");
  Serial.println(WiFi.softAPIP());
}

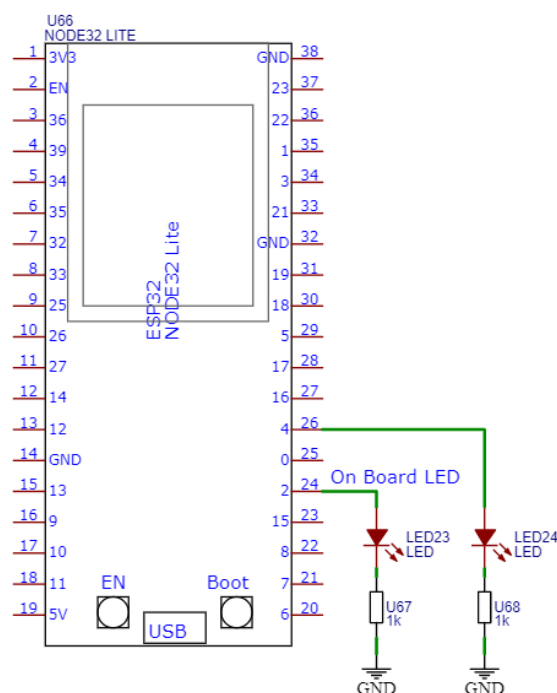
void loop() {
}
```

จากโค้ดด้านบน ให้ตอบคำถามต่อไปนี้ 1.นิสิตคิดว่าตัว Firmware ต้องการทำอะไร 2. ทำอย่างไรให้ ESP32 เชื่อมต่อ Internet ได้ 3.ผลลัพธ์ที่ได้จาก Firmware คืออะไร

1. ปลอ่ยสัญญาณสำหรับให้อุปกรณ์อื่นเชื่อมต่อ
- 2.กำหนดโหมดกำหนดชื่อ wifi และรหัสผ่านที่จะต่อให้อุปกรณ์อื่นเชื่อมต่อ แล้วใช้คำสั่ง softAP เพื่อเริ่มปลอ่ยสัญญาณ
3. **IP address: 192.168.4.1**

10.4 ESP32 Web Server

ให้นักศึกษา นำ ESP32 ต่อวงจรร่วมกับ LED ดังรูป



จากนั้นให้ทดลองเขียนโค้ด Webserver ดังนี้

```
#include <WiFi.h>
#include <WebServer.h>

const char* ssid = "ESP32";
const char* password = "12345678";

IPAddress local_ip(192, 168, 1, 1);
IPAddress gateway(192, 168, 1, 1);
IPAddress subnet(255, 255, 255, 0);

WebServer server(80);

uint8_t LED1pin = 2;
bool LED1status = LOW;

uint8_t LED2pin = 4;
bool LED2status = LOW;

void setup() {
  Serial.begin(115200);
  pinMode(LED1pin, OUTPUT);
  pinMode(LED2pin, OUTPUT);

  WiFi.softAP(ssid, password);
  WiFi.softAPConfig(local_ip, gateway, subnet);
  delay(100);

  server.on("/", handle_OnConnect);
  server.on("/led1on", handle_led1on);
  server.on("/led1off", handle_led1off);
  server.on("/led2on", handle_led2on);
  server.on("/led2off", handle_led2off);
  server.onNotFound(handle_NotFound);

  server.begin();
  Serial.println("HTTP server started");
}

void loop() {
  server.handleClient();
  if (LED1status)
  {
    digitalWrite(LED1pin, HIGH);
  }
  else
  {
    digitalWrite(LED1pin, LOW);
  }

  if (LED2status)
  {
    digitalWrite(LED2pin, HIGH);
  }
  else
  {
    digitalWrite(LED2pin, LOW);
  }
}

void handle_OnConnect() {
  LED1status = LOW;
  LED2status = LOW;
  Serial.println("GPIO2 Status: OFF | GPIO4 Status: OFF");
  server.send(200, "text/html", SendHTML(LED1status,
LED2status));
}

void handle_led1on() {
  LED1status = HIGH;
  Serial.println("GPIO2 Status: ON");
  server.send(200, "text/html", SendHTML(true,
LED2status));
}

void handle_led1off() {
  LED1status = LOW;
  Serial.println("GPIO2 Status: OFF");
  server.send(200, "text/html", SendHTML(false,
LED2status));
}
```

```
void handle_led2on() {
  LED2status = HIGH;
  Serial.println("GPIO4 Status: ON");
  server.send(200, "text/html", SendHTML(LED1status,
true));
}

void handle_led2off() {
  LED2status = LOW;
  Serial.println("GPIO4 Status: OFF");
  server.send(200, "text/html", SendHTML(LED1status,
false));
}

void handle_NotFound() {
  server.send(404, "text/plain", "Not found");
}

String SendHTML(uint8_t led1stat, uint8_t led2stat) {
  String ptr = "<!DOCTYPE html> <html>\n";
  ptr += "<head><meta name=\"viewport\"";
  content="width=device-width, initial-scale=1.0, user-
scalable=no">\n";
  ptr += "<title>LED Control</title>\n";
  ptr += "<style>html { font-family: Helvetica; display:
inline-block; margin: 0px auto; text-align: center;}\n";
  ptr += "body{margin-top: 50px;} h1 {color:
#444444;margin: 50px auto 30px;} h3 {color:
#444444;margin-bottom: 50px;}\n";
  ptr += ".button {display: block;width: 80px;background-
color: #3498db;border: none;color: white;padding: 13px
30px;text-decoration: none;font-size: 25px;margin: 0px
auto 35px;cursor: pointer;border-radius: 4px;}\n";
  ptr += ".button-on {background-color: #3498db;}\n";
  ptr += ".button-on:active {background-color:
#2980b9;}\n";
  ptr += ".button-off {background-color: #34495e;}\n";
  ptr += ".button-off:active {background-color:
#2c3e50;}\n";
  ptr += "p {font-size: 14px;color: #888;margin-bottom:
10px;}\n";
  ptr += "</style>\n";
  ptr += "</head>\n";
  ptr += "<body>\n";
  ptr += "<h1>ESP32 Web Server</h1>\n";
  ptr += "<h3>Using Access Point(AP) Mode</h3>\n";

  if (led1stat)
  {
    ptr += "<p>LED1 Status: ON</p><a class=\"button
button-off\" href=\"/led1off\">OFF</a>\n";
  }
  else
  {
    ptr += "<p>LED1 Status: OFF</p><a class=\"button
button-on\" href=\"/led1on\">ON</a>\n";
  }

  if (led2stat)
  {
    ptr += "<p>LED2 Status: ON</p><a class=\"button
button-off\" href=\"/led2off\">OFF</a>\n";
  }
  else
  {
    ptr += "<p>LED2 Status: OFF</p><a class=\"button
button-on\" href=\"/led2on\">ON</a>\n";
  }

  ptr += "</body>\n";
  ptr += "</html>\n";
  return ptr;
}
```

จากโค้ดด้านบน ให้ตอบคำถามต่อไปนี้ 1.นิสิตคิดว่าตัว Firmware ต้องการทำอะไร 2. ผลลัพธ์ที่ได้จาก Firmware คืออะไร

1.ควบคุมการเปิดปิด LED โดยใช้อุปกรณ์ที่เชื่อมต่อควบคุม

2. `HTTP server started`

`GPIO4 Status: OFF`

`GPIO2 Status: OFF | GPIO4 Status: OFF`

`GPIO2 Status: ON`

`GPIO2 Status: OFF`

`GPIO4 Status: ON`

`GPIO4 Status: OFF`

`GPIO2 Status: ON`

`GPIO2 Status: ON`

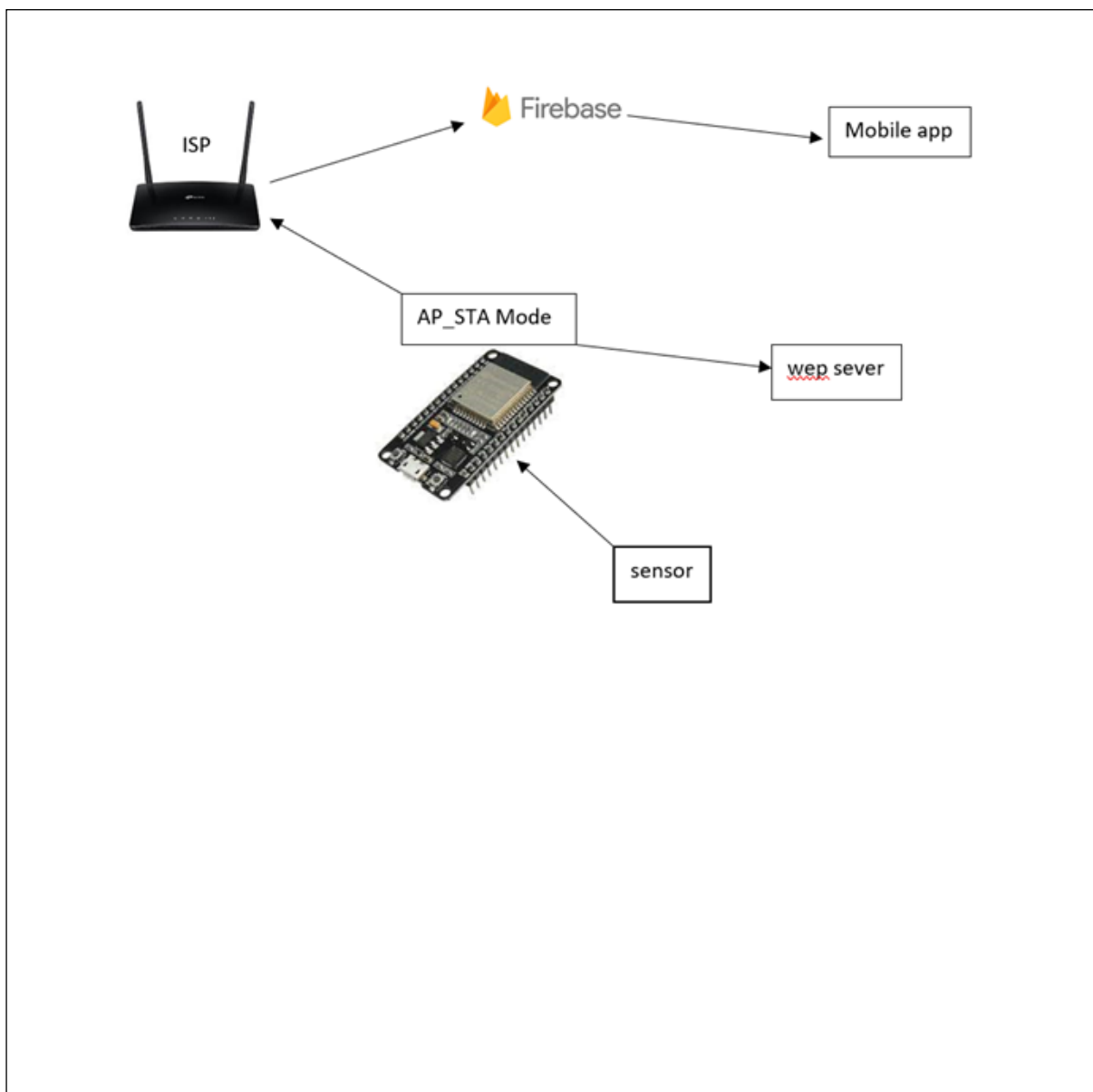
`GPIO2 Status: OFF`

10.5 Design topology

ให้นิสิตวิเคราะห์จากโจทย์ต่อไปนี้

ต้องการพัฒนาอุปกรณ์ที่วัดค่าฝุ่น PM 2.5 ซึ่งตัวอุปกรณ์จะส่งข้อมูลไปยัง Database(Firebase) และผู้ใช้สามารถดูค่าฝุ่นแบบ Realtime ได้ 2 ทาง คือ 1.แอปพลิเคชันในมือถือ 2.สามารถต่อ Wifi Direct จากอุปกรณ์โดยตรง

ให้นิสิตวาดรูปแบบการเชื่อมต่อ หรือ Diagram ที่อธิบายการทำงานของโจทย์โดยละเอียด



10.6 โจทย์ Assignment Wifi

ให้นักศึกษาเขียนโปรแกรมให้ ดังนี้

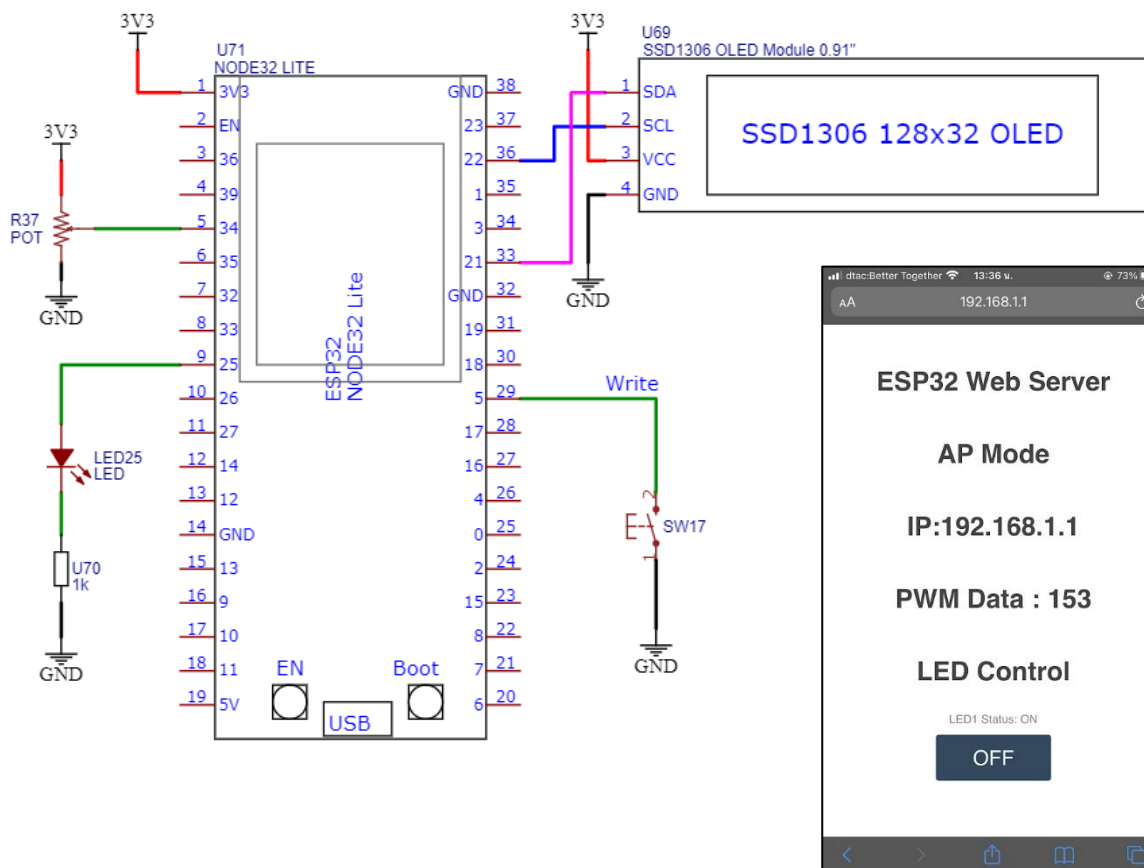
การทำงาน

- เมื่อ MCU เริ่มต้นการทำงาน ให้ อ่านค่าของ ADC ล่าสุดที่บันทึกไว้ใน EEPROM (ตำแหน่งที่บันทึก ให้ บันทึกตามรหัสสี 3 ตัวท้าย) และนำไปแสดงผลกับ LED โดยใช้ PWM
- มีการเก็บค่าการเปิด/ปิด ไว้ใน EEPROM โดยค่าการเปิด/ปิด ให้บันทึกในตำแหน่ง 0 (0x000)
- เมื่อกดปุ่ม SW17 จะอ่านค่า ADC จาก POT (R37) โดยการปรับ ให้เหมาะสมกับการนำไป Drive PWM ของ LED
- ใช้ Wifi ในโหมด AP โดยตั้ง SSID และ PASS ตามที่สะดวก
- มีการแสดงผลการทำงานผ่านทางจอ OLED Display
- มีเวปแสดงผลการทำงานของ



4.1 มีการแสดงผลค่าของ ADC ที่ควบคุม LED

4.2 มีปุ่มกด เปิด/ปิด LED ได้ ถ้าเปิดจะสว่างเท่ากับค่า PWM ที่ตั้งค่าไว้



ตัวอย่างหน้า Web (สามารถออกแบบให้แตกต่างได้)

เขียนโปรแกรมลงในกล่องคำตอบด้านล่าง และถ่ายวิดีโอผลลัพธ์ของโจทย์นี้ Upload ไฟล์ตามหมู่เรียน

ในกรณีตัวหน้ากระดาษไม่พอให้เพิ่มหน้าแทรกในไฟล์แทน

```
#include <WiFi.h>
#include <WebServer.h>
#include <SPI.h>
#include <Wire.h>
#include <Adafruit_GFX.h>
#include <Adafruit_SSD1306.h>
#include "EEPROM.h"
#define EEPROM_SIZE 1000
Adafruit_SSD1306 display(128, 32, &Wire);
#define NISIT 38
#define SW 5
#define LED 25
const char* ssid = "ESP32_kit S U S";
const char* password = "12345678";
int bufeeprom;

IPAddress local_ip(192, 168, 1, 1);
IPAddress gateway(192, 168, 1, 1);
IPAddress subnet(255, 255, 255, 0);

WebServer server(80);

uint8_t LED1pin = 2;
bool LED1status = LOW;

uint8_t LED2pin = 4;
bool LED2status = LOW;

long lasttime10ms = 0;
bool stateONOFF = true;
int adcvalue = 0;

void setup() {
  Serial.begin(115200);
  pinMode(LED1pin, OUTPUT);
  pinMode(LED2pin, OUTPUT);
  pinMode(SW, INPUT_PULLUP);
  pinMode(LED, OUTPUT);

  WiFi.softAP(ssid, password);
  WiFi.softAPConfig(local_ip, gateway, subnet);
  Serial.print("IP address: ");
  Serial.println(WiFi.softAPIP());
  delay(100);

  server.on("/", handle_OnConnect);
  server.on("/led1on", handle_led1on);
  server.on("/led1off", handle_led1off);
  server.on("/led2on", handle_led2on);
  server.on("/led2off", handle_led2off);
  server.onNotFound(handle_NotFound);

  server.begin();
  Serial.println("HTTP server started");

  if (!display.begin(SSD1306_SWITCHCAPVCC, 0x3C)) {
    Serial.println(F("SSD1306 allocation failed"));
    for (;;) // Don't proceed, loop forever
  }

  //Test LCD, Clear, Setfontsize, Color, Cursor, print, display
  display.clearDisplay();
  display.setTextSize(1);
  display.setTextColor(SSD1306_WHITE);
  display.setCursor(0, 0);
  display.printf("Lab10.6 WIFI");
  display.display();

  if (!EEPROM.begin(EEPROM_SIZE)) {
    Serial.println("Failed init EEPROM");
    delay(1000);
    ESP.restart();
  }
  bufeeprom = EEPROM.readUChar(NISIT);
  Serial.println("Success init EEPROM");

  ledcSetup(0, 1000, 8);
  ledcAttachPin(LED, 0);

  int buffStar = EEPROM.readUChar(0);
  if(buffStar == 1){
    LED2status = HIGH;
  }else{
    LED2status = LOW;
  }
}
```

เขียนโปรแกรมลงในกล่องคำตอบด้านล่าง และถ่ายวิดีโอผลลัพธ์ของโจทย์นี้ Upload ไฟล์ตามหมู่เรียน

ในกรณีตัวหน้ากระดาษไม่พอให้เพิ่มหน้าแทรกในไฟล์แทน

```
void loop() {
  server.handleClient();
  if (LED1status)
  {
    digitalWrite(LED1pin, HIGH);
  }
  else
  {
    digitalWrite(LED1pin, LOW);
  }

  if (LED2status)
  {
    //digitalWrite(LED2pin, HIGH);
    ledcWrite(0, bufeeprom);
  }
  else
  {
    //digitalWrite(LED2pin, LOW);
    ledcWrite(0, 0);
  }

  if (millis() - lasttime10ms >= 10) {
    lasttime10ms = millis();
    adcvalue = analogRead(34);

    if (LED2status == HIGH) {
      ledcWrite(0, bufeeprom);
      display.clearDisplay();
      display.setTextSize(1);
      display.setTextColor(SSD1306_WHITE);
      display.setCursor(0, 0);
      display.printf("ON  ADC:%04d", adcvalue);
      display.setCursor(0, 8);
      display.printf("Address: %d(0x%x)", NISIT, NISIT);
      display.setCursor(0, 16);
      display.printf("Data: %d(0x%x)", bufeeprom, bufeeprom);
      display.display();
    } else {
      display.clearDisplay();
      display.setTextSize(1);
      display.setTextColor(SSD1306_WHITE);
      display.setCursor(0, 0);
      display.printf("OFF ADC:%04d", adcvalue);
      display.setCursor(0, 8);
      display.printf("Address: %d(0x%x)", NISIT, NISIT);
      display.setCursor(0, 16);
      display.printf("Data: %d(0x%x)", bufeeprom, bufeeprom);
      display.display();
    }
  }
  if (digitalRead(SW) == LOW) {
    delay(100);
    int adconv = (int)(adcvalue * 255 / 4095);
    EEPROM.writeUChar(NISIT, adconv);
    EEPROM.commit();
    delay(200);
    bufeeprom = EEPROM.readUChar(NISIT);
  }
}
```



```
void handle_OnConnect() {
    LED1status = LOW;
    LED2status = LOW;
    Serial.println("GPIO2 Status: OFF | GPIO4 Status: OFF");
    server.send(200, "text/html", SendHTML(LED1status, LED2status));
}

void handle_led1on() {
    LED1status = HIGH;
    Serial.println("GPIO2 Status: ON");
    server.send(200, "text/html", SendHTML(true, LED2status));
}

void handle_led1off() {
    LED1status = LOW;
    Serial.println("GPIO2 Status: OFF");
    server.send(200, "text/html", SendHTML(false, LED2status));
}

void handle_led2on() {
    LED2status = HIGH;
    Serial.println("GPIO4 Status: ON");
    server.send(200, "text/html", SendHTML(LED1status, true));

    EEPROM.writeUChar(0, 1);
    EEPROM.commit();
}

void handle_led2off() {
    LED2status = LOW;
    Serial.println("GPIO4 Status: OFF");
    server.send(200, "text/html", SendHTML(LED1status, false));

    EEPROM.writeUChar(0, 0);
    EEPROM.commit();
}

void handle_NotFound() {
    server.send(404, "text/plain", "Not found");
}

String SendHTML(uint8_t led1stat, uint8_t led2stat) {
    String ptr = "<!DOCTYPE html> <html>\n";
    ptr += "<head><meta name=\"viewport\" content=\"width=device-width, initial-scale=1.0, user-scalable=no\">\n";
    ptr += "<title>LED Control</title>\n";
    ptr += "<style>html { font-family: Helvetica; display: inline-block; margin: 0px auto; text-align: center;}\n";
    ptr += "body{margin-top: 50px;} h1 {color: #444444;margin: 50px auto 30px;} h3 {color: #444444;margin-bottom: 50px;}\n";
    ptr += ".button {display: block;width: 80px;background-color: #3498db;border: none;color: white;padding: 13px 30px;text-decoration: none;font-size: 25px;margin: 0px auto 35px;cursor: pointer;border-radius: 4px;}\n";
    ptr += ".button-on {background-color: #3498db;}\n";
    ptr += ".button-on:active {background-color: #2980b9;}\n";
    ptr += ".button-off {background-color: #34495e;}\n";
    ptr += ".button-off:active {background-color: #2c3e50;}\n";
    ptr += "p {font-size: 14px;color: #888;margin-bottom: 10px;}\n";
    ptr += "</style>\n";
    ptr += "</head>\n";
    ptr += "<body>\n";
    ptr += "<h1>ESP32 Web Server</h1>\n";
    ptr += "<h3>AP Mode</h3>\n";
    ptr += "<h3>IP:192.168.1.1</h3>\n";
    ptr += "<h3>PWM data.\" + (String)bufeeprom +\"</h3>\n";
    ptr += "<h3>LED CONTROL</h3>\n";
    if (led2stat)
    {
        ptr += "<p>LED2 Status: ON</p><a class=\"button button-off\" href=\"/led2off\">OFF</a>\n";
    }
    else
    {
        ptr += "<p>LED2 Status: OFF</p><a class=\"button button-on\" href=\"/led2on\">ON</a>\n";
    }

    ptr += "</body>\n";
    ptr += "</html>\n";
    return ptr;
}
```