Department of Computer Science

Name:	Muhammad Hanzala Zahid
Class:	BSCS 5th-B
Registration No:	23-NTU-CS-1067
Assignment#	01
Course Name:	Embedded IoT Systems (CSE-3080)
Submitted To:	Sir Nasir
Submission Date:	25-10-2025

TASK-A

Code Screenshots:

```
// Name: Muhammad Hanzala Zahid
// Roll no: 23-NTU-CS-1067

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

#define SCREEN_WIDTH 128 // OLED display width, in pixels
#define SCREEN_HEIGHT 64 // OLED display height, in pixels
Adafruit_SSD1306 oled(SCREEN_WIDTH, SCREEN_HEIGHT, &Wire, -1); // No reset pin
#define yellowLED   19 // GPIO19 for Yellow LED
#define greenLED  18 // GPIO18 for Green LED
#define redLED   17 // GPIO17 for Red LED
#define MODE_BUTTON   25 // GPIO25 for Mode Button
#define RESET_BUTTON  26 // GPIO26 for Reset Button
#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 // 5 kHz PWM frequency
#define PWM_RES     10 // 10-bit resolution
hw_timer_t *blinkTimer = nullptr; // Timer for Blink mode
volatile int blinkStep = 0; // Step in Blink sequence
int currentMode = 0; // 0: All OFF, 1: Alternate Blink, 2: All ON, 3: PWM
Fading
bool prevBtnMode = HIGH; // Previous state of Mode button
bool prevBtnReset = HIGH; // Previous state of Reset button
unsigned long lastDebounceTime = 0; // For button debounce
const int debounceDelay = 500; // 500ms debounce delay

void displayMode() { // Function: Display current mode on OLED
  oled.clearDisplay(); // Clear display
  oled.setTextSize(2); // Set text size
  oled.setTextColor(SSD1306_WHITE); // Set text color to white
  oled.setCursor(0, 0); // Set cursor to top-left
  oled.println(" LED Modes"); // Title
  oled.drawLine(0, 18, 127, 18, SSD1306_WHITE); // Draw line under title
  oled.setTextSize(1); // Set smaller text size
  oled.setCursor(10, 30); // Set cursor for mode display
  switch (currentMode) { // Display mode description
    case 0: // All LEDs OFF
      oled.print("Mode 1: All OFF"); // Display mode 1
```

```
      break;
    case 1: // Alternate Blink LEDs
      oled.print("Mode 2: All blinking"); // Display mode 2
      break;
    case 2: // All LEDs ON
      oled.print("Mode 3: All ON"); // Display mode 3
      break;
    case 3: // PWM Fading LEDs
      oled.print("Mode 4: PWM Fading"); // Display mode 4
      break;
  }
  oled.display(); // Update display
}

void IRAM_ATTR onBlinkTimer() { // Timer ISR for Blink mode
  if (currentMode == 2) return;  // Only run in Sequence mode
  blinkStep = (blinkStep + 1) % 3;  // 0→1→2→0 repeat
  switch (blinkStep) { // Set LED states based on blink step
    case 0: // Yellow ON
      ledcWrite(PWM_YELLOW_CHANNEL, 255); // Yellow ON
      ledcWrite(PWM_GREEN_CHANNEL, 0); // Green OFF
      ledcWrite(PWM_RED_CHANNEL, 0); // Red OFF
      break;
    case 1: // Green ON
      ledcWrite(PWM_YELLOW_CHANNEL, 0); // Yellow OFF
      ledcWrite(PWM_GREEN_CHANNEL, 255); // Green ON
      ledcWrite(PWM_RED_CHANNEL, 0); // Red OFF
      break;
    case 2: // Red ON
      ledcWrite(PWM_YELLOW_CHANNEL, 0); // Yellow OFF
      ledcWrite(PWM_GREEN_CHANNEL, 0); // Green OFF
      ledcWrite(PWM_RED_CHANNEL, 255); // Red ON
      break;
  }
}

void setup() {
  Serial.begin(115200); // Initialize Serial
  pinMode(yellowLED, OUTPUT); // Set Yellow LED pin as OUTPUT
  pinMode(greenLED, OUTPUT); // Set Green LED pin as OUTPUT
  pinMode(redLED, OUTPUT); // Set Red LED pin as OUTPUT
  pinMode(MODE_BUTTON, INPUT_PULLUP); // Set Mode Button pin as INPUT_PULLUP
  pinMode(RESET_BUTTON, INPUT_PULLUP); // Set Reset Button pin as INPUT_PULLUP
  if (!oled.begin(SSD1306_SWITCHCAPVCC, 0x3C)) {
    Serial.println(F("OLED initialization failed!"));
    for (;;) {}
  }
  oled.clearDisplay();
```

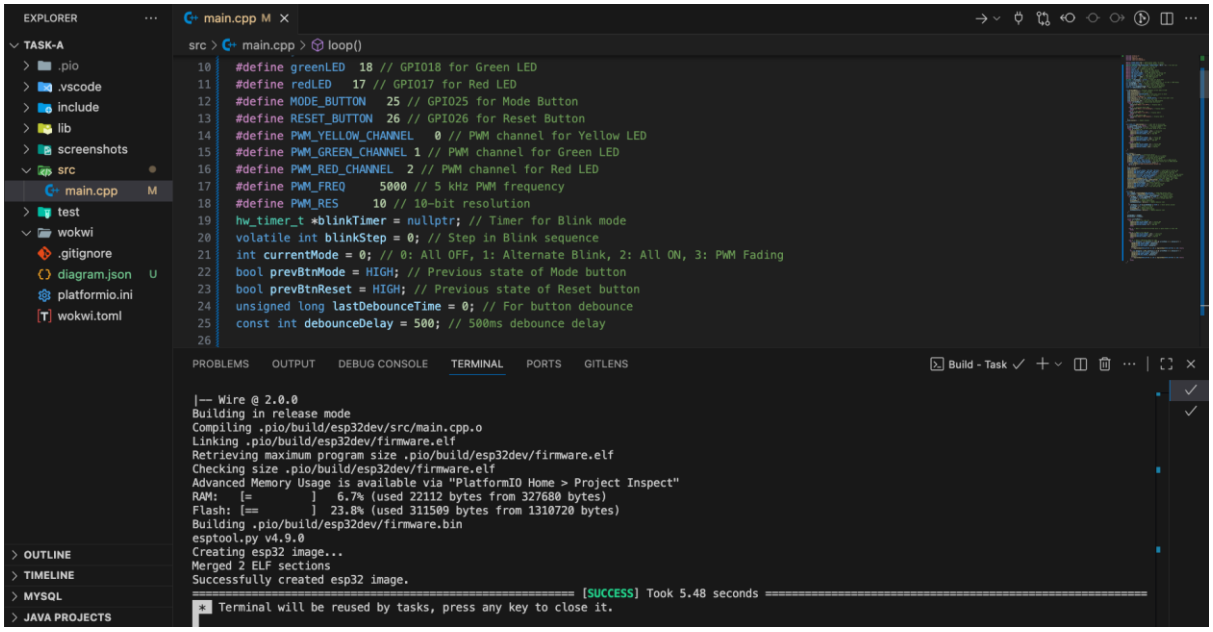
```
  oled.display();
  ledcSetup(PWM_YELLOW_CHANNEL, PWM_FREQ, PWM_RES); // Setup PWM for Yellow
LED
  ledcSetup(PWM_GREEN_CHANNEL, PWM_FREQ, PWM_RES); // Setup PWM for Green LED
  ledcSetup(PWM_RED_CHANNEL, PWM_FREQ, PWM_RES); // Setup PWM for Red LED
  ledcAttachPin(yellowLED, PWM_YELLOW_CHANNEL); // Attach Yellow LED pin to
PWM channel
  ledcAttachPin(greenLED, PWM_GREEN_CHANNEL); // Attach Green LED pin to PWM
channel
  ledcAttachPin(redLED, PWM_RED_CHANNEL); // Attach Red LED pin to PWM channel
  blinkTimer = timerBegin(0, 80, true); // Initialize timer (80MHz / 80 =
1MHz)
  timerAttachInterrupt(blinkTimer, &onBlinkTimer, true); // Attach ISR
  timerAlarmWrite(blinkTimer, 500000, true); // Set alarm to 500ms
  timerAlarmEnable(blinkTimer); // Enable the alarm
  ledcWrite(PWM_YELLOW_CHANNEL, 0); // Turn OFF Yellow LED
  ledcWrite(PWM_GREEN_CHANNEL, 0); // Turn OFF Green LED
  ledcWrite(PWM_RED_CHANNEL, 0); // Turn OFF Red LED
  displayMode();
}

void loop() {
  bool btnMode = digitalRead(MODE_BUTTON); // Read Mode button state
  bool btnReset = digitalRead(RESET_BUTTON); // Read Reset button state
  if (millis() - lastDebounceTime > debounceDelay) { // Debounce check
    if (btnMode == LOW && prevBtnMode == HIGH) { // Mode button pressed
      currentMode = (currentMode + 1) % 4; // Cycle through modes 0-3
      blinkStep = 0; // Reset blink step
      displayMode(); // Update OLED display
      lastDebounceTime = millis(); // Update debounce timer
    }
    if (btnReset == LOW && prevBtnReset == HIGH) { // Reset button pressed
      currentMode = 0; // Reset to Mode 0
      blinkStep = 0; // Reset blink step
      displayMode(); // Update OLED display
      lastDebounceTime = millis(); // Update debounce timer
    }
  }

  prevBtnMode = btnMode;
  prevBtnReset = btnReset;

  switch (currentMode) {
    case 0: // Mode 1: All LEDs OFF
      ledcWrite(PWM_YELLOW_CHANNEL, 0); // Yellow OFF
      ledcWrite(PWM_GREEN_CHANNEL, 0); // Green OFF
      ledcWrite(PWM_RED_CHANNEL, 0); // Red OFF
      break;
```

```
    case 1: // Mode 2: Alternate Blink Mode which is gonna handle in Timer ISR
      break;

    case 2: // Mode 3: All LEDs ON
      ledcWrite(PWM_YELLOW_CHANNEL, 255); // Yellow ON
      ledcWrite(PWM_GREEN_CHANNEL, 255); // Green ON
      ledcWrite(PWM_RED_CHANNEL, 255); // Red ON
      break;

    case 3: // Mode 4: PWM Fading Mode for LEDs
      for (int dutyCycle = 0; dutyCycle <= 1024 && currentMode == 3;
dutyCycle++) {
        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--) {
        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;
  }
}
```

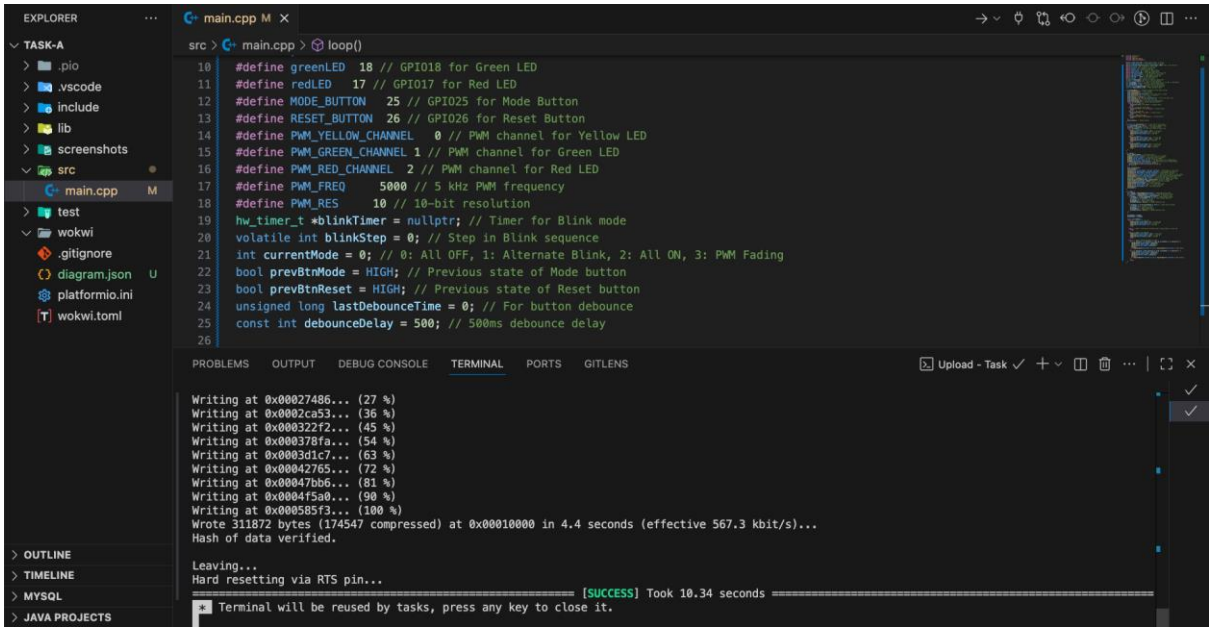
Short explanation about code:

This project controls three LEDs (yellow, green, and red) and shows the current mode on an OLED display. It has four modes i.e all LEDs off, blinking one by one, all LEDs on, and smooth fading using PWM. You can switch between these modes using the Mode button, and the Reset button brings everything back to the “All Off” mode. The blinking pattern is handled automatically by a timer, while the display updates to show the active mode

Build Output:

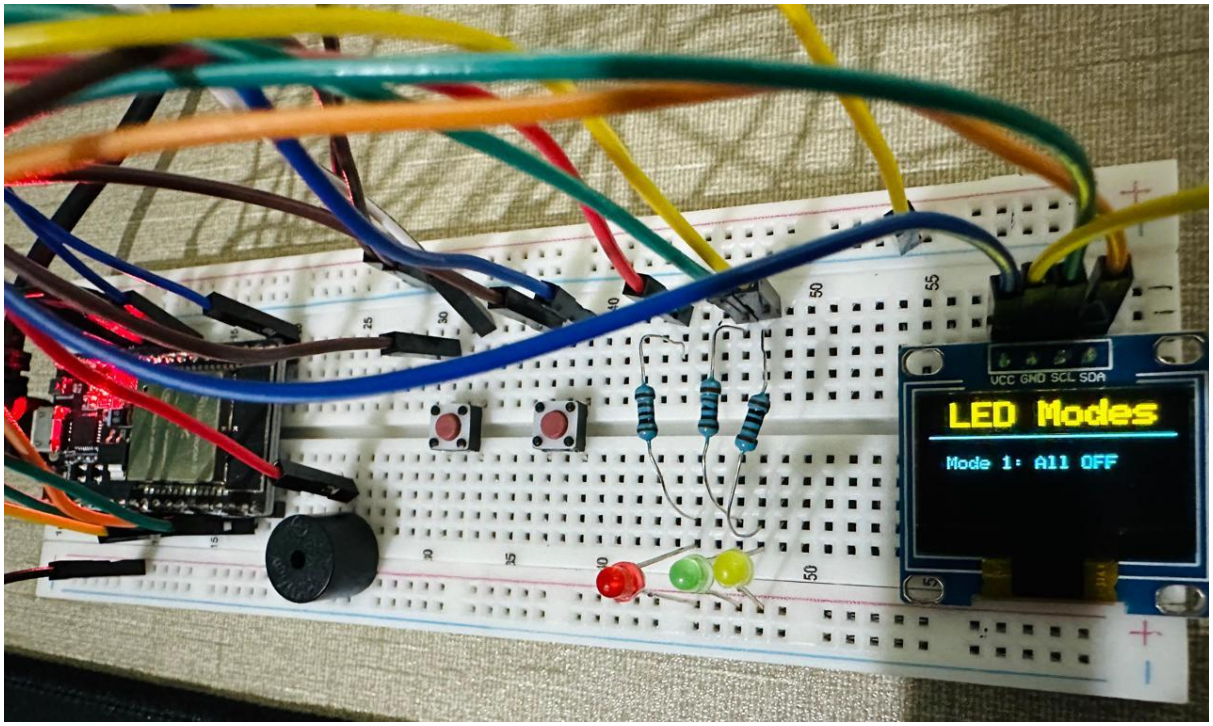


Upload Output:

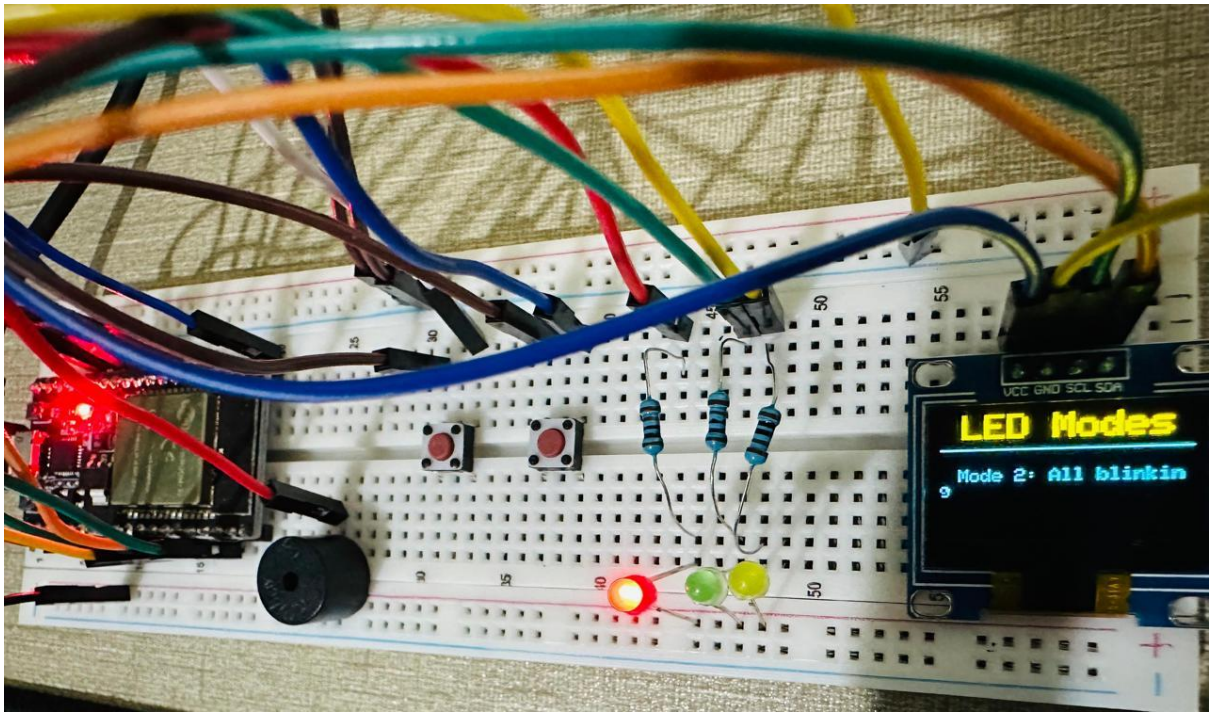


Hardware working:

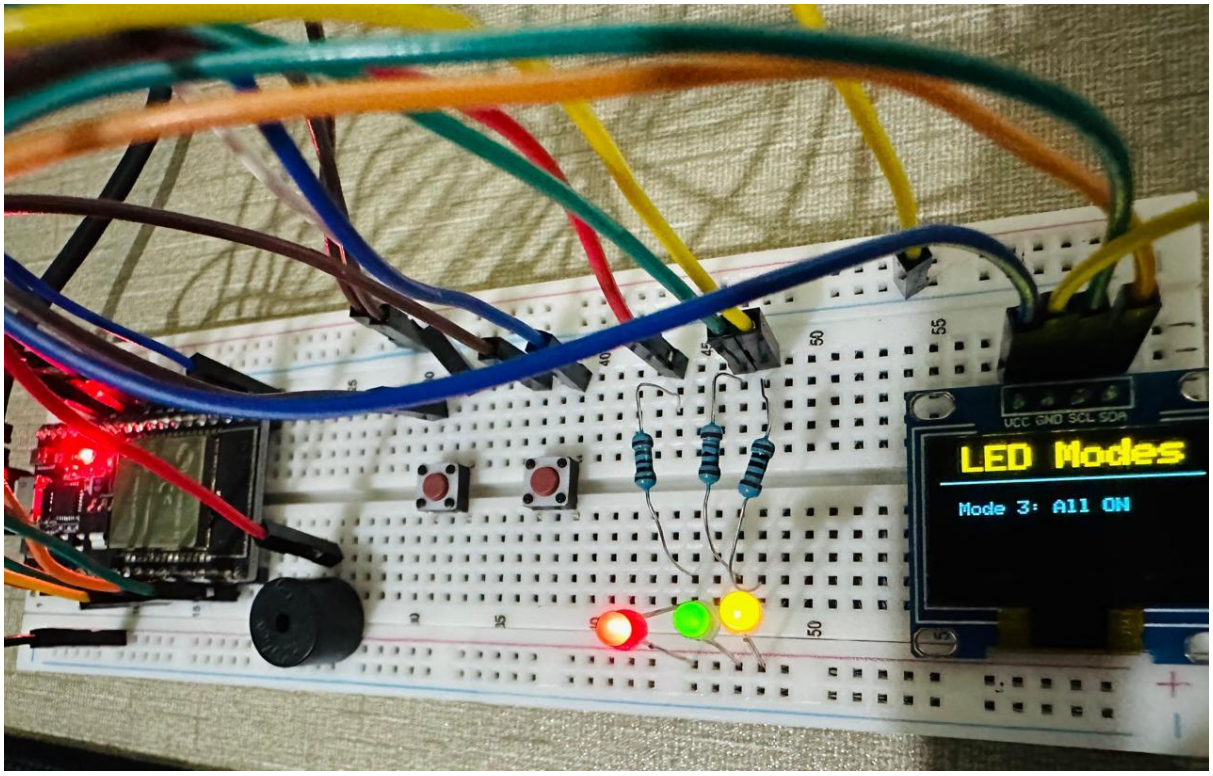
1: All LEDS OFF:



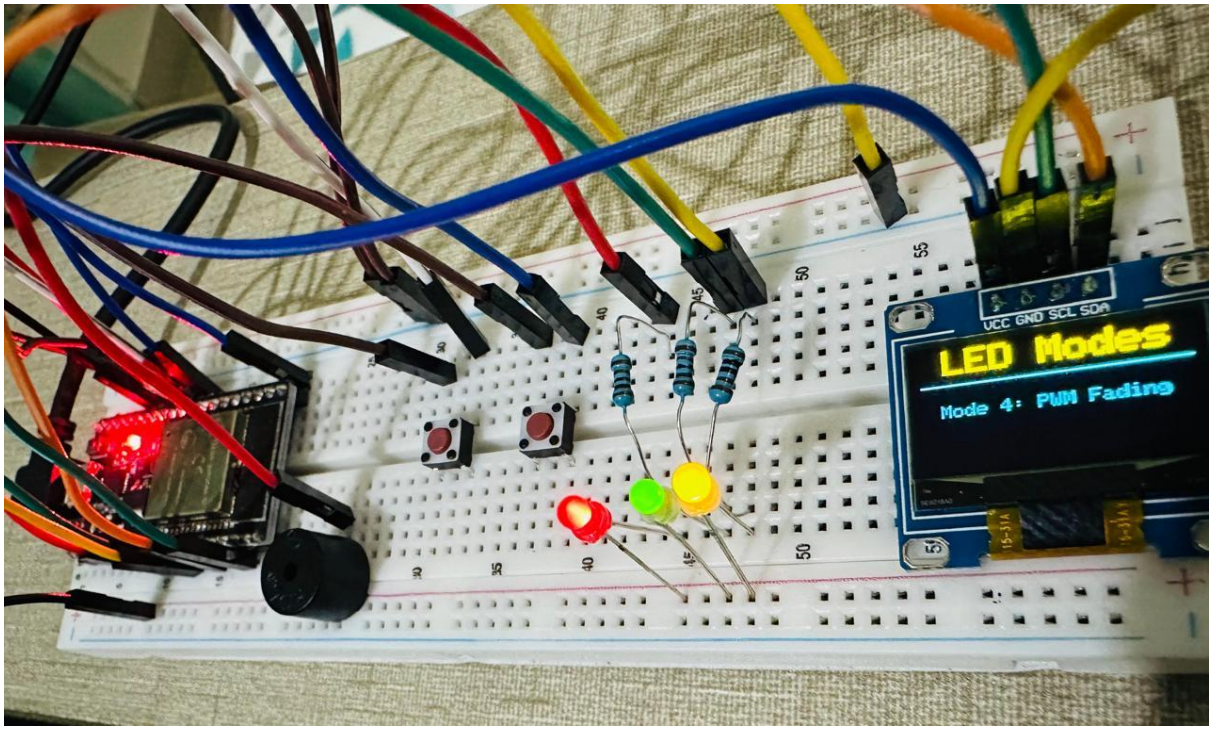
2: All LEDS blinking:



3: All LEDS on:



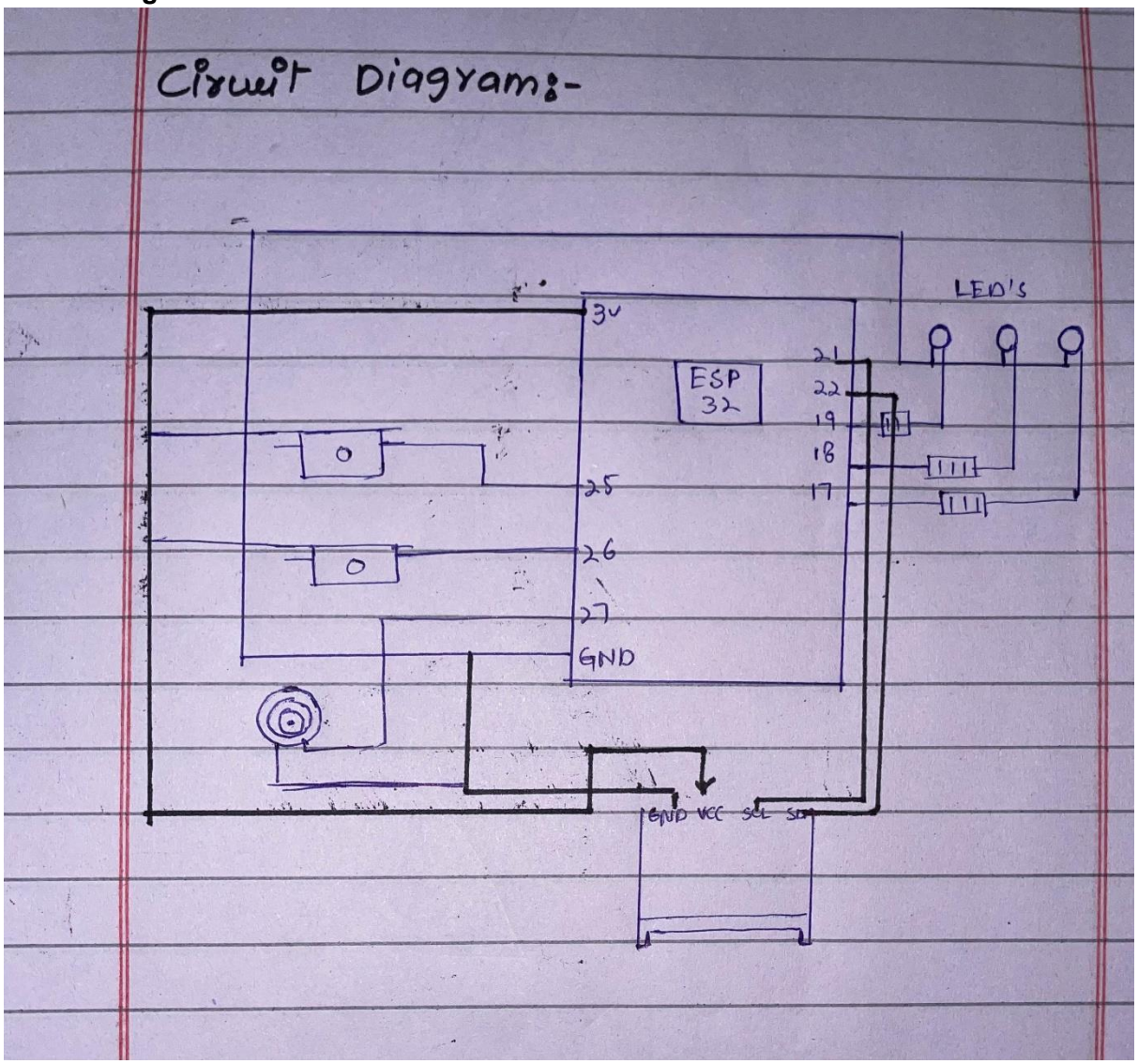
4: PWM working (high to low):



Wokwi Project link:

https://wokwi.com/projects/445771605982073857

Circuit Diagram:



Pin Maping Chart:

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