

National Textile University, Faisalabad



Department of Computer Science

Name:	Samar Khalid
Class:	BSCS_5B
Registration No:	23-NTU-CS-1091
Assignment:	1 (task a)
Course Name:	Embedded IoT systems
Submitted To:	Sir Nasir Mahmood
Submission Date:	23-10-2025

Assignment 1

Task A

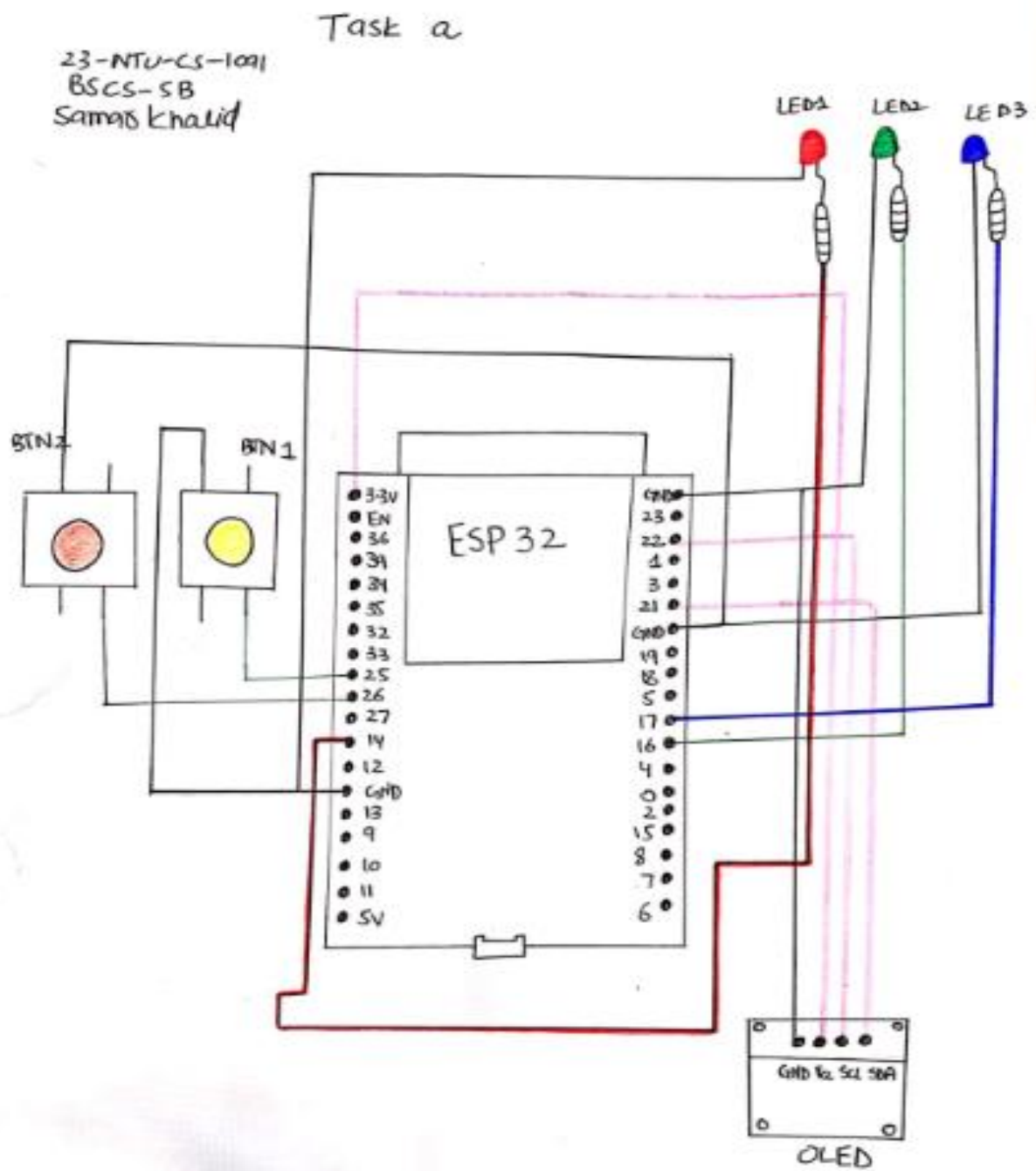
Task Explanation:

This ESP32 code controls three LEDs and an OLED display using two buttons. There are four modes: All OFF, Alternate Blink, All ON, and PWM Fade. Button 1 changes the mode, and Button 2 resets to “All OFF.” The OLED shows the current mode. In Alternate Blink, LEDs 1 and 3 blink opposite LED 2; in All ON, all LEDs stay lit and in PWM Fade, all LEDs smoothly fade in and out. The program provides easy control of LED patterns with clear OLED feedback.

Pin Diagram:

Device Name	Device Pin	ESP-32 pin
OLED	vcc	3.3 V
OLED	GND	GND
OLED	SDA	GPIO21
OLED	SCL	GPIO22
LED1	Cathode (Short leg)	GND
LED1	Anode (Long leg)	GPIO14
LED2	Cathode (Short leg)	GND
LED2	Anode (Long leg)	GPIO16
LED3	Cathode (Short leg)	GND
LED3	Anode (Long leg)	GPIO17
Button1	One leg	GND
Button1	Other leg	GPIO25
Button2	One leg	GND
Button2	Other leg	GPIO26

Circuit Diagram:



Code Screenshot:

```
task-a > src > main.cpp > onTimer()
1  //Week3 Assignment_1 Task_a
2  //Use one button to cycle through LED modes and use second button to reset to OFF
3  //Embedded IoT System Fall-2025
4  //Samar Khalid                      Reg no:23-NTU-CS-1091
5
6  #include <Arduino.h>
7  #include <Wire.h>
8  #include <Adafruit_GFX.h>
9  #include <Adafruit_SSD1306.h>
10
11 //OLED Setup
12 #define SCREEN_WIDTH 128
13 #define SCREEN_HEIGHT 64
14 Adafruit_SSD1306 display(SCREEN_WIDTH, SCREEN_HEIGHT, &Wire, -1);
15
16 //Define Pin numbers
17 #define LED1 14
18 #define LED2 16
19 #define LED3 17
20 #define BTN1 25 //Mode button
21 #define BTN2 26 //Reset button
22
23 //PWM Setup
24 #define PWM_CH1 0
25 #define PWM_CH2 1
26 #define PWM_CH3 2
27 #define FREQ 5000
28 #define RES 8
29
```

```

30 //Timer
31 hw_timer_t *My_timer = nullptr;
32 volatile bool toggleState = false;
33
34 //Global variable
35 int mode = 0; // 0=OFF, 1=Alternate Blink, 2=All ON, 3=PWM Fade
36 bool lastBtn1 = HIGH, lastBtn2 = HIGH;
37 unsigned long lastDebounce = 0;
38 const int debounceDelay = 200;
39
40 //Display Helper
41 void showMode() {
42     display.clearDisplay();
43     display.setTextSize(2);
44     display.setTextColor(SSD1306_WHITE);
45     display.setCursor(10, 20);
46     switch (mode) {
47         case 0: display.print("All OFF"); break;
48         case 1: display.print("Alternate Blink"); break;
49         case 2: display.print("All ON"); break;
50         case 3: display.print("PWM Fade"); break;
51     }
52     display.display();
53 }
54
55 //Timer ISR
56 void IRAM_ATTR onTimer() {
57     if (mode != 1) return;
58

```

```

59     toggleState = !toggleState;
60
61     ledcWrite(PWM_CH1, toggleState ? 255 : 0);
62     ledcWrite(PWM_CH2, toggleState ? 0 : 255);
63     ledcWrite(PWM_CH3, toggleState ? 255 : 0);
64 }
65
66 //Setup
67 void setup() {
68     Serial.begin(115200);
69
70     pinMode(LED1, OUTPUT);
71     pinMode(LED2, OUTPUT);
72     pinMode(LED3, OUTPUT);
73     pinMode(BTN1, INPUT_PULLUP);
74     pinMode(BTN2, INPUT_PULLUP);
75
76     //OLED setup
77     if (!display.begin(SSD1306_SWITCHCAPVCC, 0x3C)) {
78         Serial.println(F("SSD1306 allocation failed"));
79         for (;;);
80     }
81     display.clearDisplay();
82     display.display();
83

```

```

84     //PWM setup
85     ledcSetup(PWM_CH1, FREQ, RES);
86     ledcSetup(PWM_CH2, FREQ, RES);
87     ledcSetup(PWM_CH3, FREQ, RES);
88     ledcAttachPin(LED1, PWM_CH1);
89     ledcAttachPin(LED2, PWM_CH2);
90     ledcAttachPin(LED3, PWM_CH3);
91
92     //Timer setup for blinking
93     My_timer = timerBegin(0, 80, true);           //1 tick = 1  $\mu$ s (80 MHz / 80)
94     timerAttachInterrupt(My_timer, &onTimer, true);
95     timerAlarmWrite(My_timer, 500000, true);      //Toggle every 0.5 sec
96     timerAlarmEnable(My_timer);
97
98     //Initial state
99     ledcWrite(PWM_CH1, 0);
100    ledcWrite(PWM_CH2, 0);
101    ledcWrite(PWM_CH3, 0);
102    showMode();
103 }
104
105 //Loop
106 void loop() {
107     bool btn1 = digitalRead(BTN1);
108     bool btn2 = digitalRead(BTN2);
109

```

```

110     //Debounce logic
111     if (millis() - lastDebounce > debounceDelay) {
112         if (btn1 == LOW && lastBtn1 == HIGH) {
113             mode = (mode + 1) % 4;
114             showMode();
115             lastDebounce = millis();
116         }
117         if (btn2 == LOW && lastBtn2 == HIGH) {
118             mode = 0;
119             showMode();
120             lastDebounce = millis();
121         }
122     }
123
124     lastBtn1 = btn1;
125     lastBtn2 = btn2;
126
127     //Mode Logic
128     switch (mode) {
129         case 0: // All OFF
130             ledcWrite(PWM_CH1, 0);
131             ledcWrite(PWM_CH2, 0);
132             ledcWrite(PWM_CH3, 0);
133             break;
134
135         case 1: //Alternate Blink (handled by timer)
136             //nothing in loop
137             break;
138

```

```

135     case 1: //Alternate Blink (handled by timer)
136         //nothing in loop
137         break;
138
139     case 2: // All ON
140         ledcWrite(PWM_CH1, 255);
141         ledcWrite(PWM_CH2, 255);
142         ledcWrite(PWM_CH3, 255);
143         break;
144
145     case 3: //PWM Fade (all together)
146         for (int d = 0; d <= 255 && mode == 3; d++) {
147             ledcWrite(PWM_CH1, d);
148             ledcWrite(PWM_CH2, d);
149             ledcWrite(PWM_CH3, d);
150             delay(10);
151             if (digitalRead(BTN1) == LOW || digitalRead(BTN2) == LOW) return;
152         }
153         for (int d = 255; d >= 0 && mode == 3; d--) {
154             ledcWrite(PWM_CH1, d);
155             ledcWrite(PWM_CH2, d);
156             ledcWrite(PWM_CH3, d);
157             delay(10);
158             if (digitalRead(BTN1) == LOW || digitalRead(BTN2) == LOW) return;
159         }
160         break;
161     }
162 }

```

VS code Build success:

The screenshot shows the Visual Studio Code interface with the following components:

- Explorer (Left):** Displays the project structure. The 'task-a' folder is selected, showing sub-files like '.pio', '.vscode', 'include', 'lib', 'src', 'main.cpp', 'test', 'gitignore', 'diagram.json', and 'platformio.ini'.
- Main Editor:** Shows the 'main.cpp' file with the following code:


```

task-a > src > C: main.cpp > onTimer()
14 Adafruit_SSD1306 display(SCREEN_WIDTH, SCREEN_HEIGHT, &Wire, -1);
15
16 //Define Pin numbers
17 #define LED1 14
18 #define LED2 16
19 #define LED3 17
20 #define BTN1 25 //Mode button
21 #define BTN2 26 //Reset button
22
23 //PWM Setup
24 #define PWM_CH1 0
25 #define PWM_CH2 1
26 #define PWM_CH3 2
27 #define FREQ 5000
28 #define RES 8
29
30 //Timer
31 hw_timer_t *My_timer = nullptr;
32 volatile bool toggleState = false;
33

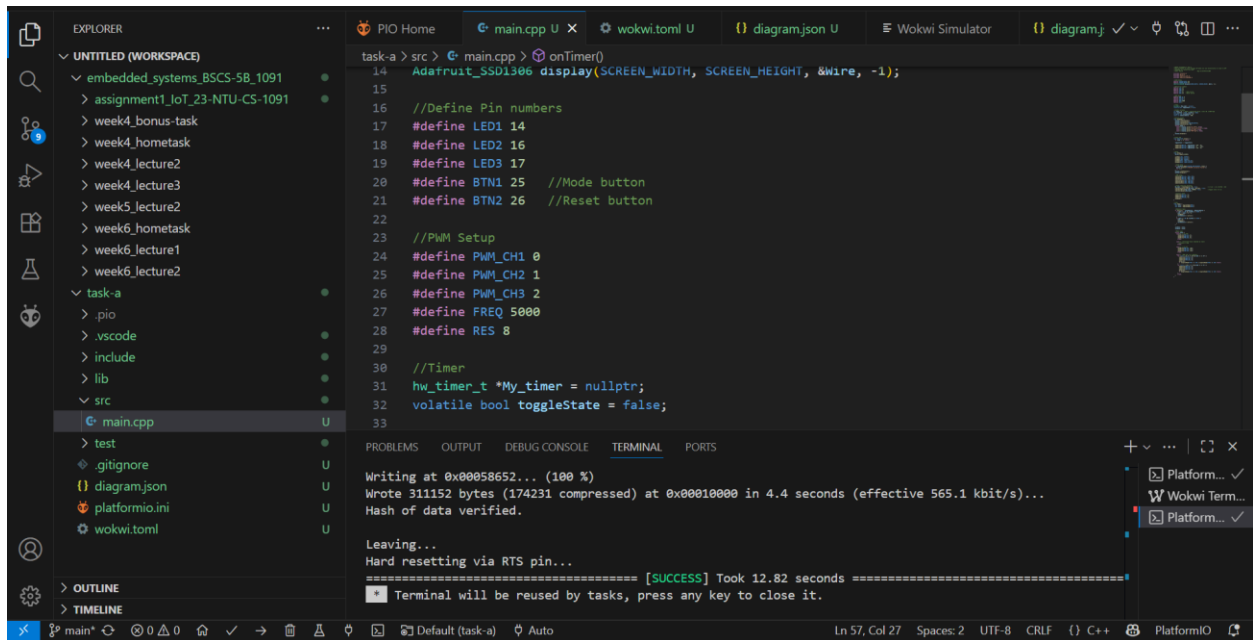
```
- Output (Bottom):** Shows the build process for 'PlatformIO: Build (task-a)'. The output indicates that the 'firmware.bin' file was successfully built using 'esptool.py v4.9.0' and 'Building .pio/build/nodemcu-32s/firmware.bin'. The build took 6.15 seconds and was successful.


```

Flash: [==] 23.7% (used 310793 bytes from 1310720 bytes)
Building .pio/build/nodemcu-32s/firmware.bin
esptool.py v4.9.0
Creating esp32 image...
Merged 2 ELF sections
Successfully created esp32 image.
===== [SUCCESS] Took 6.15 seconds =====
Terminal will be reused by tasks, press any key to close it.

```

Upload on ESP-32 success:



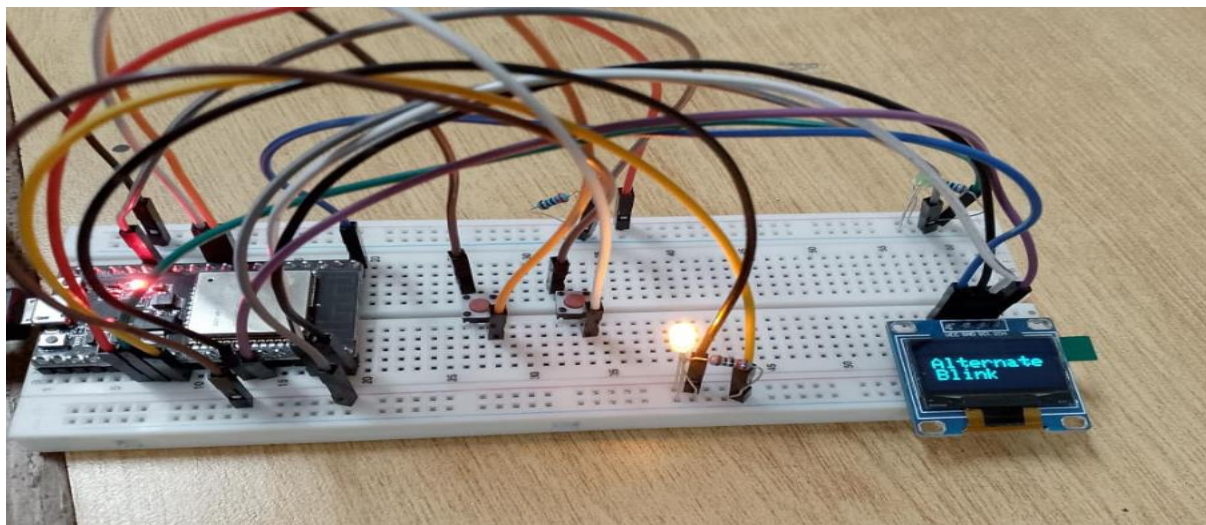
Wokwi Link:

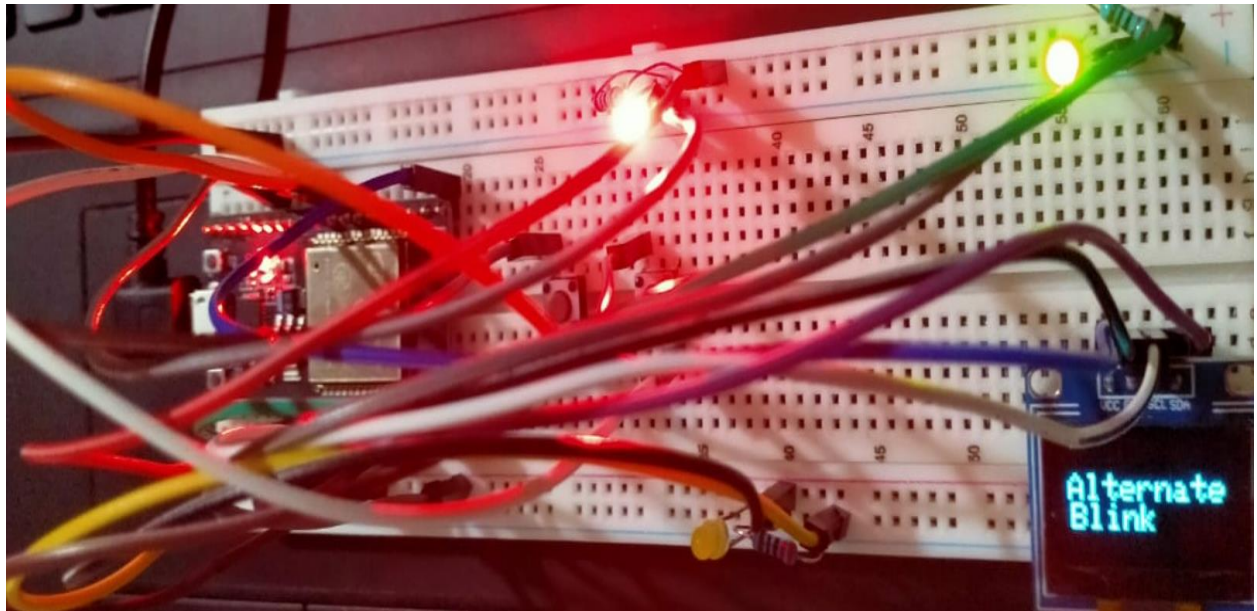
<https://wokwi.com/projects/445509046222119937>

Output on kit:

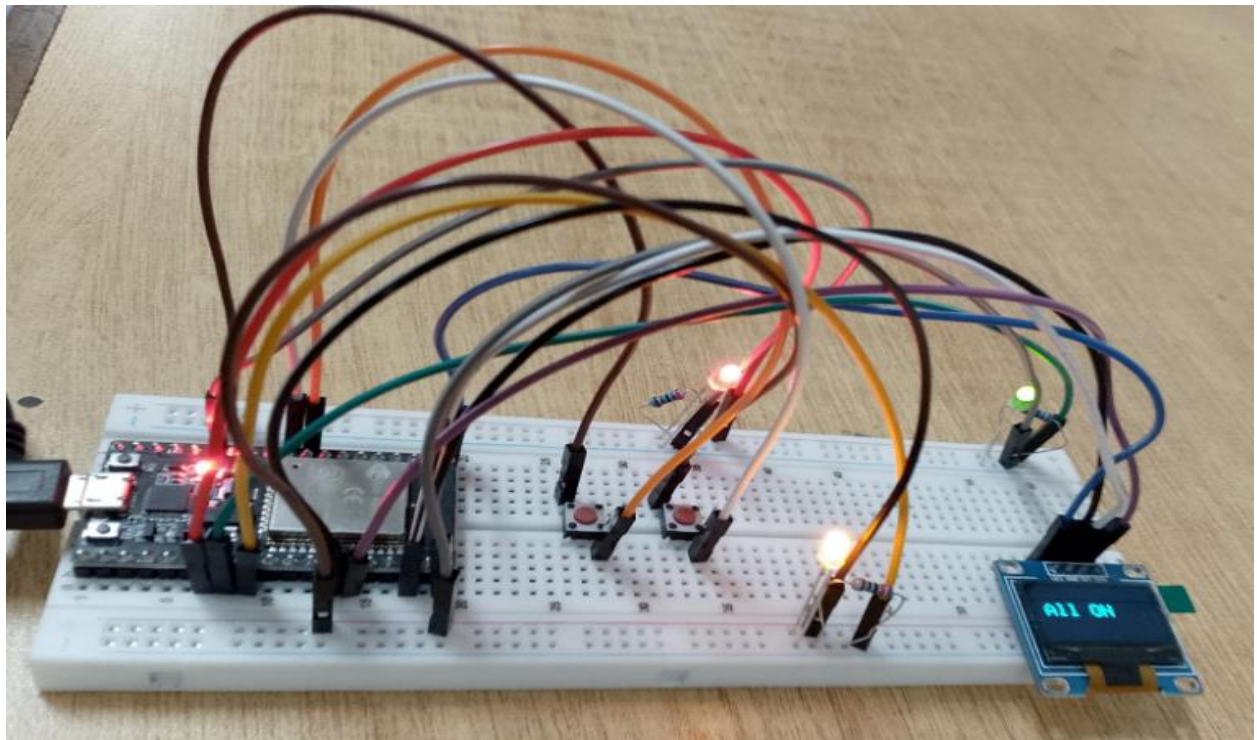
Button 1 Functionalities:

Alternate Blink:



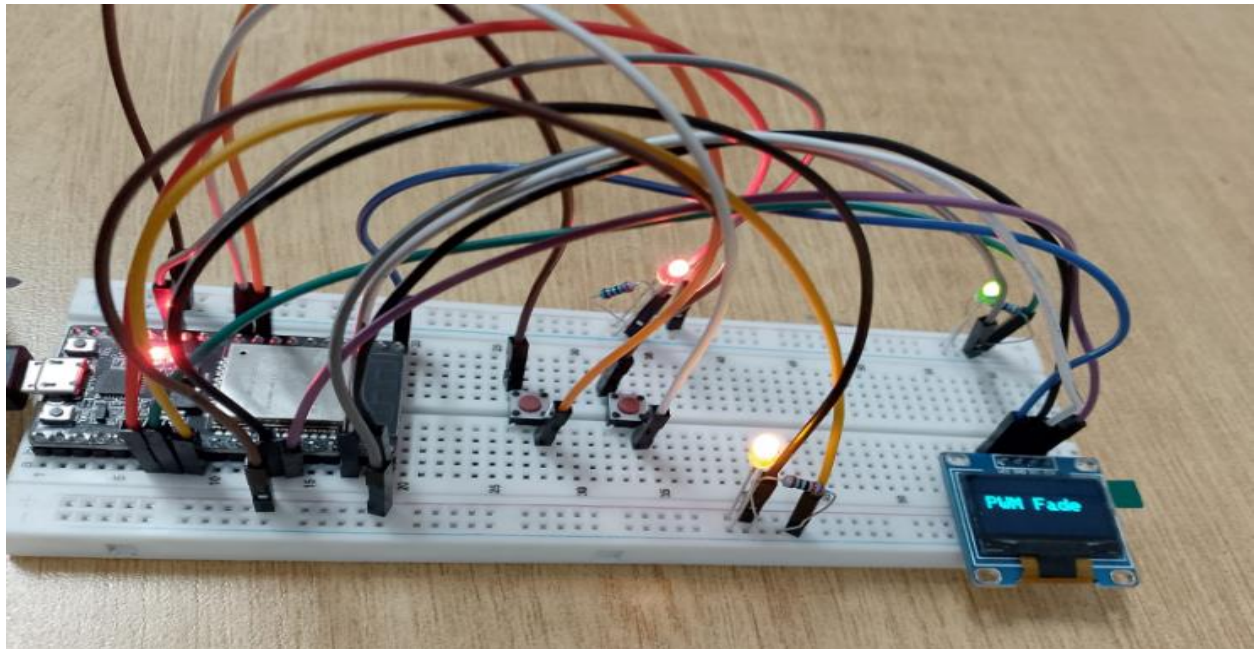


All LEDs ON:

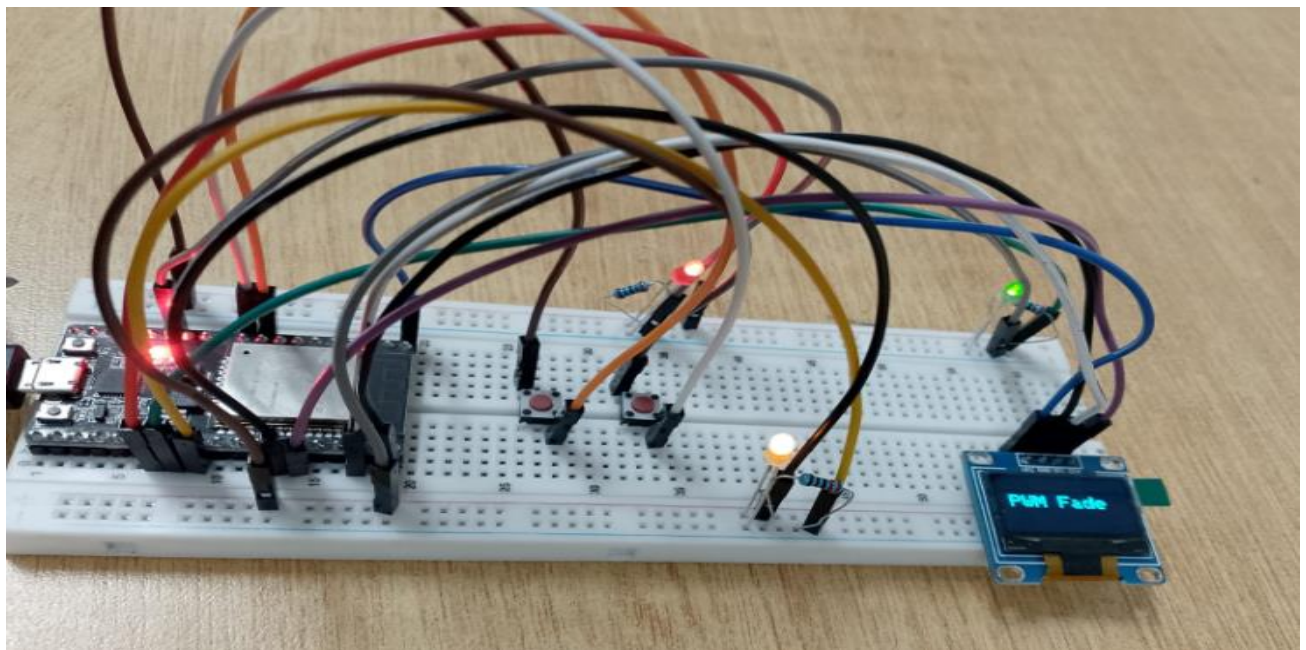


PWM Fade:

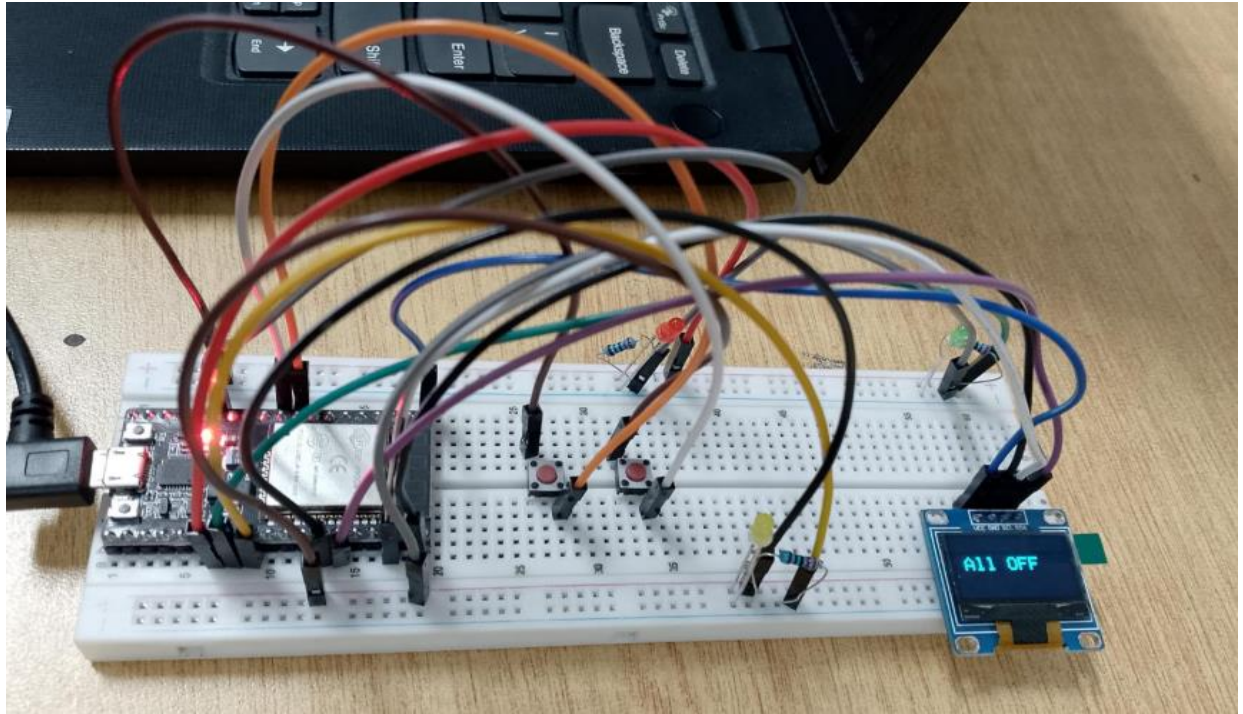
Full Glow:



FADE:

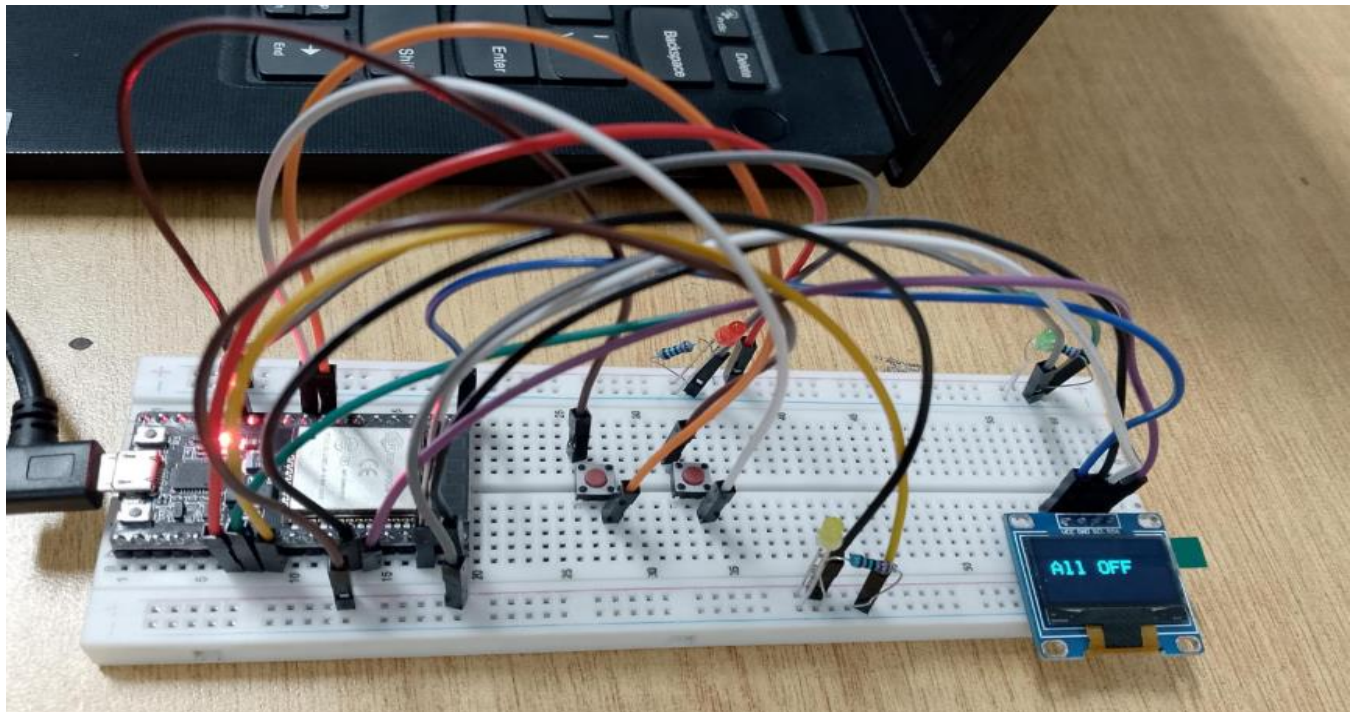


ALL OFF:



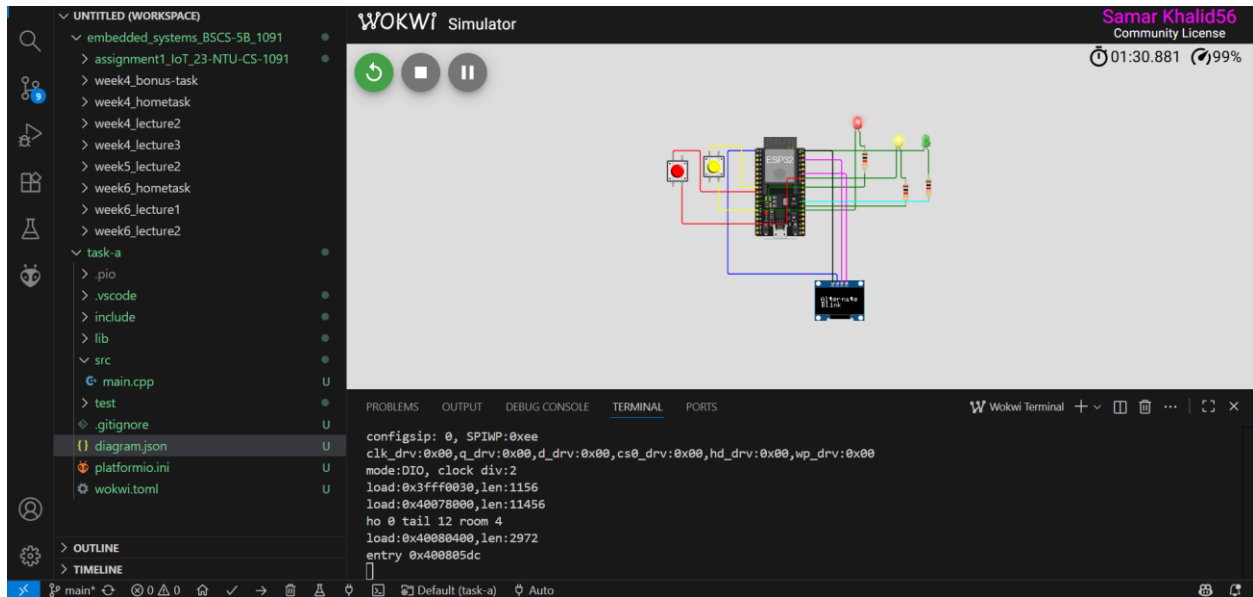
Button 2:

On pressing button 2, all LEDS reset to OFF.

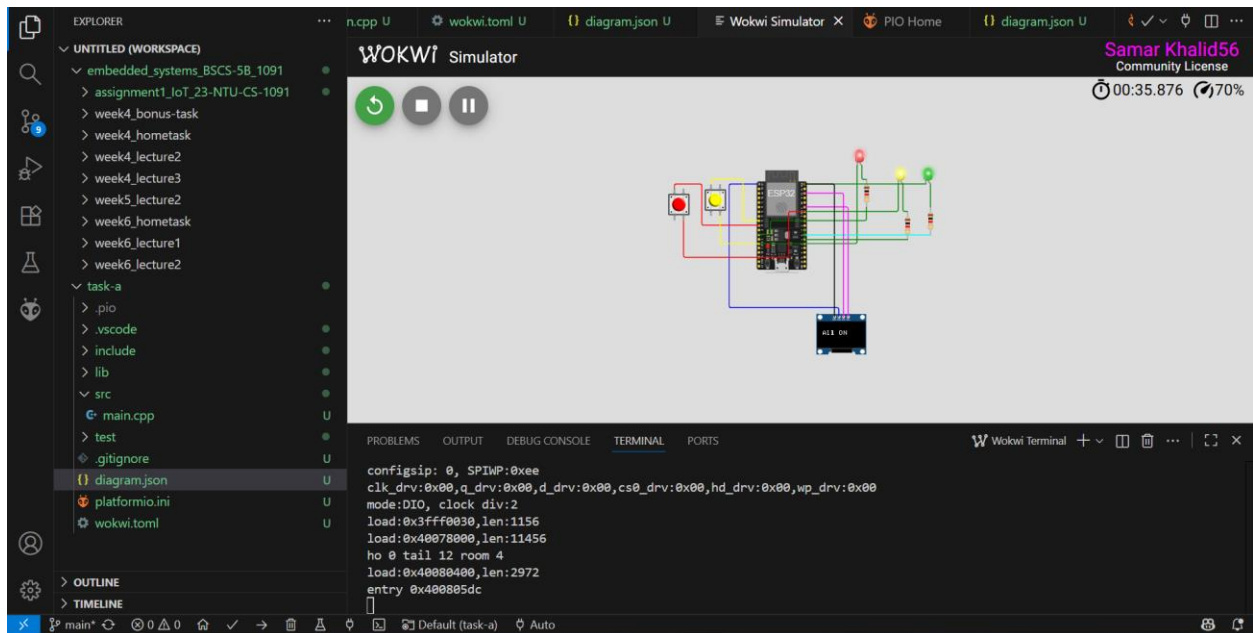


Output on Wokwi:

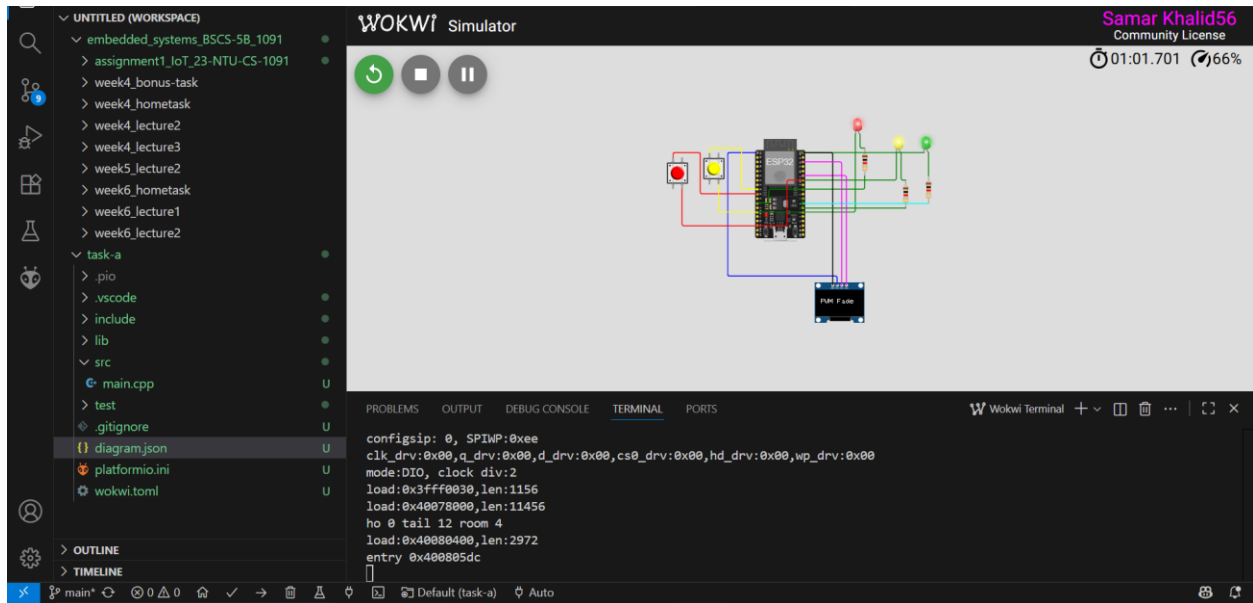
Alternate Blink:



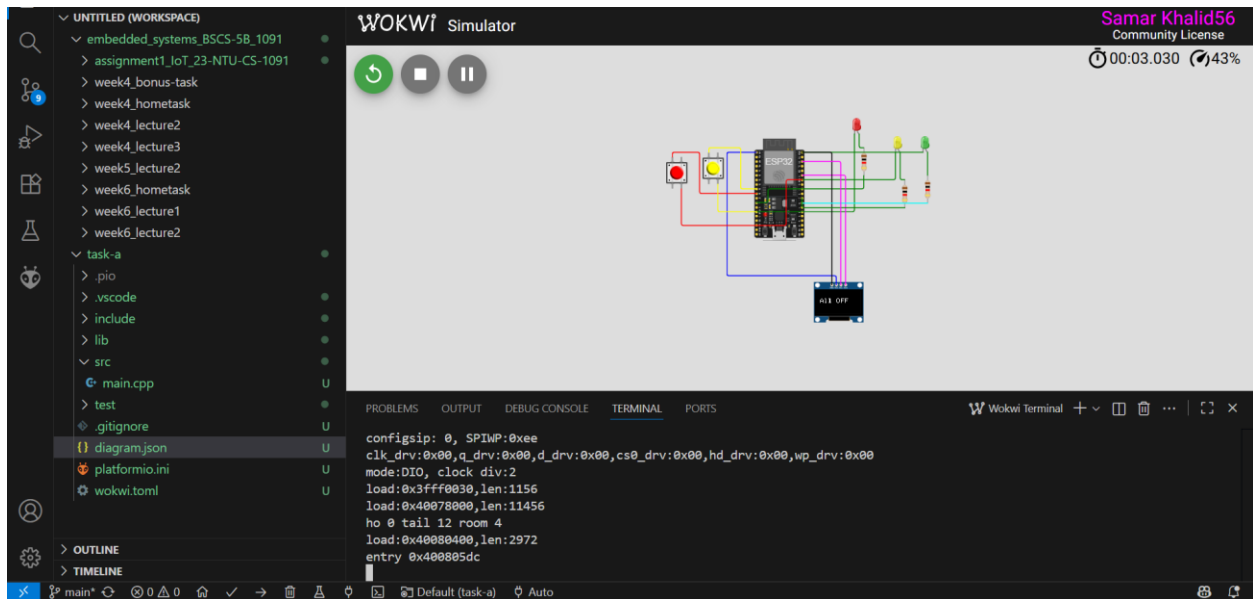
ALL LEDS ON:



PWM Fade:



ALL OFF:



Handwritten code:

Handwritten Code

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

// OLED Setup
#define SCREEN_WIDTH 128
#define SCREEN_HEIGHT 64
Adafruit_SSD1306 display(SCREEN_WIDTH,
    SCREEN_HEIGHT, &Wire, -1);

// Define Pin numbers
#define LED1 14
#define LED2 16
#define LED3 17
#define BTN1 25 // Mode button
#define BTN2 26 // Reset button

// PWM Setup
#define PWM_CH1 0
#define PWM_CH2 1
#define PWM_CH3 2
#define FREQ 5000
#define RES 8
```

```
// Timer
```

```
hw_timer_t *My_timer = nullptr;
```

```
volatile bool toggleState = false;
```

```
// Global variable
```

```
int mode = 0 // 0 = OFF, 1 = Alternate Blink
```

```
// 2 = All ON, 3 = PWM Fade
```

```
bool lastBtn1 = HIGH, lastBtn2 = HIGH;
```

```
unsigned long lastDebounce = 0;
```

```
const int debounceDelay = 200;
```

```
// Display Helper
```

```
void ShowMode() {
```

```
    display.clearDisplay();
```

```
    display.setTextSize(2);
```

```
    display.setTextColors(SSD1306_WHITE);
```

```
    display.setCursor(10, 20);
```

```
    switch (mode) {
```

```
        case 0: display.print("All OFF"); break;
```

```
        case 1: display.print("Alternate Blink"); break;
```

```
        case 2: display.print("All ON"); break;
```

```
        case 3: display.print("PWM Fade"); break;
```

```
    }
```

```
    display.display();
```

```
}
```

// Timer ISR

void IRAM_ATTR onTimer() {

if (mode != 1) return;

toggleState = !toggleState;

ledcWrite(PWM_CH1, toggleState ? 255 : 0);

ledcWrite(PWM_CH2, toggleState ? 0 : 255);

ledcWrite(PWM_CH3, toggleState ? 255 : 0);

}

// Setup

void setup() {

Serial.begin(115200);

pinMode(LED1, OUTPUT);

pinMode(LED2, OUTPUT);

pinMode(LED3, OUTPUT);

pinMode(BTN1, INPUT_PULLUP);

pinMode(BTN2, INPUT_PULLUP);

// OLED Setup

if (!display.begin(SSD1306_SWITCHCAPVCC, 0x3C)) {

Serial.println(F("SSD1306 allocation failed"));

for(;;);

}

display.clearDisplay();

display.display();

// Debounce Logic

```
if (millis() - lastDebounce > debounceDelay) {  
  if (btn1 == LOW && lastBtn1 == HIGH) {  
    mode = (mode + 1) % 4;  
    showMode();  
    lastDebounce = millis();  
  }  
  if (btn2 == LOW && lastBtn2 == HIGH) {  
    mode = 0;  
    showMode();  
    lastDebounce = millis();  
  }  
}  
lastBtn1 = btn1;  
lastBtn2 = btn2;
```

// Mode Logic

```
switch (mode) {  
  case 0: // All OFF  
    ledcWrite(PWM_CH1, 0);  
    ledcWrite(PWM_CH2, 0);  
    ledcWrite(PWM_CH3, 0);  
    break;  
  case 1: // Alternate Blink (handled by timer)  
    // nothing in loop  
    break;
```

case 2 : // All ON

```
ledcWrite (PWM-CH1, 255);  
ledcWrite (PWM-CH2, 255);  
ledcWrite (PWM-CH3, 255);  
break;
```

case 3: // PWM Fade (all together)

```
for (int d = 0; d <= 255 && mode == 3; d++) {
```

```
    ledcWrite (PWM-CH1, d);
```

```
    ledcWrite (PWM-CH2, d);
```

```
    ledcWrite (PWM-CH3, d);
```

```
    delay(10);
```

```
    if (digitalRead (BTN1) == LOW ||
```

```
        digitalRead (BTN2) == LOW) return;
```

```
}
```

```
for (int d = 255; d >= 0 && mode == 3; d--) {
```

```
    ledcWrite (PWM-CH1, d);
```

```
    ledcWrite (PWM-CH2, d);
```

```
    ledcWrite (PWM-CH3, d);
```

```
    delay(10);
```

```
    if (digitalRead (BTN1) == LOW ||
```

```
        digitalRead (BTN2) == LOW) return;
```

```
}
```

```
break;
```

```
}
```

```
}
```

