

National Textile University, Faisalabad



Name:	Muhammad Zohaib Warrach
Class:	BS-CS(B)
Registration No:	23-NTU-CS-1080
Assignment 1:	Task A
Course Name:	IoT & Embedded Systems
Submitted To:	Dr. Nasir Mehmood

Document of Assignment 1

Task A

Code Screenshot:



```
1 #include <Arduino.h>
2 #include <Wire.h>
3 #include <Adafruit_GFX.h>
4 #include <Adafruit_SSD1306.h>
5
6 #define SCREEN_WIDTH 128 // OLED display width, in pixels
7 #define SCREEN_HEIGHT 64 // OLED display height, in pixels
8 Adafruit_SSD1306 oled(SCREEN_WIDTH, SCREEN_HEIGHT, &Wire, -1); // No reset pin
9 #define yellowLED 19 // GPIO19 for Yellow LED
10 #define greenLED 18 // GPIO18 for Green LED
11 #define redLED 17 // GPIO17 for Red LED
12 #define MODE_BUTTON 25 // GPIO25 for Mode Button
13 #define RESET_BUTTON 26 // GPIO26 for Reset Button
14 #define PWM_YELLOW_CHANNEL 0 // PWM channel for Yellow LED
15 #define PWM_GREEN_CHANNEL 1 // PWM channel for Green LED
16 #define PWM_RED_CHANNEL 2 // PWM channel for Red LED
17 #define PWM_FREQ 5000 // 5 kHz PWM frequency
18 #define PWM_RES 10 // 10-bit resolution
19 hw_timer_t *blinkTimer = nullptr; // Timer for Blink mode
20 volatile int blinkStep = 0; // Step in Blink sequence
21 int currentMode = 0; // 0: All OFF, 1: Alternate Blink, 2: All ON, 3: PWM Fading
22 bool prevBtnMode = HIGH; // Previous state of Mode button
23 bool prevBtnReset = HIGH; // Previous state of Reset button
24 unsigned long lastDebounceTime = 0; // For button debounce
25 const int debounceDelay = 500; // 500ms debounce delay
26
27 void displayMode() { // Function: Display current mode on OLED
28     oled.clearDisplay(); // Clear display
29     oled.setTextSize(2); // Set text size
30     oled.setTextColor(SSD1306_WHITE); // Set text color to white
31     oled.setCursor(0, 0); // Set cursor to top-left
32     oled.println(" LED Modes"); // Title
33     oled.drawLine(0, 18, 127, 18, SSD1306_WHITE); // Draw line under title
34     oled.setTextSize(1); // Set smaller text size
35     oled.setCursor(10, 30); // Set cursor for mode display
36     switch (currentMode) { // Display mode description
37         case 0: // All LEDs OFF
38             oled.print("Mode 1: All OFF"); // Display mode 1
39             break;
40         case 1: // Alternate Blink LEDs
41             oled.print("Mode 2: All blinking"); // Display mode 2
42             break;
43         case 2: // All LEDs ON
44             oled.print("Mode 3: All ON"); // Display mode 3
45             break;
46         case 3: // PWM Fading LEDs
47             oled.print("Mode 4: PWM Fading"); // Display mode 4
48             break;
49     }
50     oled.display(); // Update display
51 }
52
53 void IRAM_ATTR onBlinkTimer() { // Timer ISR for Blink mode
54     if (currentMode == 2) return; // Only run in Sequence mode
55     blinkStep = (blinkStep + 1) % 3; // 0-1-2-0 repeat
56     switch (blinkStep) { // Set LED states based on blink step
57         case 0: // Yellow ON
58             ledcWrite(PWM_YELLOW_CHANNEL, 255); // Yellow ON
59             ledcWrite(PWM_GREEN_CHANNEL, 0); // Green OFF
60             ledcWrite(PWM_RED_CHANNEL, 0); // Red OFF
61     }
62 }
```

```

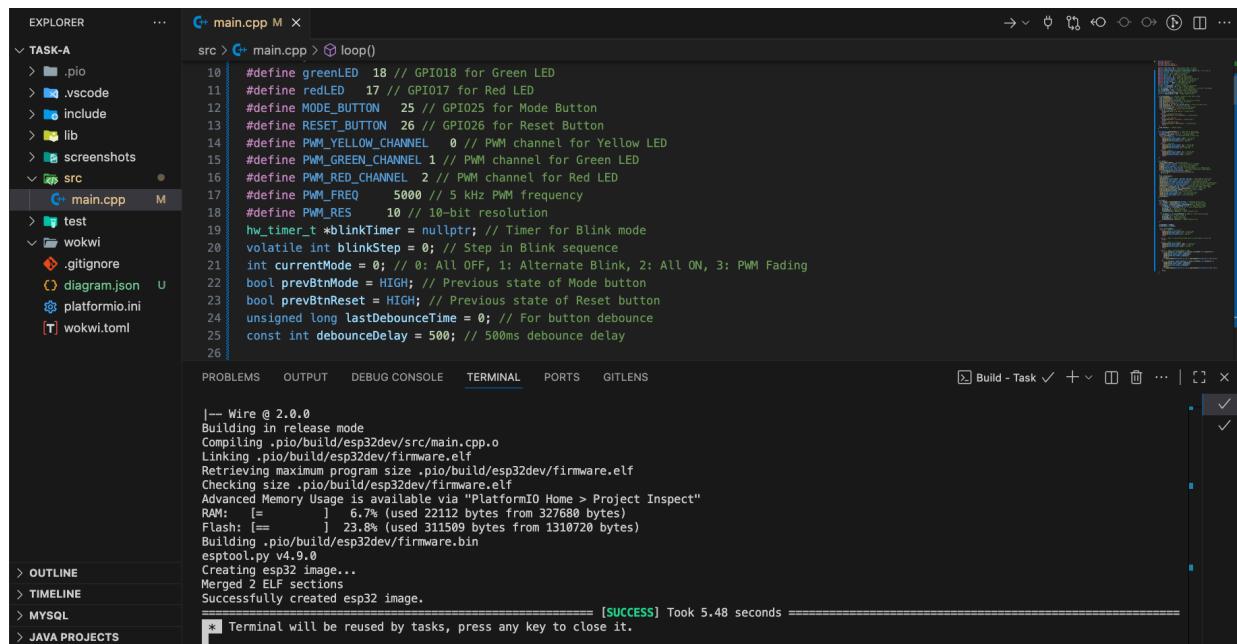
59         ledcWrite(PWM_GREEN_CHANNEL, 0); // Green OFF
60         ledcWrite(PWM_RED_CHANNEL, 0); // Red OFF
61         break;
62     case 1: // Green ON
63         ledcWrite(PWM_YELLOW_CHANNEL, 0); // Yellow OFF
64         ledcWrite(PWM_GREEN_CHANNEL, 255); // Green ON
65         ledcWrite(PWM_RED_CHANNEL, 0); // Red OFF
66         break;
67     case 2: // Red ON
68         ledcWrite(PWM_YELLOW_CHANNEL, 0); // Yellow OFF
69         ledcWrite(PWM_GREEN_CHANNEL, 0); // Green OFF
70         ledcWrite(PWM_RED_CHANNEL, 255); // Red ON
71         break;
72     }
73 }
74
75 void setup() {
76     Serial.begin(115200); // Initialize Serial
77     pinMode(yellowLED, OUTPUT); // Set Yellow LED pin as OUTPUT
78     pinMode(greenLED, OUTPUT); // Set Green LED pin as OUTPUT
79     pinMode(redLED, OUTPUT); // Set Red LED pin as OUTPUT
80     pinMode(MODE_BUTTON, INPUT_PULLUP); // Set Mode Button pin as INPUT_PULLUP
81     pinMode(RESET_BUTTON, INPUT_PULLUP); // Set Reset Button pin as INPUT_PULLUP
82     if (!oled.begin(SSD1306_SWITCHCAPVCC, 0x3C)) {
83         Serial.println(F("OLED initialization failed!"));
84         for (;;) {}
85     }
86     oled.clearDisplay();
87     oled.display();
88     ledcSetup(PWM_YELLOW_CHANNEL, PWM_FREQ, PWM_RES); // Setup PWM for Yellow LED
89     ledcSetup(PWM_GREEN_CHANNEL, PWM_FREQ, PWM_RES); // Setup PWM for Green LED
90     ledcSetup(PWM_RED_CHANNEL, PWM_FREQ, PWM_RES); // Setup PWM for Red LED
91     ledcAttachPin(yellowLED, PWM_YELLOW_CHANNEL); // Attach Yellow LED pin to PWM channel
92     ledcAttachPin(greenLED, PWM_GREEN_CHANNEL); // Attach Green LED pin to PWM channel
93     ledcAttachPin(redLED, PWM_RED_CHANNEL); // Attach Red LED pin to PWM channel
94     blinkTimer = timerBegin(0, 80, true); // Initialize timer (80MHz / 80 = 1MHz)
95     timerAttachInterrupt(blinkTimer, &onBlinkTimer, true); // Attach ISR
96     timerAlarmWrite(blinkTimer, 500000, true); // Set alarm to 500ms
97     timerAlarmEnable(blinkTimer); // Enable the alarm
98     ledcWrite(PWM_YELLOW_CHANNEL, 0); // Turn OFF Yellow LED
99     ledcWrite(PWM_GREEN_CHANNEL, 0); // Turn OFF Green LED
100    ledcWrite(PWM_RED_CHANNEL, 0); // Turn OFF Red LED
101    displayMode();
102 }
103
104 void loop() {
105     bool btnMode = digitalRead(MODE_BUTTON); // Read Mode button state
106     bool btnReset = digitalRead(RESET_BUTTON); // Read Reset button state
107     if (millis() - lastDebounceTime > debounceDelay) { // Debounce check
108         if (btnMode == LOW && prevBtnMode == HIGH) { // Mode button pressed
109             currentMode = (currentMode + 1) % 4; // Cycle through modes 0-3
110             blinkStep = 0; // Reset blink step
111             displayMode(); // Update OLED display
112             lastDebounceTime = millis(); // Update debounce timer
113         }
114         if (btnReset == LOW && prevBtnReset == HIGH) { // Reset button pressed
115             currentMode = 0; // Reset to Mode 0
116             blinkStep = 0; // Reset blink step
117             displayMode(); // Update OLED display
118             lastDebounceTime = millis(); // Update debounce timer
119         }
120     }
121     prevBtnMode = btnMode;
122     prevBtnReset = btnReset;
123
124

```

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:

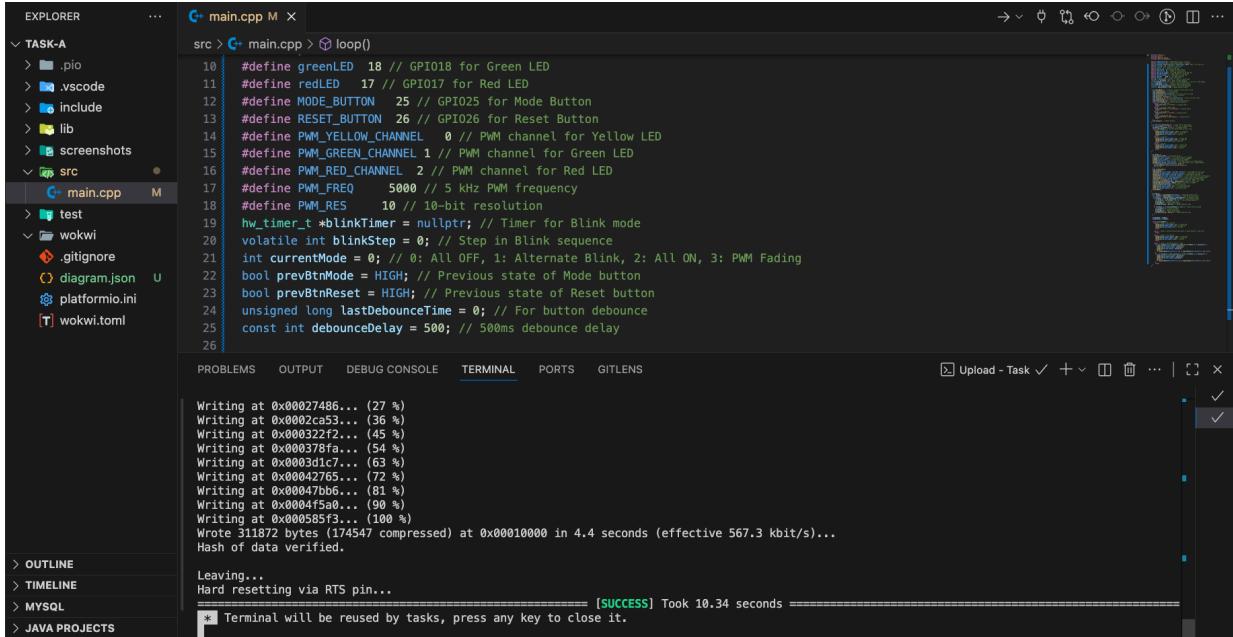


```
src > C:\main.cpp > loop()
10: #define greenLED 18 // GPIO18 for Green LED
11: #define redLED 17 // GPIO17 for Red LED
12: #define MODE_BUTTON 25 // GPIO25 for Mode Button
13: #define RESET_BUTTON 26 // GPIO26 for Reset Button
14: #define PWM_YELLOW_CHANNEL 0 // PWM channel for Yellow LED
15: #define PWM_GREEN_CHANNEL 1 // PWM channel for Green LED
16: #define PWM_RED_CHANNEL 2 // PWM channel for Red LED
17: #define PWM_FREQ 5000 // 5 kHz PWM frequency
18: #define PWM_RES 10 // 10-bit resolution
19: hw_timer_t *blinkKTimer = nullptr; // Timer for Blink mode
20: volatile int blinkStep = 0; // Step in Blink sequence
21: int currentMode = 0; // 0: All OFF, 1: Alternate Blink, 2: All ON, 3: PWM Fading
22: bool prevBtnMode = HIGH; // Previous state of Mode button
23: bool prevBtnReset = HIGH; // Previous state of Reset button
24: unsigned long lastDebounceTime = 0; // For button debounce
25: const int debounceDelay = 500; // 500ms debounce delay
26

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS GITLENS

|--- Wire @ 2.0.0
Building in release mode
Compiling .pio/build/esp32dev/src/main.cpp.o
Linking .pio/build/esp32dev/firmware.elf
Retrieving maximum program size .pio/build/esp32dev/firmware.elf
Checking size .pio/build/esp32dev/firmware.elf
Advanced Memory Usage is available via "PlatformIO Home > Project Inspect"
RAM: [=====] 6.7% (used 22112 bytes from 327680 bytes)
Flash: [=====] 23.8% (used 311509 bytes from 1310720 bytes)
Building .pio/build/esp32dev/firmware.bin
esptool.py v4.9.0
Creating esp32 image...
Merged 2 ELF sections
Successfully created esp32 image.
* Terminal will be reused by tasks, press any key to close it. [SUCCESS] Took 5.48 seconds =====
```

Upload Output:



```

EXPLORER    ...
TASK-A
> .pio
> .vscode
> include
> lib
> screenshots
src
  main.cpp M
> test
wokwi
  .gitignore
  diagram.json U
  platformio.ini
  wokwi.toml

src > main.cpp > loop()

10  #define greenLED 18 // GPIO18 for Green LED
11  #define redLED 17 // GPIO17 for Red LED
12  #define MODE_BUTTON 25 // GPIO25 for Mode Button
13  #define RESET_BUTTON 26 // GPIO26 for Reset Button
14  #define PWM_YELLOW_CHANNEL 0 // PWM channel for Yellow LED
15  #define PWM_GREEN_CHANNEL 1 // PWM channel for Green LED
16  #define PWM_RED_CHANNEL 2 // PWM channel for Red LED
17  #define PWM_FREQ 5000 // 5 kHz PWM frequency
18  #define PWM_RES 10 // 10-bit resolution
19  hw_timer_t *blinkTimer = nullptr; // Timer for Blink mode
20  volatile int blinkStep = 0; // Step in Blink sequence
21  int currentMode = 0; // 0: All OFF, 1: Alternate Blink, 2: All ON, 3: PWM Fading
22  bool prevBtnMode = HIGH; // Previous state of Mode button
23  bool prevBtnReset = HIGH; // Previous state of Reset button
24  unsigned long lastDebounceTime = 0; // For button debounce
25  const int debounceDelay = 500; // 500ms debounce delay
26

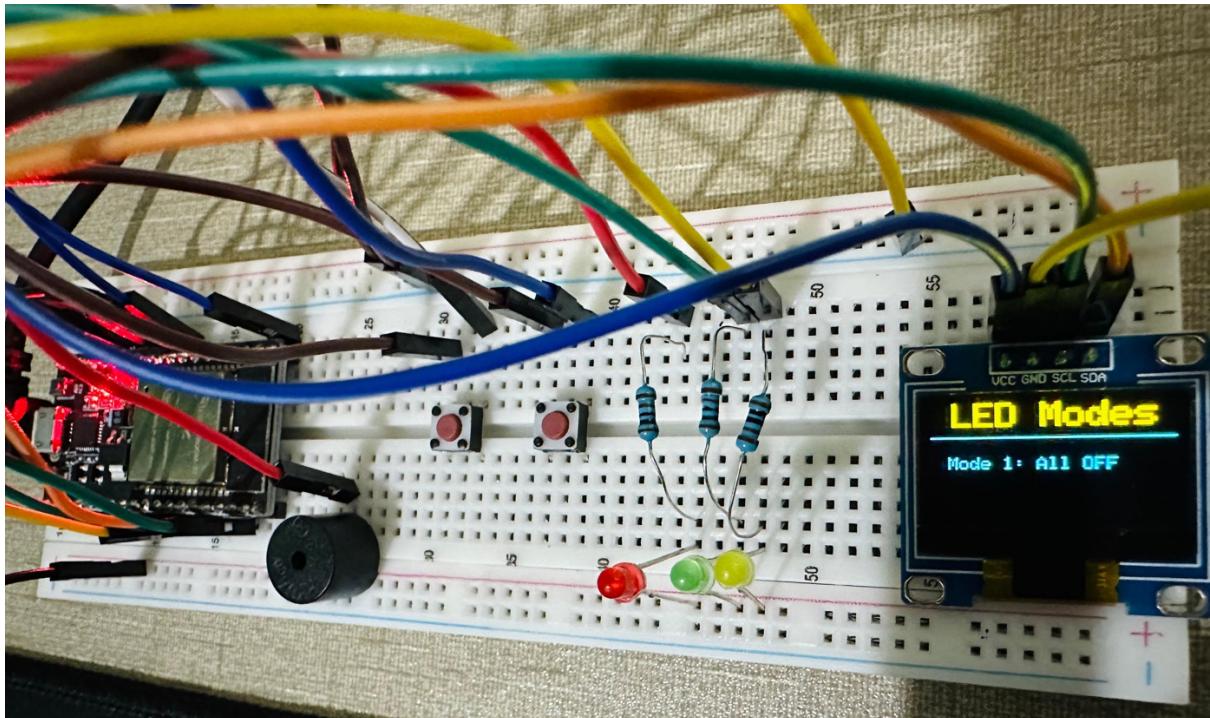
PROBLEMS    OUTPUT    DEBUG CONSOLE    TERMINAL    PORTS    GITLENS
TERMINAL
Writing at 0x00027486... (27 %)
Writing at 0x0002ca53... (36 %)
Writing at 0x000322f2... (45 %)
Writing at 0x000378fa... (54 %)
Writing at 0x0003d1c7... (63 %)
Writing at 0x00042765... (72 %)
Writing at 0x00047b66... (81 %)
Writing at 0x0004f5a0... (90 %)
Writing at 0x000585f3... (100 %)
Wrote 311872 bytes (174547 compressed) at 0x00010000 in 4.4 seconds (effective 567.3 kbit/s)...
Hash of data verified.

Leaving...
Hard resetting via RTS pin...
[TIMING] Took 10.34 seconds
* Terminal will be reused by tasks, press any key to close it.

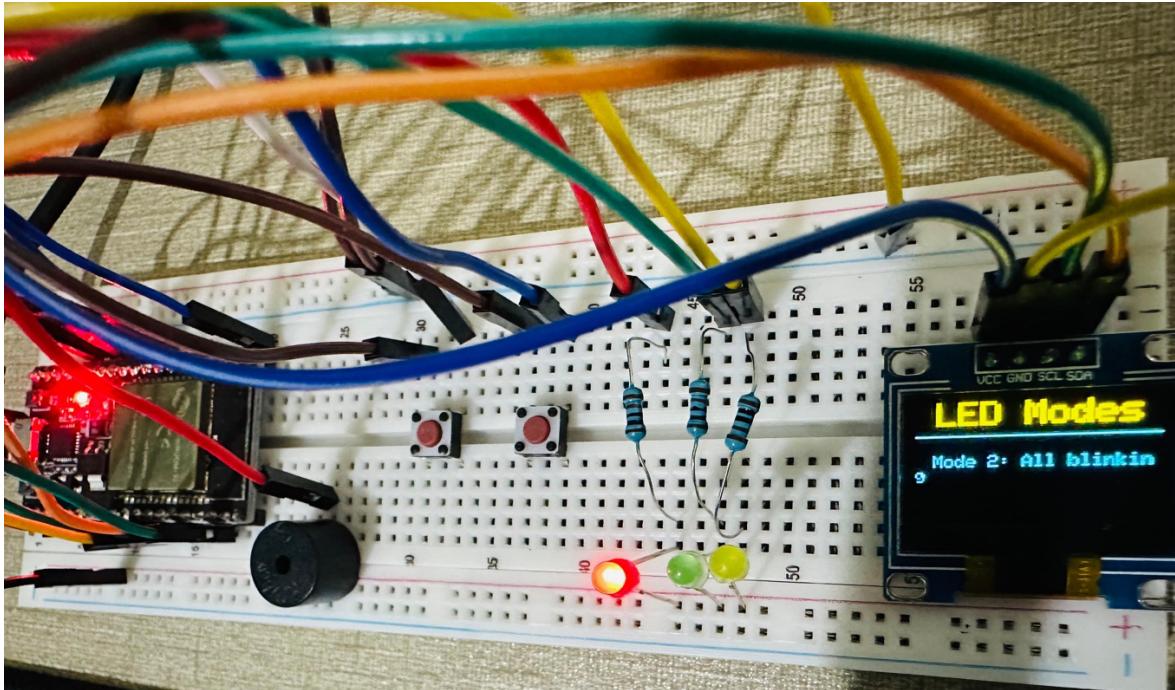
```

Hardware Output:

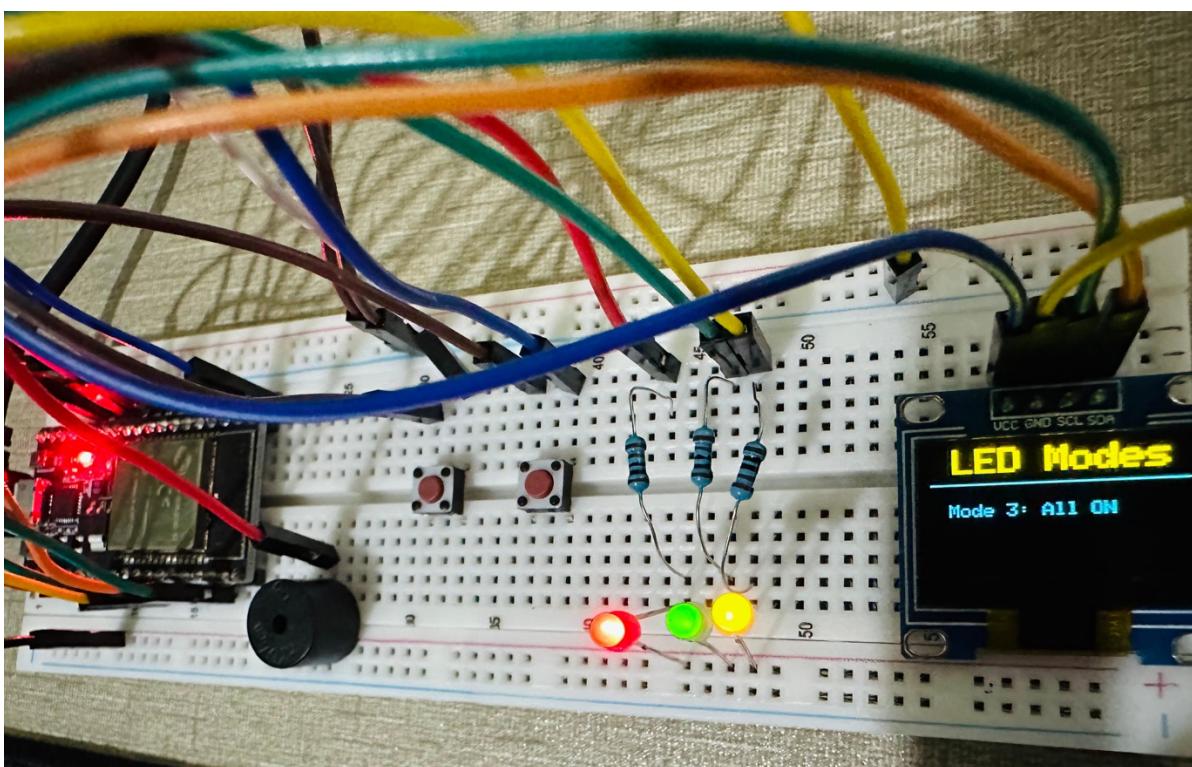
Mode 1: All LEDs OFF:



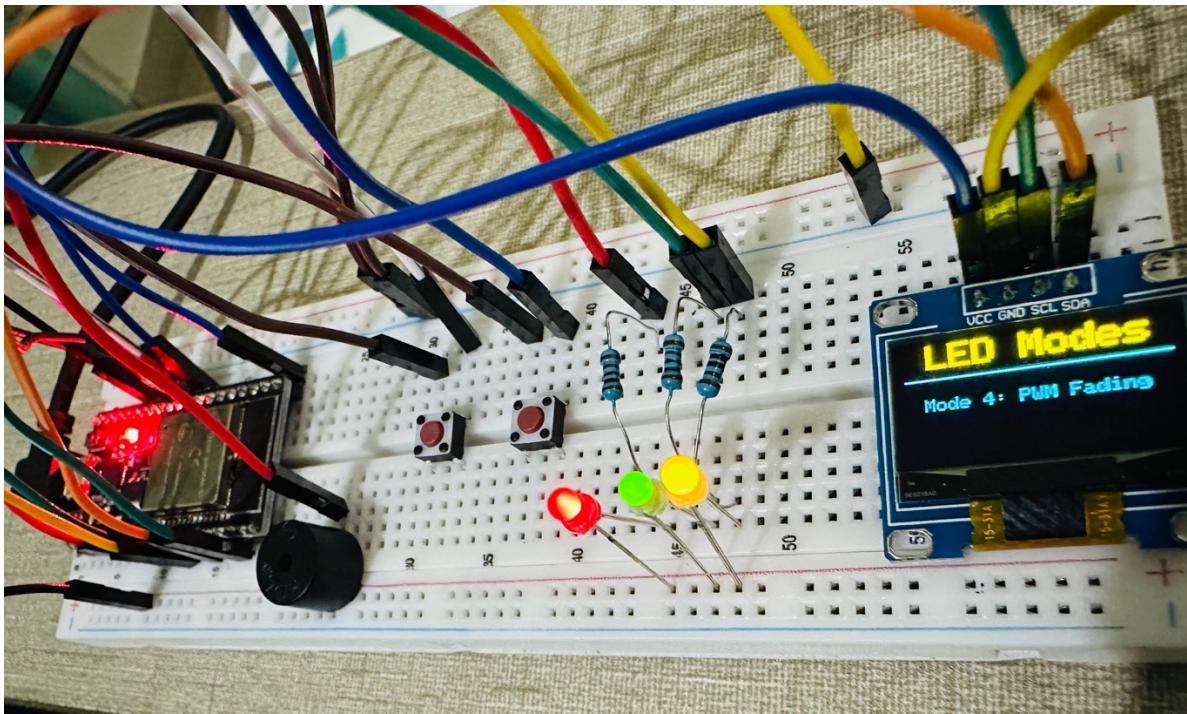
Mode 2: All LEDs Blinking:



Mode 3: All LEDs ON:



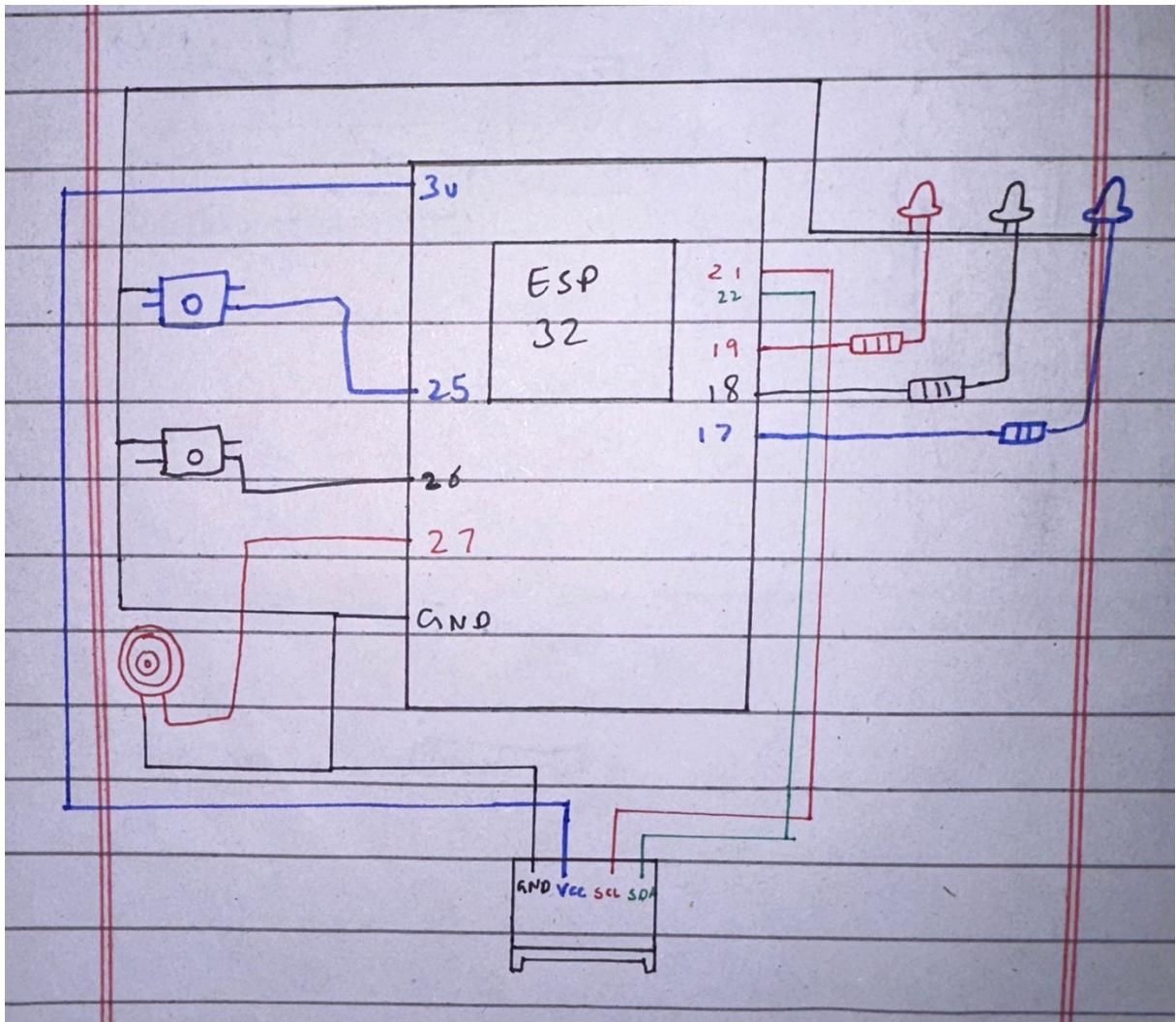
Mode 4: PWM Fading:



Wokwi Project Link:

<https://wokwi.com/projects/444997908922052609>

Circuit Diagram:



Pin Map Diagram:

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