National Textile University, Faisalabad

Department of Computer Science

Name:	MUHAMMAD UMAIR
Class:	BSCS 5th-B
Registration No:	23-NTU-CS-1054
Assignment No:	# 1 Task A
Course Name:	IoT Embedded Systems
Submitted To:	Nasir Mehmood
Submission Date:	26 - 10 - 2025

Assignment No 1

Task B

Code

```
// Muhammad Umair    23-NTU-CS-1054
// Task A: Multi-Mode LED Control System with OLED Display and PWM

#include <Arduino.h>
#include <Wire.h>
#include <Adafruit_GFX.h>
#include <Adafruit_SSD1306.h>

//  OLED CONFIGURATION
#define SCREEN_WIDTH 128       // OLED display width (pixels)
#define SCREEN_HEIGHT 64       // OLED display height (pixels)
Adafruit_SSD1306 oled(SCREEN_WIDTH, SCREEN_HEIGHT, &Wire, -1); // OLED object (no reset pin)

//  GPIO PIN ASSIGNMENTS
#define yellowLED 19           // Yellow LED pin
#define greenLED 18            // Green LED pin
#define redLED 17              // Red LED pin
#define MODE_BUTTON 25         // Mode button pin
#define RESET_BUTTON 26        // Reset button pin

//  PWM SETTINGS
#define PWM_YELLOW_CHANNEL 0   // PWM channel for yellow LED
#define PWM_GREEN_CHANNEL 1    // PWM channel for green LED
#define PWM_RED_CHANNEL 2      // PWM channel for red LED
#define PWM_FREQ 5000          // PWM frequency (5 kHz)
#define PWM_RES 10             // PWM resolution (10-bit = 0–1023 range)

//  TIMER VARIABLES
hw_timer_t *blinkTimer = nullptr;  // Timer handler for blink sequence
volatile int blinkStep = 0;        // Tracks which LED is active in blink mode

//  STATE VARIABLES
int currentMode = 0;               // 0: All OFF, 1: Blink, 2: All ON, 3: PWM Fade
bool prevBtnMode = HIGH;           // Stores previous Mode button state
bool prevBtnReset = HIGH;          // Stores previous Reset button state
unsigned long lastDebounceTime = 0;// Tracks time for debounce
const int debounceDelay = 500;     // 500 ms debounce delay
```

```
//  FUNCTION: Display current mode on OLED
void displayMode() {
  oled.clearDisplay();
  oled.setTextSize(2);
  oled.setTextColor(SSD1306_WHITE);
  oled.setCursor(0, 0);
  oled.println(" LED Modes");
  oled.drawLine(0, 18, 127, 18, SSD1306_WHITE);
  oled.setTextSize(1);
  oled.setCursor(10, 30);

  // Show mode title
  switch (currentMode) {
    case 0: oled.print("Mode 1: All OFF"); break;
    case 1: oled.print("Mode 2: Blinking"); break;
    case 2: oled.print("Mode 3: All ON"); break;
    case 3: oled.print("Mode 4: PWM Fade"); break;
  }

  oled.display(); // Refresh OLED
}

//  INTERRUPT SERVICE ROUTINE (Blink Mode)
void IRAM_ATTR onBlinkTimer() {
  if (currentMode != 1) return; // Only run when in blink mode

  blinkStep = (blinkStep + 1) % 3; // Step through 0 → 1 → 2 → 0

  switch (blinkStep) {
    case 0: // Yellow ON
      ledcWrite(PWM_YELLOW_CHANNEL, 255);
      ledcWrite(PWM_GREEN_CHANNEL, 0);
      ledcWrite(PWM_RED_CHANNEL, 0);
      break;

    case 1: // Green ON
      ledcWrite(PWM_YELLOW_CHANNEL, 0);
      ledcWrite(PWM_GREEN_CHANNEL, 255);
      ledcWrite(PWM_RED_CHANNEL, 0);
```

```
      case 1: // Green ON
        ledcWrite(PWM_YELLOW_CHANNEL, 0);
        ledcWrite(PWM_GREEN_CHANNEL, 255);
        ledcWrite(PWM_RED_CHANNEL, 0);
        break;

      case 2: // Red ON
        ledcWrite(PWM_YELLOW_CHANNEL, 0);
        ledcWrite(PWM_GREEN_CHANNEL, 0);
        ledcWrite(PWM_RED_CHANNEL, 255);
        break;
    }
  }

  //  SETUP FUNCTION
  void setup() {
    Serial.begin(115200); // Initialize serial monitor

    // Configure LED pins
    pinMode(yellowLED, OUTPUT);
    pinMode(greenLED, OUTPUT);
    pinMode(redLED, OUTPUT);

    // Configure button pins
    pinMode(MODE_BUTTON, INPUT_PULLUP);
    pinMode(RESET_BUTTON, INPUT_PULLUP);

    // Initialize OLED
    if (!oled.begin(SSD1306_SWITCHCAPVCC, 0x3C)) {
      Serial.println(F("OLED initialization failed!"));
      for (;;) {} // Halt execution if OLED not found
    }
    oled.clearDisplay();
    oled.display();

    // Setup PWM for all LEDs
    ledcSetup(PWM_YELLOW_CHANNEL, PWM_FREQ, PWM_RES);
```

Default (Task A) Auto Live Share Spaces: 2

```
  ledcAttachPin(redLED, PWM_RED_CHANNEL);

  // Setup hardware timer (used for blink mode)
  blinkTimer = timerBegin(0, 80, true);                // 1 tick = 1 µs (80MHz / 80)
  timerAttachInterrupt(blinkTimer, &onBlinkTimer, true);
  timerAlarmWrite(blinkTimer, 500000, true);           // 500 ms interval
  timerAlarmEnable(blinkTimer);

  // Turn off all LEDs initially
  ledcWrite(PWM_YELLOW_CHANNEL, 0);
  ledcWrite(PWM_GREEN_CHANNEL, 0);
  ledcWrite(PWM_RED_CHANNEL, 0);

  displayMode(); // Display initial mode
}

//  MAIN LOOP
void loop() {
  bool btnMode = digitalRead(MODE_BUTTON);
  bool btnReset = digitalRead(RESET_BUTTON);

  // Debounce both buttons
  if (millis() - lastDebounceTime > debounceDelay) {
    //  Mode Button
    if (btnMode == LOW && prevBtnMode == HIGH) {
      currentMode = (currentMode + 1) % 4; // Cycle through modes 0-3
      blinkStep = 0;
      displayMode();
      lastDebounceTime = millis();
    }

    //  Reset Button
    if (btnReset == LOW && prevBtnReset == HIGH) {
      currentMode = 0; // Reset to Mode 0 (All OFF)
      blinkStep = 0;
      displayMode();
      lastDebounceTime = millis();
    }
  }
```

```
prevBtnMode = btnMode;
prevBtnReset = btnReset;

// MODE HANDLING
switch (currentMode) {
  case 0: // MODE 1: All OFF
    ledcWrite(PWM_YELLOW_CHANNEL, 0);
    ledcWrite(PWM_GREEN_CHANNEL, 0);
    ledcWrite(PWM_RED_CHANNEL, 0);
    break;

  case 1: // MODE 2: Blinking (handled by timer ISR)
    break;

  case 2: // MODE 3: All ON
    ledcWrite(PWM_YELLOW_CHANNEL, 255);
    ledcWrite(PWM_GREEN_CHANNEL, 255);
    ledcWrite(PWM_RED_CHANNEL, 255);
    break;

  case 3: // MODE 4: PWM Fading Effect
    for (int dutyCycle = 0; dutyCycle <= 1024 && currentMode == 3; dutyCycle++) {
      // Fade-in
      ledcWrite(PWM_YELLOW_CHANNEL, dutyCycle);
      ledcWrite(PWM_GREEN_CHANNEL, dutyCycle);
      ledcWrite(PWM_RED_CHANNEL, dutyCycle);
      delay(5);
      if (digitalRead(MODE_BUTTON) == LOW || digitalRead(RESET_BUTTON) == LOW) return;
    }

    for (int dutyCycle = 1024; dutyCycle >= 0 && currentMode == 3; dutyCycle--) {
      // Fade-out
      ledcWrite(PWM_YELLOW_CHANNEL, dutyCycle);
      ledcWrite(PWM_GREEN_CHANNEL, dutyCycle);
      ledcWrite(PWM_RED_CHANNEL, dutyCycle);
      delay(5);
      if (digitalRead(MODE_BUTTON) == LOW || digitalRead(RESET_BUTTON) == LOW) return;
    }
    break;
```

Brief Explanation about Code:

This program is built for an **ESP32** that uses three LEDs (yellow, green, and red), two buttons, and an **OLED display**. It allows switching between multiple lighting modes with smooth transitions and clear visual feedback.

When the system starts, the OLED shows the current mode. There are **four modes** in total, controlled by the **Mode** and **Reset** buttons:

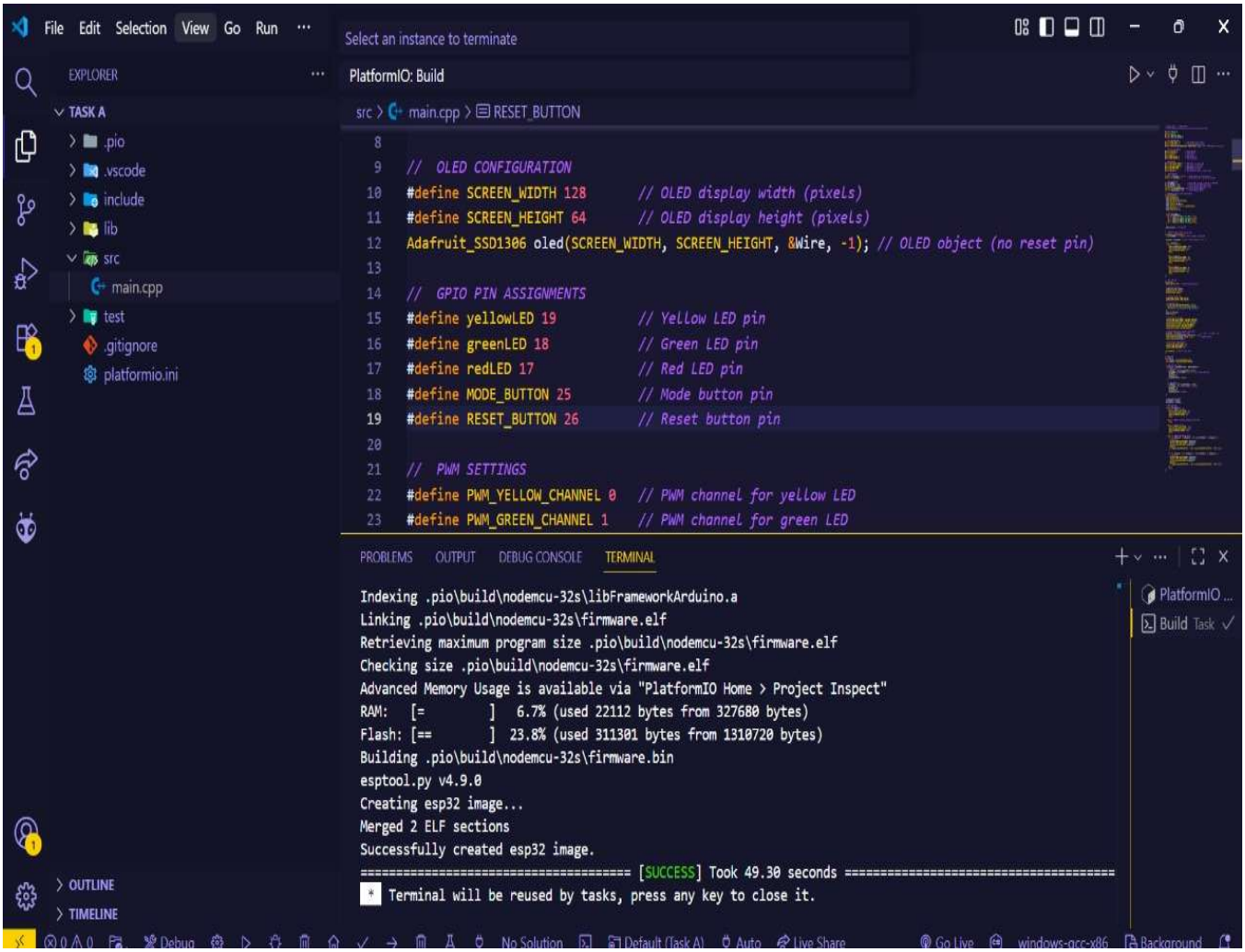
- **Mode 1 – All OFF:** All LEDs remain turned off.

- **Mode 2 – Blinking:** LEDs blink one after another (yellow → green → red) automatically using a timer.
- **Mode 3 – All ON:** All three LEDs light up together.
- **Mode 4 – PWM Fade:** LEDs gradually fade in and out using PWM (Pulse Width Modulation).

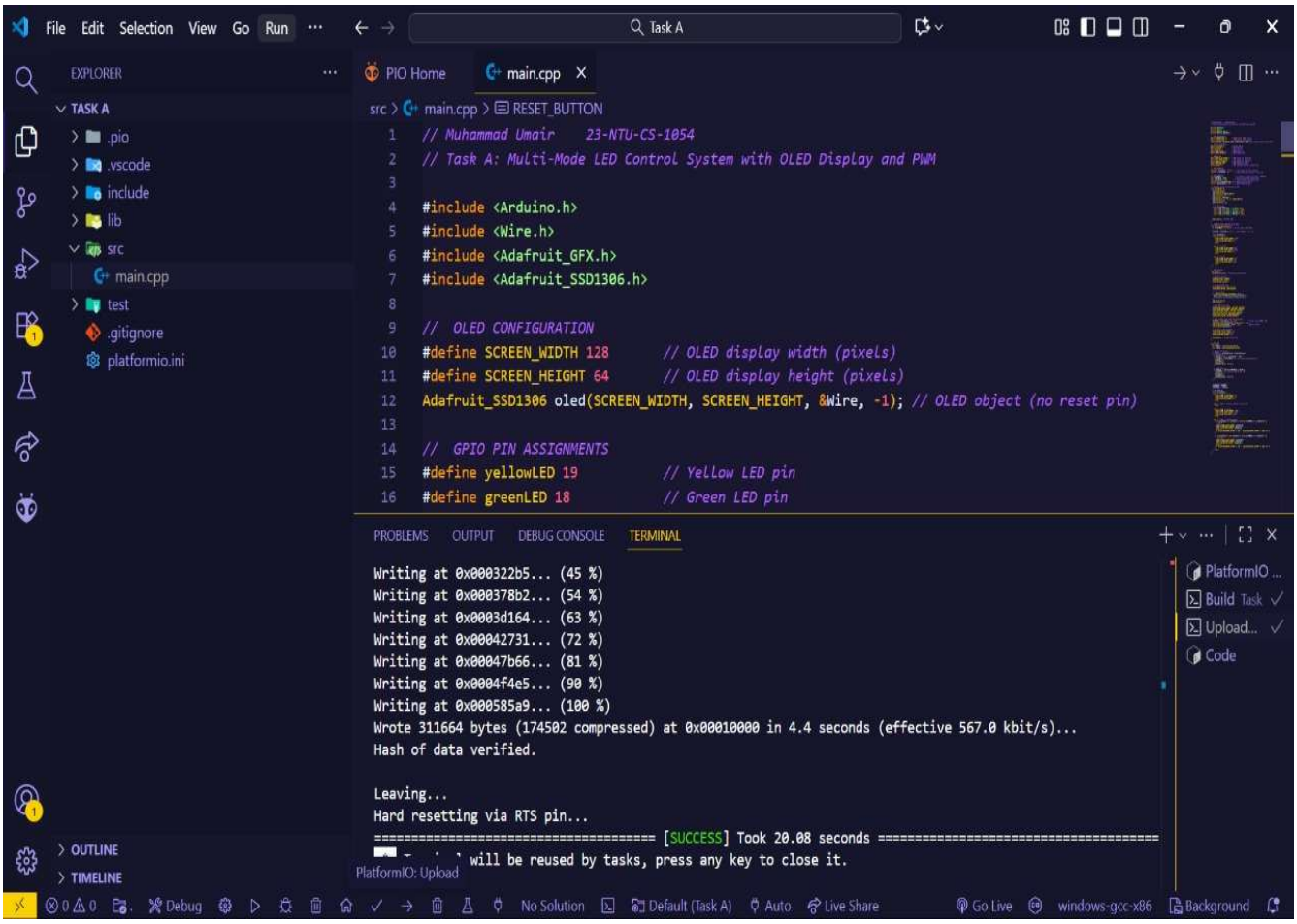
Pressing the **Mode button** cycles through these modes, while the **Reset button** returns the system to "All OFF." The **OLED display** updates each time to show the current mode.

In short, the project demonstrates **multi-mode LED control** with real-time **OLED feedback**, **PWM brightness effects**, and **responsive button handling** using hardware timers and interrupts.

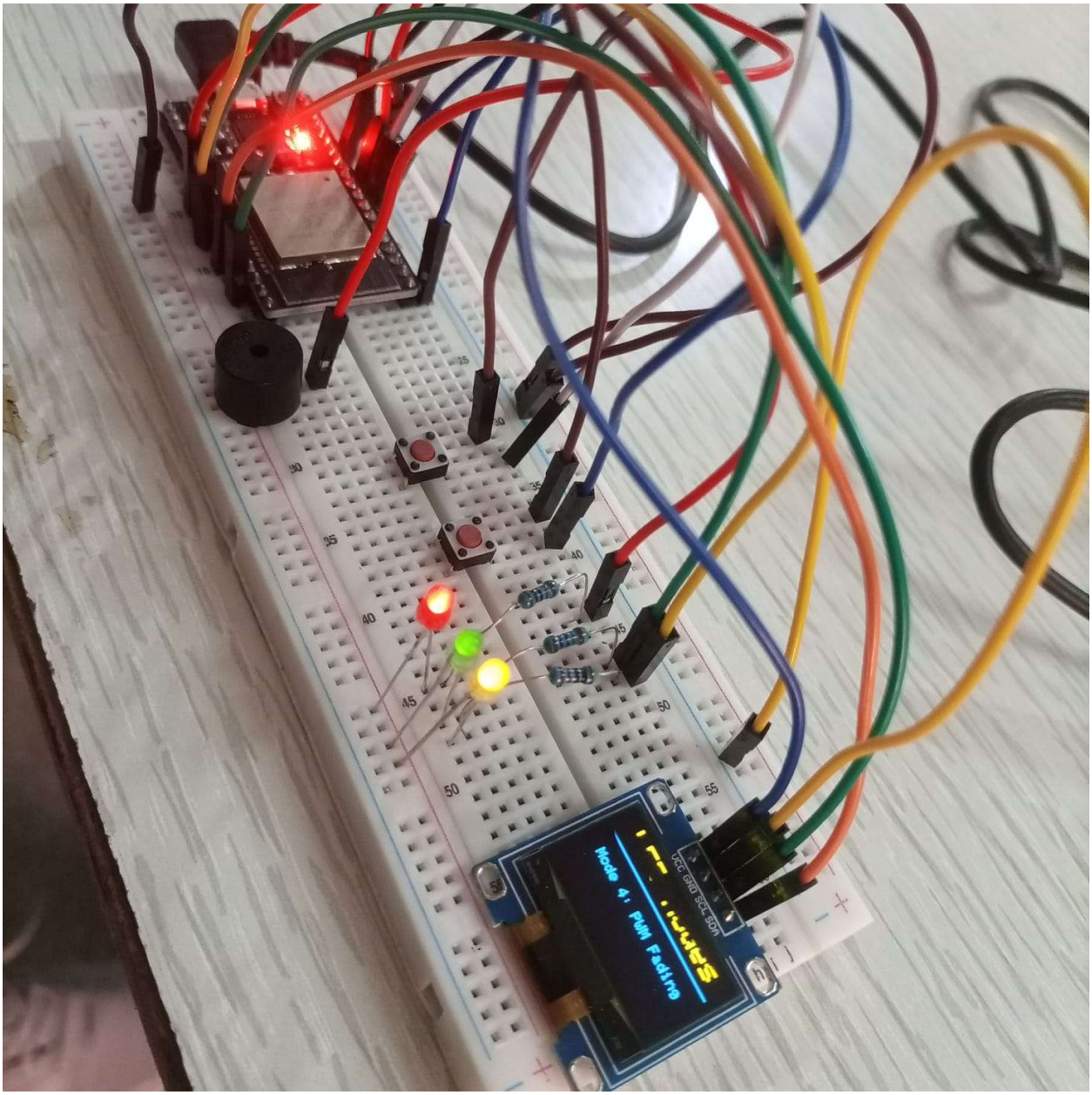
Build Output



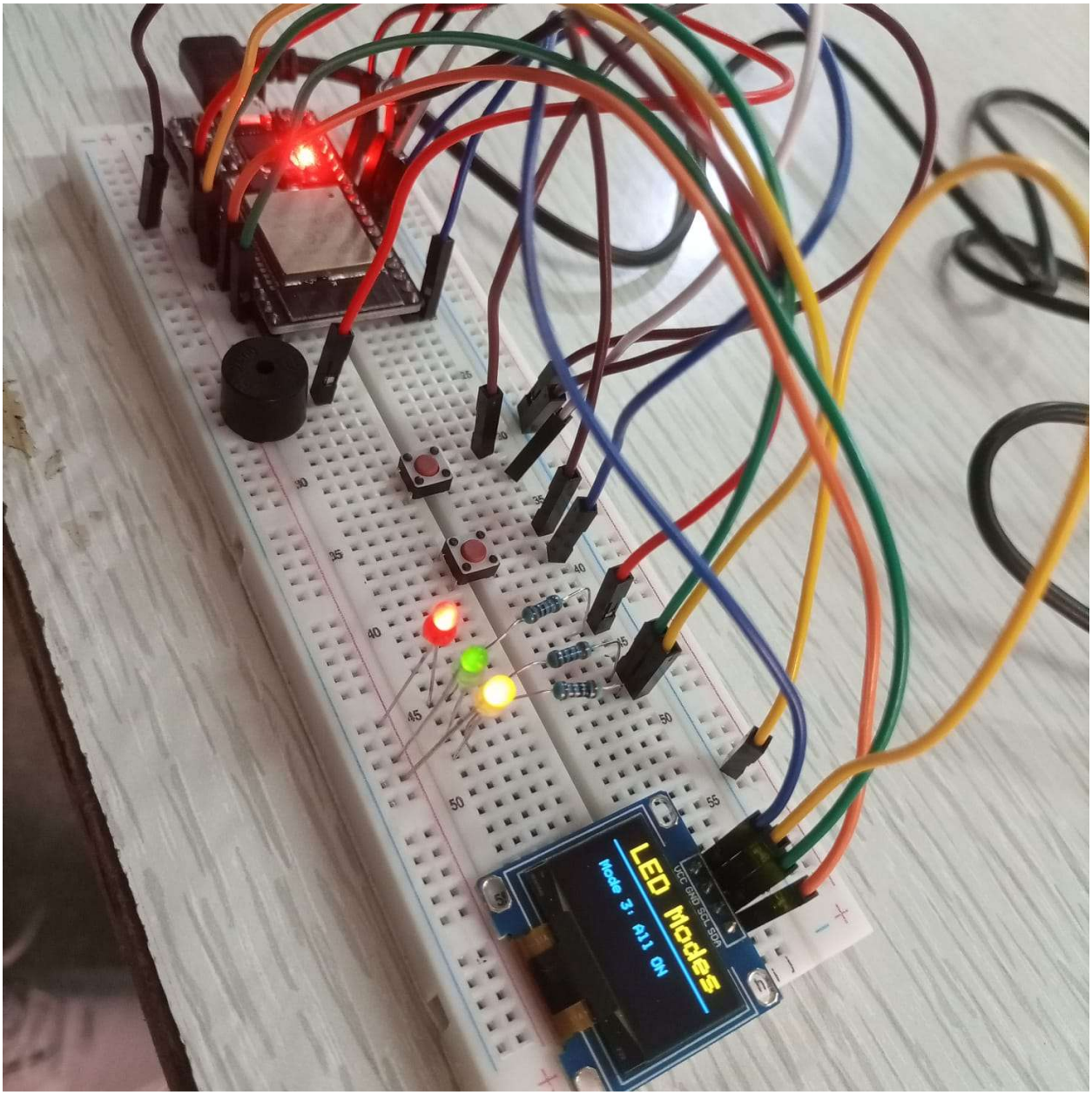
Upload Output



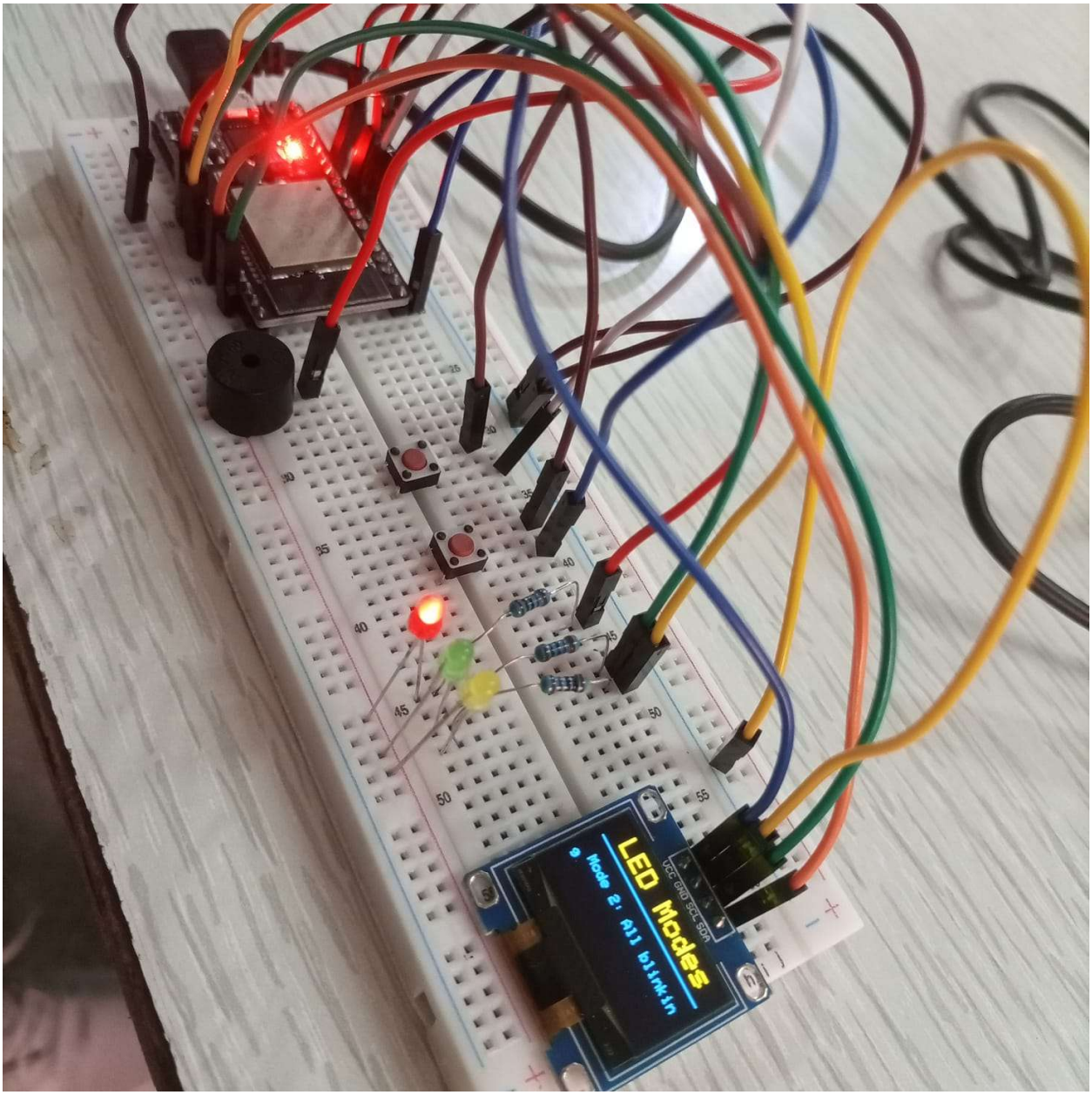
PWM Fading



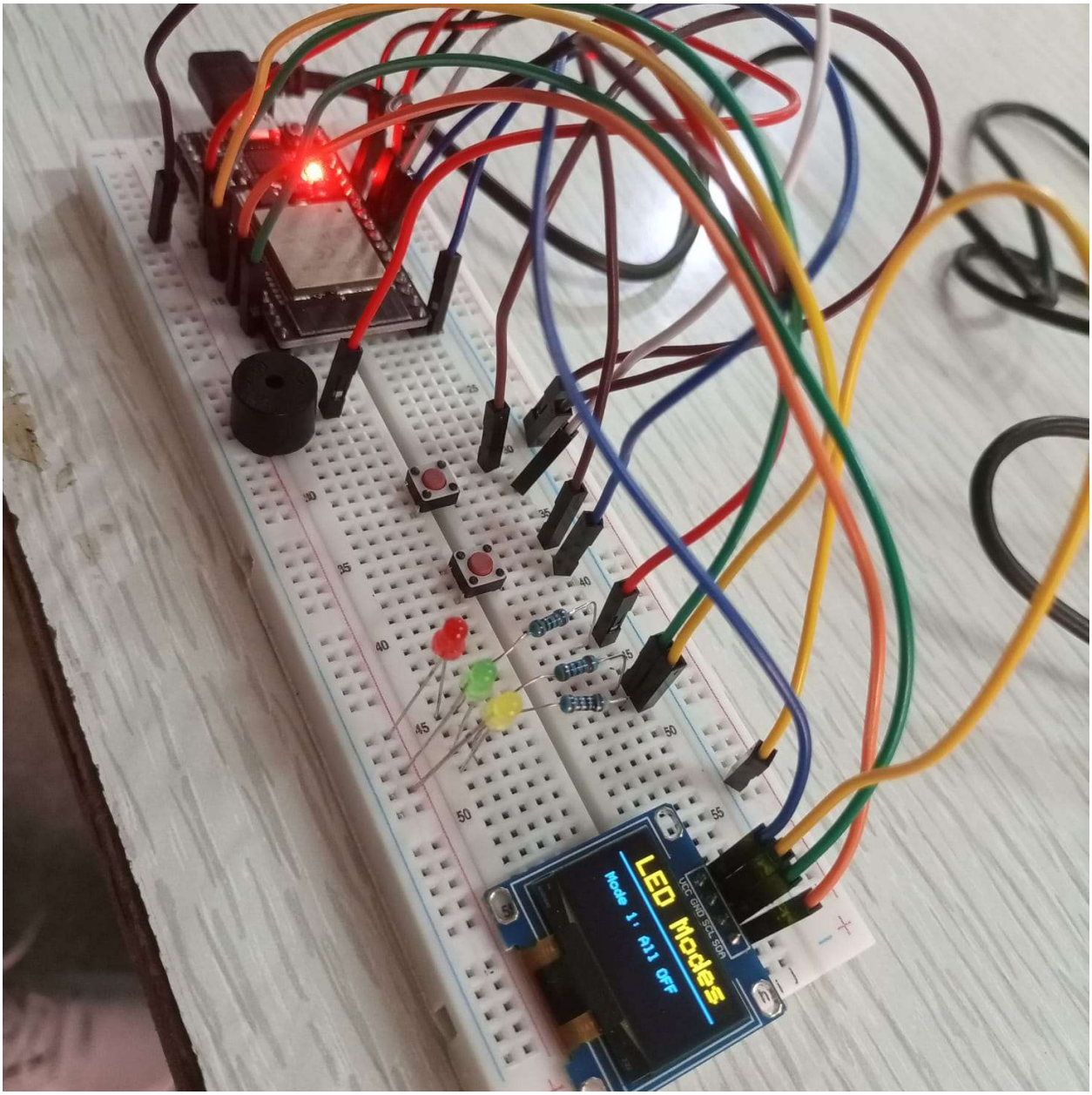
All ON



All Blinking



All Off



Pin Map

Pin No	Name	Function	Use Case
GND .2	Ground	Common Ground	For all LEDs, Buzzer, Buttons, OLED
25	GPIO 25	Pin for Blue Button	Output for Blue Button (Modebtn)
26	GPIO 26	Pin for White Button	Output for White Button (Resetbtn)
27	GPIO 27	Pin for Buzzer	Output for Buzzer
3v3	Power	3.3V output power	OLED VCC
22	GPIO 22	I2C SCL	OLED SCL
21	GPIO 21	I2C SDA	OLED SDA
19	GPIO 19	Pin for Yellow LED	Output for Yellow LED
18	GPIO 18	Pin for Green LED	Output for Green LED
17	GPIO 17	Pin for Red LED	Output for Red LED

Sketch