

# National Textile University, Faisalabad



## Department of Computer Science

<b>Name:</b>	Rizwana Bashir
<b>Class:</b>	BSCS-5 <sup>th</sup> -B
<b>Registration No:</b>	23-NTU-CS-FL-1088
<b>Assignment:</b>	1
<b>Submission Date:</b>	26-Oct-2025
<b>Course Name:</b>	Embedded IoT Systems
<b>Submitted To:</b>	Sir Nasir Mahmood

# Embedded IoT Systems Assignment 1

## Task A: LED Modes.

```
main.cpp x PIO Home
Assignment_1_Task_A_LED_Modes > src > main.cpp > ...
1  //Rizwana Bashir
2  //23-NTU-CS-1088
3  #include <Arduino.h>
4  #include <Wire.h>
5  #include <Adafruit_GFX.h>
6  #include <Adafruit_SSD1306.h>
7
8  // --- OLED Configuration ---
9  #define SCREEN_WIDTH 128
10 #define SCREEN_HEIGHT 64
11 #define OLED_ADDR 0x3C
12 Adafruit_SSD1306 display(SCREEN_WIDTH, SCREEN_HEIGHT, &Wire, -1);
13
14 // --- Pin Definitions ---
15 #define BUTTON1 34
16 #define BUTTON2 35
17 #define RED_LED 19
18 #define BLUE_LED 18
19 #define PURPLE_LED 17
20
21 // --- Global Variables ---
22 int mode = 1;
23 bool lastButton1 = HIGH;
24 bool lastButton2 = HIGH;
25 unsigned long lastButtonTime = 0;
26
27 // Variables for blinking/fading
28 unsigned long lastBlinkTime = 0;
29 int blinkState = 0;
30 int fadeValue = 0;
31 int fadeDirection = 1;
```

Assignment\_1\_Task\_A\_LED\_Modes > src > main.cpp > ...

```
32
33 // --- Function Prototype (required for PlatformIO) ---
34 void showMode();
35
36 void setup() {
37     Serial.begin(115200);
38     Wire.begin(); // I2C start for OLED
39
40     // Initialize OLED
41     if (!display.begin(SSD1306_SWITCHCAPVCC, OLED_ADDR)) {
42         Serial.println(F("OLED initialization failed!"));
43         while (true); // Stop if OLED fails
44     }
45
46     display.clearDisplay();
47     display.setTextSize(1);
48     display.setTextColor(SSD1306_WHITE);
49
50     // Set pin modes
51     pinMode(REDA_LED, OUTPUT);
52     pinMode(BLUE_LED, OUTPUT);
53     pinMode(PURPLE_LED, OUTPUT);
54     pinMode(BUTTON1, INPUT_PULLUP);
55     pinMode(BUTTON2, INPUT_PULLUP);
56
57     // Show initial mode
58     showMode();
59 }
60
61 void loop() {
62     bool button1State = digitalRead(BUTTON1);
63     bool button2State = digitalRead(BUTTON2);
```

Assignment\_1\_Task\_A\_LED\_Modes > src > main.cpp > ...

```
61 void loop() {
62     // --- Mode Button ---
63     if (button1State == LOW && lastButton1 == HIGH && millis() - lastButtonTime > 300) {
64         mode++;
65         if (mode > 4) mode = 1;
66         showMode();
67         lastButtonTime = millis();
68     }
69
70     // --- Reset Button ---
71     if (button2State == LOW && lastButton2 == HIGH && millis() - lastButtonTime > 300) {
72         mode = 1;
73         showMode();
74         lastButtonTime = millis();
75     }
76
77     lastButton1 = button1State;
78     lastButton2 = button2State;
79
80     // --- Mode Functions ---
81     switch (mode) {
82         case 1: // All OFF
83             analogWrite(RED_LED, 0);
84             analogWrite(BLUE_LED, 0);
85             analogWrite(PURPLE_LED, 0);
86             break;
87
88         case 2: // All LEDs alternate blink together
89             if (millis() - lastBlinkTime >= 500) {
90                 blinkState = !blinkState;
91                 analogWrite(RED_LED, blinkState ? 255 : 0);
92                 analogWrite(BLUE_LED, blinkState ? 255 : 0);
93                 analogWrite(PURPLE_LED, blinkState ? 255 : 0);
94                 lastBlinkTime = millis();
95             }
96     }
97 }
```

```
92     if (millis() - lastBlinkTime >= 500) {
99         break;
100     case 3: // All ON
101         analogWrite(RED_LED, 255);
102         analogWrite(BLUE_LED, 255);
103         analogWrite(PURPLE_LED, 255);
104         break;
105     case 4: // PWM fade (non-blocking)
106         if (millis() - lastBlinkTime >= 10) {
107             fadeValue += fadeDirection * 5;
108             if (fadeValue >= 255 || fadeValue <= 0) fadeDirection *= -1;
109
110             analogWrite(RED_LED, fadeValue);
111             analogWrite(BLUE_LED, 255 - fadeValue);
112             analogWrite(PURPLE_LED, (fadeValue / 2) + 50);
113
114             lastBlinkTime = millis();
115         }
116         break;
117     }
118 }
119 // --- OLED Display Function ---
120 void showMode() {
121     display.clearDisplay();
122     display.setCursor(0, 10);
123     display.setTextSize(2);
124     display.print("Mode ");
125     display.print(mode);
126     display.setTextSize(1);
127     display.setCursor(0, 40);
128 }
```

Ln 99, Col 13 Spaces: 2

**Code Built successfully:**

```
File Edit Selection View Go Run ... Untitled (Workspace)
EXPLORER
  UNTITLED (WORKSPACE)
  Assignment_1_Task_A_LED_Modes
    .pio
    .vscode
    include
    lib
    src
      main.cpp
    test
    .gitignore
    platformio.ini
  main.cpp
  test
  .gitignore
  platformio.ini

main.cpp
1 //Rizwana Bashir
2 //23-NTU-CS-1088
3 #include <Arduino.h>
4 #include <Wire.h>
5 #include <Adafruit_GFX.h>
6 #include <Adafruit_SSD1306.h>
7
8 // --- OLED Configuration ---
9 #define SCREEN_WIDTH 128
10 #define SCREEN_HEIGHT 64
11 #define OLED_ADDR 0x3C
12 Adafruit_SSD1306 display(SCREEN_WIDTH, SCREEN_HEIGHT, &Wire, -1);
13
14 // --- Pin Definitions ---
15 #define BUTTON1 34
16 #define BUTTON2 35
17 #define RED_LED 19
18 #define BLUE_LED 18
19 #define PURPLE_LED 17
20
21 // --- Global Variables ---
22 int mode = 1;

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
Writing at 0x00057fd6... (100 %)
Wrote 306240 bytes (171747 compressed) at 0x00010000 in 4.4 seconds (effective 552.4 kbit/s)...
Hash of data verified.
Leaving...
Hard resetting via RTS pin...
===== [SUCCESS] Took 19.70 seconds =====
Terminal will be reused by tasks, press any key to close it.

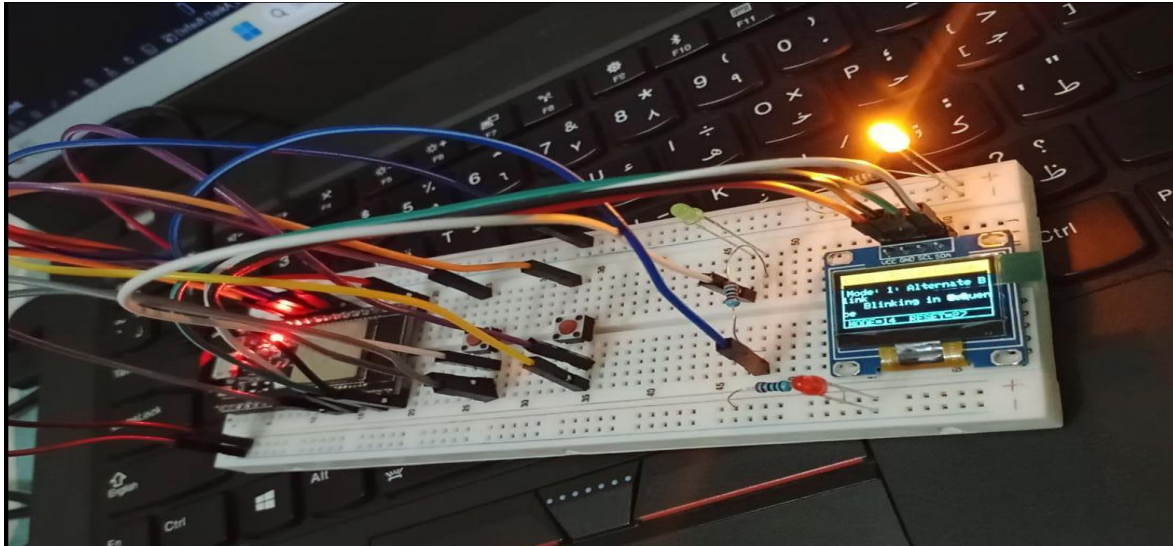
Ln 2, Col 17 Spaces: 2 UTF-8 CRLF {} C++ Go Live PlatformIO
```

---

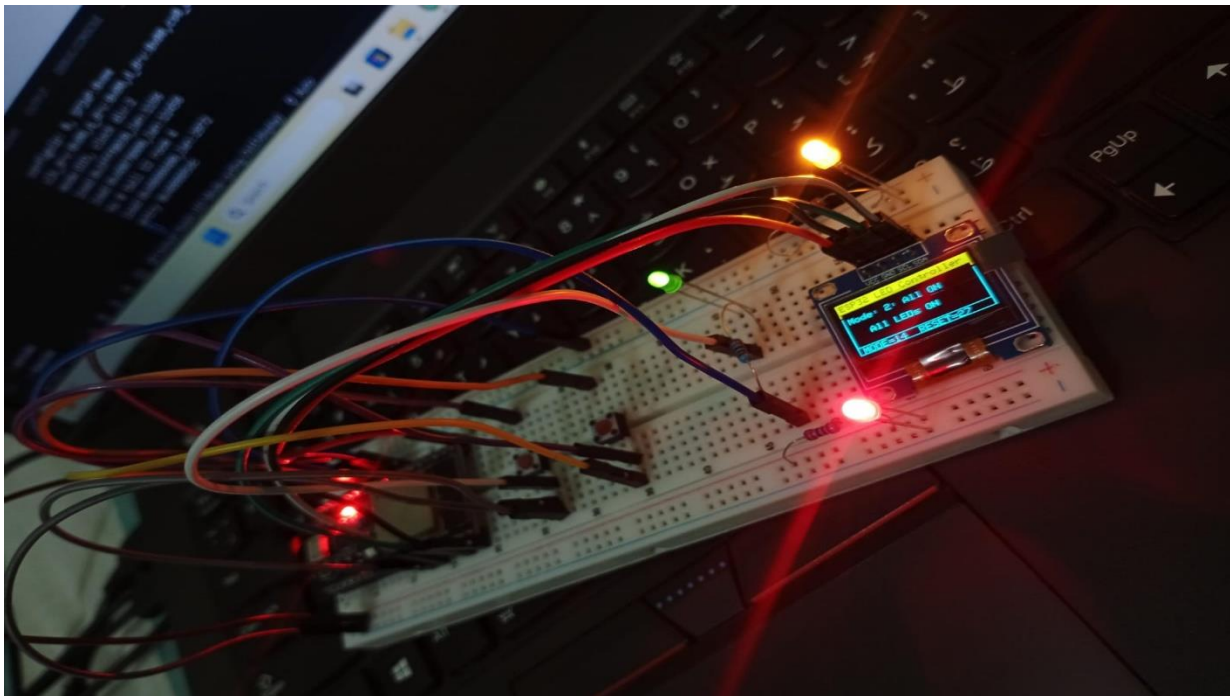
## Different Outputs on Hardware:

---

### 1. Alternate LEDs

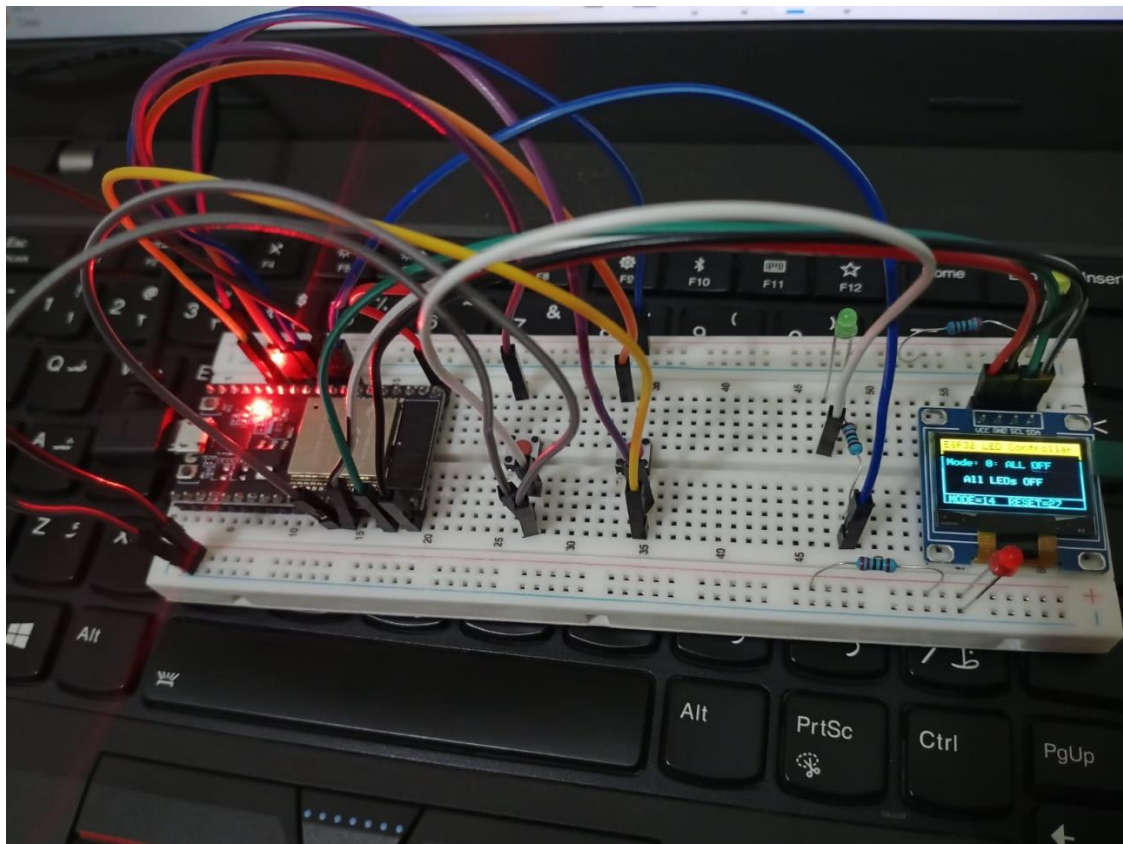
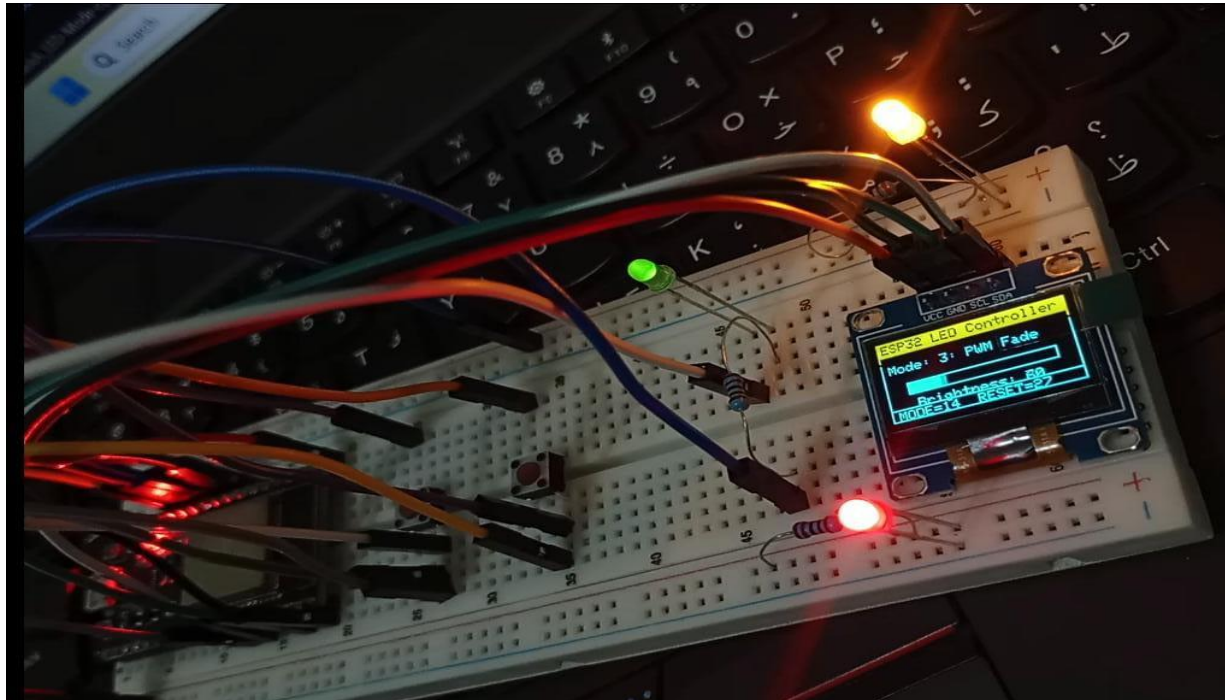


### 2. Blink LEDs





### 3.PWM Fading



## Handwritten code and Diagram

### Task A: Assignment #1 "One button to cycle through LED nodes"

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

//---OLED setup---
#define SCREEN_Width 128
#define SCREEN_Height 64
#define OLED_ADDR 0x3C
Adafruit_SSD1306 display(SCREEN_Width,
    SCREEN_Height, &Wire, -1);
// Pin configuration
#define Buttons 34
#define Button2 64
#define OLED_ADDR 0x3C
#define Red_LED 14
#define Blue_LED 18
#define Purple_LED 17
// Variables
int mode = 1;
bool lastButton1 = HIGH;
bool lastButton2 = HIGH;
unsigned long lastButtonTime = 0;

unsigned long lastBlinkTime = 0;
int blinkState = 0;
```

```
if (buttonState == LOW && button == HIGH &&
    millis() - lastButtonTime > 300) {
    mode = 1;
    ShowMode();
    lastButtonTime = millis();
}
// Mode functions.
Switch(Mode) {
    case 1: // All OFF
        AnalogWrite(Red_LED, 0);
        AnalogWrite(Blue_LED, 0);
        AnalogWrite(Purple_LED, 0);
        break;
    // All ON with fast Alternate blink
    case 2:
        if (millis() - lastBlinkTime > 500) {
            blinkState = !blinkState;
            analogWrite(1, blinkState);
            lastBlinkTime = millis();
        }
        break;
    // All ON
    case 3:
        if (millis() - lastBlinkTime > 10)
            analogWrite(LED, 255);
        break;
```

```
int fadeValue = 0;
int fadeDirection = 1;

void showMode();

void setup() {
    Serial.begin(115200);
    Wire.begin();
    if (!display.begin(I2C("OLED failed!")));
        while (true);
    }
    display.clearDisplay();
    // OLED display
    pinMode(Red_LED, output);
    pinMode(Blue_LED, output);
    pinMode(Purple_LED, output);
    pinMode(Button1, input_pullup);
    pinMode(Button2, input_pullup);

    ShowMode();
}

void loop() {
    bool buttonState = digitalRead(Button1);
    bool buttonState = digitalRead(Button2);
```

```
case 4: // PWM display
    if (millis() - lastBlink > 10) {
        fadeValue += fadeDirection * 5;
        if (fadeValue >= 255 || fadeValue <= 0)
            fadeDirection *= -1;

        AnalogWrite(Red_LED, fadeValue);
        AnalogWrite(Blue_LED, 255 - fadeValue);
        AnalogWrite(Purple_LED, fadeValue / 2 + 50);

        lastBlink = millis();
    }
    break;
}

void showMode() {
    // OLED display
    display.clearDisplay();
    display.setCursor(0, 10);
    display.setTextSize(2);
    display.display();
    Switch(Mode) {
        case 1: display.print("All LED's OFF"); break;
        case 2: display.print("All blink Alternate"); break;
        case 3: display.print("All LED's ON"); break;
        case 4: display.print("All PWM Fade"); break;
    }
    display.display();
}
```



**Figure:**

