

# National Textile University,

## Faisalabad



### Department of Computer Science

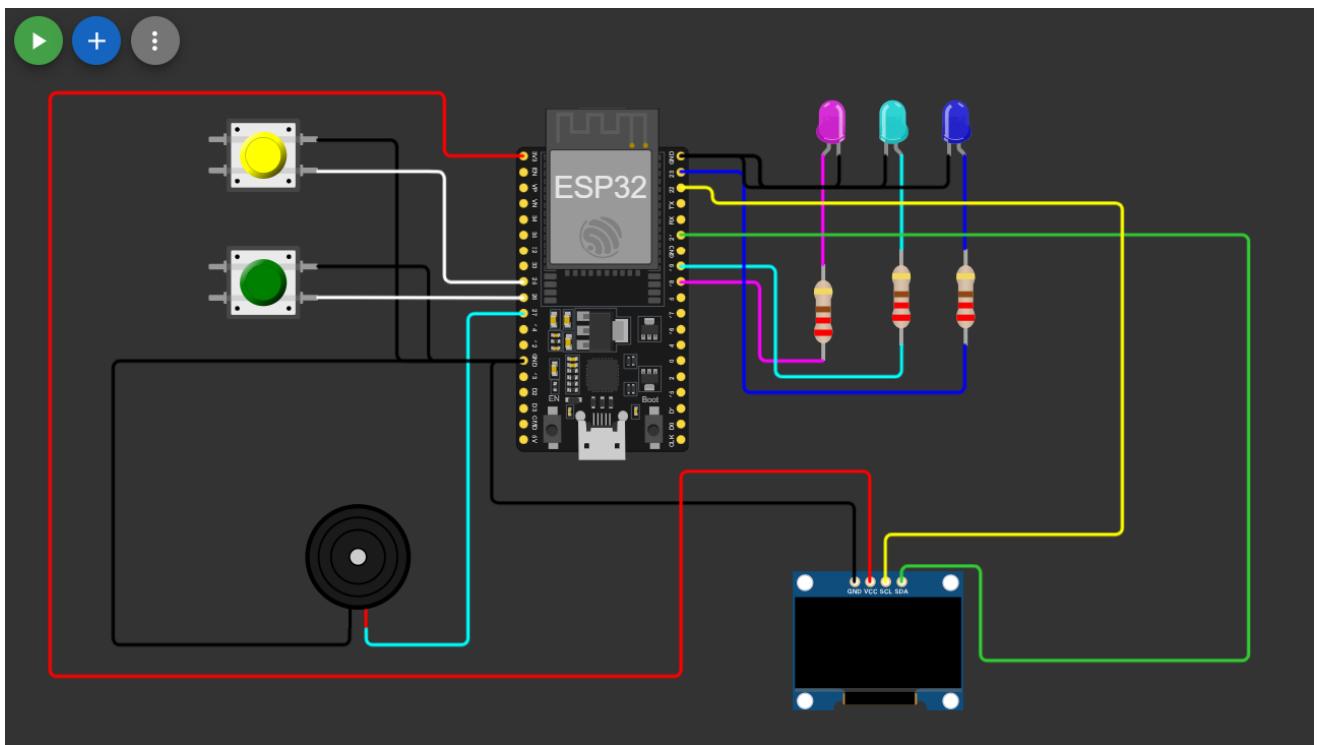
<b>Name:</b>	Amna
<b>Class:</b>	BSCS-5A
<b>Registration No:</b>	23-NTU-CS-1013
<b>Assignment No:</b>	1
<b>Course Name:</b>	Embedded IOT Systems
<b>Submitted To:</b>	Sir Nasir Mahmood
<b>Submission Date:</b>	26-10-2025

## Documentation

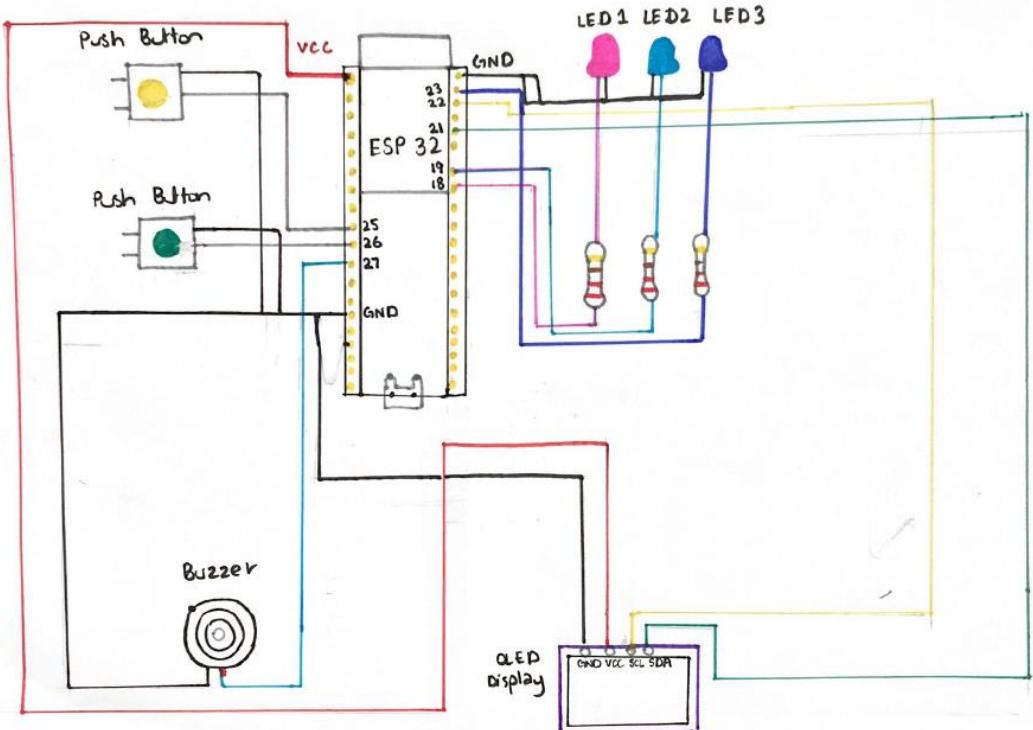
### Question 3 — Implementation

#### 1. Circuit Diagram:

1.1. 2 push buttons • 3 LEDs • 1 buzzer • 1 OLED



## 1.2. Hand sketch



## 2. Pin Map

Components	ESP32 PIN
LED 1 Pink	GPIO 18
LED 2 Cyan	GPIO 19
LED 3 Blue	GPIO 23
Buzzer	GPIO 27
Push Button Yellow	GPIO 25
Push Button Green	GPIO 26
OLED Display SDA	GPIO 21
OLED Display SCL	GPIO 22
OLED Display VCC	3.3 V
OLED Display GND	GND
Common Ground	GND

## **2.1. Description:**

### **Task A – LED Mode Control**

- Yellow Push button is used to change LED modes
- Green Push button resets to OFF.
- The OLED shows the current mode.

### **Task B – Single Button Press Detection**

- Green button is used for both actions.
- Short press → Toggle LEDs
- Long press → Play buzzer tone
- The OLED shows the press type.

## **3. Task A**

**Coding: Use one button to cycle through LED modes (display the current state on the OLED):**

- 1. Both OFF**
- 2. Alternate blink**
- 3. Both ON**
- 4. PWM fade Use the second button to reset to OFF.**

### **3.1. Code**

```

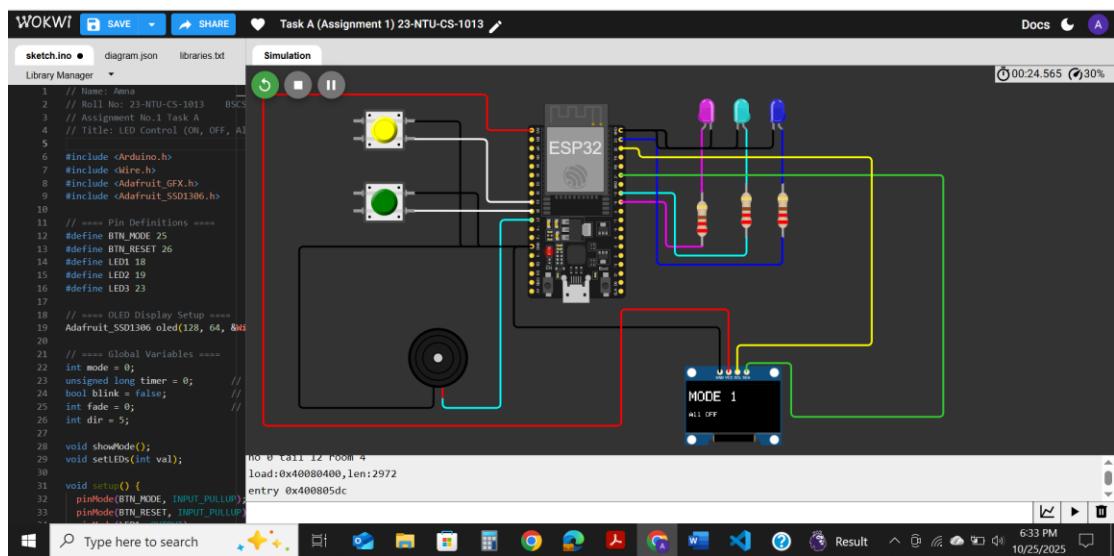
Task A > src > main.cpp > ...
1  // Name: Amna
2  // Roll No: 23-NTU-CS-1013      BSCS-5A
3  // Assignment No.1 Task A
4  // Title: LED Control (ON, OFF, Alternate Blink, PWM Fade)
5
6  #include <Arduino.h>
7  #include <Wire.h>
8  #include <Adafruit_GFX.h>
9  #include <Adafruit_SSD1306.h>
10 // === Pin Definitions ===
11 #define BTN_MODE 25
12 #define BTN_RESET 26
13 #define LED1 18
14 #define LED2 19
15 #define LED3 23
16
17 // === OLED Display Setup ===
18 Adafruit_SSD1306 oled(128, 64, &Wire, -1);
19
20 // === Global Variables ===
21 int mode = 0;                      // Timer for delays
22 unsigned long timer = 0;            // Toggle state for blinking
23 bool blink = false;                // PWM brightness level
24 int fade = 0;
25 int dir = 5;
26
27 void showMode();
28 void setLEDs(int val);
29
30 void setup() {
31     pinMode(BTN_MODE, INPUT_PULLUP);
32     pinMode(BTN_RESET, INPUT_PULLUP);
33     pinMode(LED1, OUTPUT);
34     pinMode(LED2, OUTPUT);
35     pinMode(LED3, OUTPUT);
36
37     // Initialize I2C and OLED
38     Wire.begin(21, 22);
39     oled.begin(SSD1306_SWITCHCAPVCC, 0x3C);
40     showMode();
41 }
42
43
44 void loop() {
45     // --- Handle Mode Button ---
46     if (digitalRead(BTN_MODE) == LOW) {
47         delay(200); // debounce delay
48         mode = (mode + 1) % 4;
49         showMode();
50     }
51
52     // --- Handle Reset Button ---
53     if (digitalRead(BTN_RESET) == LOW) {
54         delay(200); // debounce delay
55         mode = 0;
56         setLEDs(0);
57         showMode();
58     }
59
60     // --- Mode Operations ---
61     if (mode == 0) {
62         // Mode 1: All LEDs OFF
63         setLEDs(0);
64     }
65     else if (mode == 1) {
66         // Mode 2: Alternate Blinking LEDs
67         if (millis() - timer > 500) {
68             timer = millis();
69             blink = !blink;
70         }
71     }
72 }

```

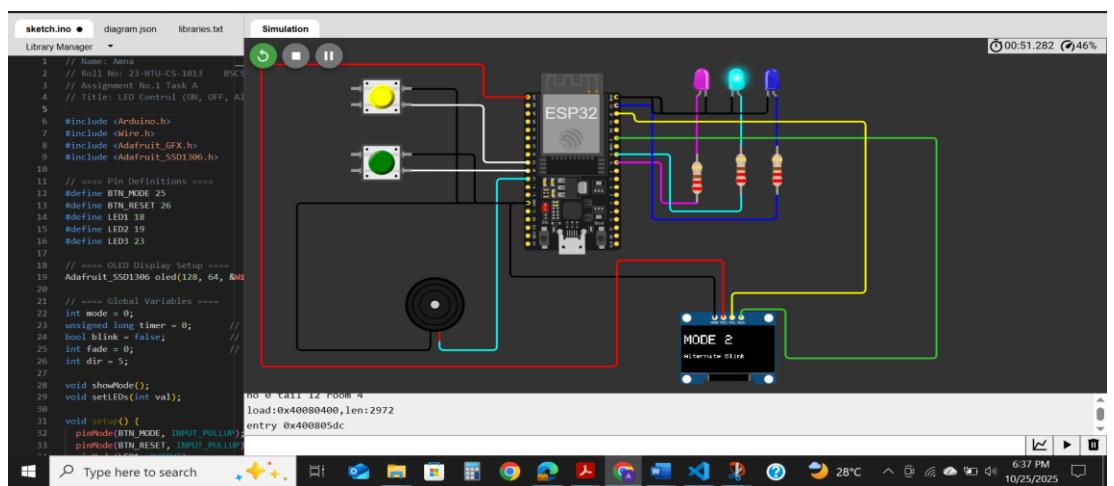
```
71     if (blink) {
72         analogWrite(LED1, 255);
73         analogWrite(LED2, 0);
74         analogWrite(LED3, 255);
75     } else {
76         analogWrite(LED1, 0);
77         analogWrite(LED2, 255);
78         analogWrite(LED3, 0);
79     }
80 }
81 }
82 else if (mode == 2) {
83     // Mode 3: All LEDs ON
84     setLEDs(255);
85 }
86 else if (mode == 3) {
87     // Mode 4: PWM Fade Effect
88     if (millis() - timer > 20) {
89         timer = millis();
90         fade += dir;
91         if (fade >= 255 || fade <= 0) dir = -dir;
92         analogWrite(LED1, fade);
93         analogWrite(LED2, fade);
94         analogWrite(LED3, fade);
95     }
96 }
97 }
98
99 // ===== Helper Functions =====
100 void setLEDs(int val) {
101     analogWrite(LED1, val);
102     analogWrite(LED2, val);
103     analogWrite(LED3, val);
104 }
105
106 void showMode() {
107     oled.clearDisplay();
108     oled.setTextSize(2);
109     oled.setTextColor(WHITE);
110     oled.setCursor(0, 10);
111     oled.print("MODE ");
112     oled.println(mode + 1);
113     oled.setTextSize(1);
114     oled.setCursor(0, 40);
115
116     if (mode == 0) oled.print("All OFF");
117     if (mode == 1) oled.print("Alternate Blink");
118     if (mode == 2) oled.print("All ON");
119     if (mode == 3) oled.print("Fade Mode");
120
121     oled.display();
122 }
123
```

### 3.2. Output Screenshots

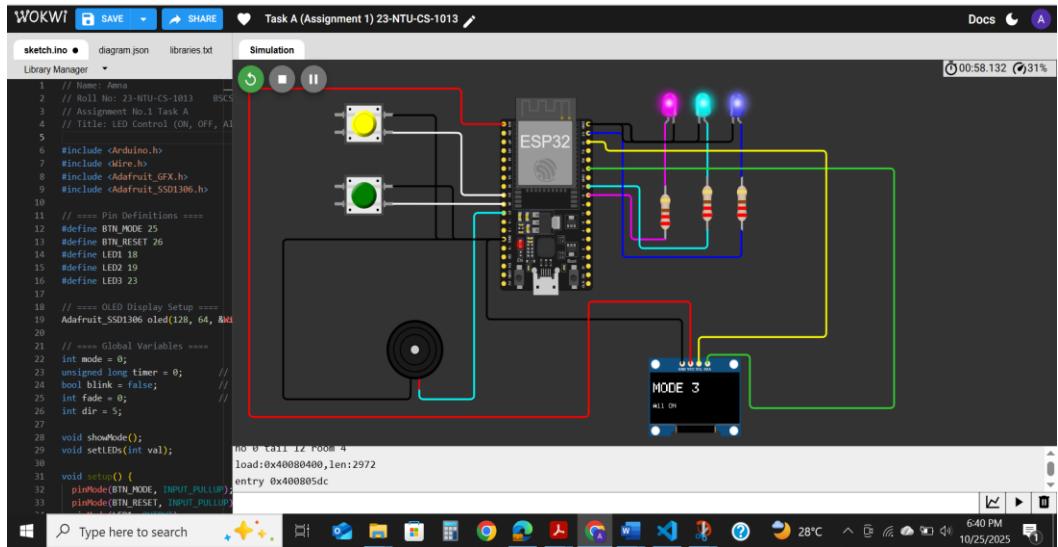
#### a. Mode 1: All OFF



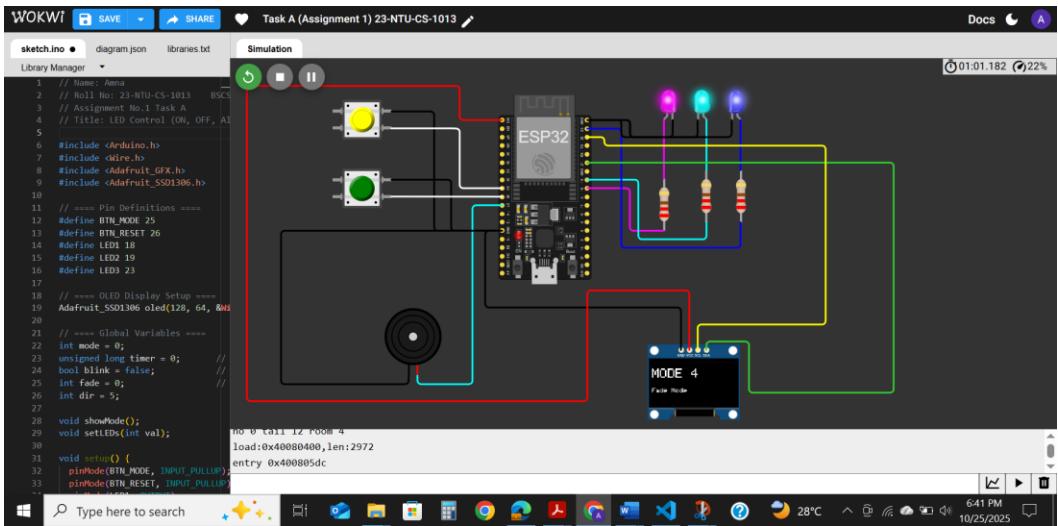
#### b. Mode 2: Alternate Blink



### c. Mode 2: All ON



### d. PWM Fade



### 3.3. Wokwi Link

<https://wokwi.com/projects/445523771586063361>

## 4. Task B

**Coding: Use a single button with press-type detection (display the event on the OLED):**

- Short press → toggle LED • Long press (> 1.5 s) → play a buzzer tone

### 4.1. Code

```
Task B > src > C main.cpp > handleLongPress()
1 // Name: Amna
2 // Roll No: 23-NTU-CS-1013
3 // Assignment No.1
4 // Task B
5 // Title: Buzzer Tone + LED Toggle
6
7 #include <Wire.h>
8 #include <Adafruit_GFX.h>
9 #include <Adafruit_SSD1306.h>
10
11 #define OLED_WIDTH 128
12 #define OLED_HEIGHT 64
13 Adafruit_SSD1306 display(OLED_WIDTH, OLED_HEIGHT, &Wire, -1);
14
15 // pin connections
16 #define LED1 18
17 #define LED2 19
18 #define LED3 23
19 #define BUZZER 27
20 #define BTN_MODE 25 // button in yellow color
21
22 #define LONG_PRESS_TIME 1500 // 1.5 sec for long press
23
24
25 bool ledState = false;
26 bool isPressed = false;
27 unsigned long pressStart = 0;
28
29 void setup() {
```

```
Task B > src > main.cpp > OLED_HEIGHT
29 void setup() {
30     Serial.begin(115200); // start serial monitor for debugging
31
32     // setting up pins
33     pinMode(LED1, OUTPUT);
34     pinMode(LED2, OUTPUT);
35     pinMode(LED3, OUTPUT);
36     pinMode(BUZZER, OUTPUT);
37     pinMode(BTN_MODE, INPUT_PULLUP); // using pull-up since button goes to GND
38
39     // initialize OLED display
40     if (!display.begin(SSD1306_SWITCHCAPVCC, 0x3C)) {
41         Serial.println("OLED not found :(");
42         while (1); // stop here if display not connected
43     }
44
45     // show startup message
46     display.clearDisplay();
47     display.setTextSize(1);
48     display.setTextColor(SSD1306_WHITE);
49     display.setCursor(0, 0);
50     display.println("System Ready...");
51     display.display();
52
53     // make sure everything starts off
54     digitalWrite(LED1, LOW);
55     digitalWrite(LED2, LOW);
56     digitalWrite(LED3, LOW);
57     digitalWrite(BUZZER, LOW);
58 }
59
60 void loop() {
61     bool buttonPressed = (digitalRead(BTN_MODE) == LOW); // check if button pressed
62
63     // when button is pressed
64     if (buttonPressed && !isPressed) {
65         isPressed = true;
66         pressStart = millis(); // record time when pressed
67     }
68
69     // when button is released
70     if (!buttonPressed && isPressed) {
71         unsigned long pressDuration = millis() - pressStart;
72         isPressed = false;
73
74         // check if it was long or short press
75         if (pressDuration >= LONG_PRESS_TIME) {
76             handleLongPress();
77         } else {
78             handleShortPress();
79         }
80     }
81 }
82
83 // function for short press
84 // toggles LEDs on/off and shows it on screen
85 void handleShortPress() {
```

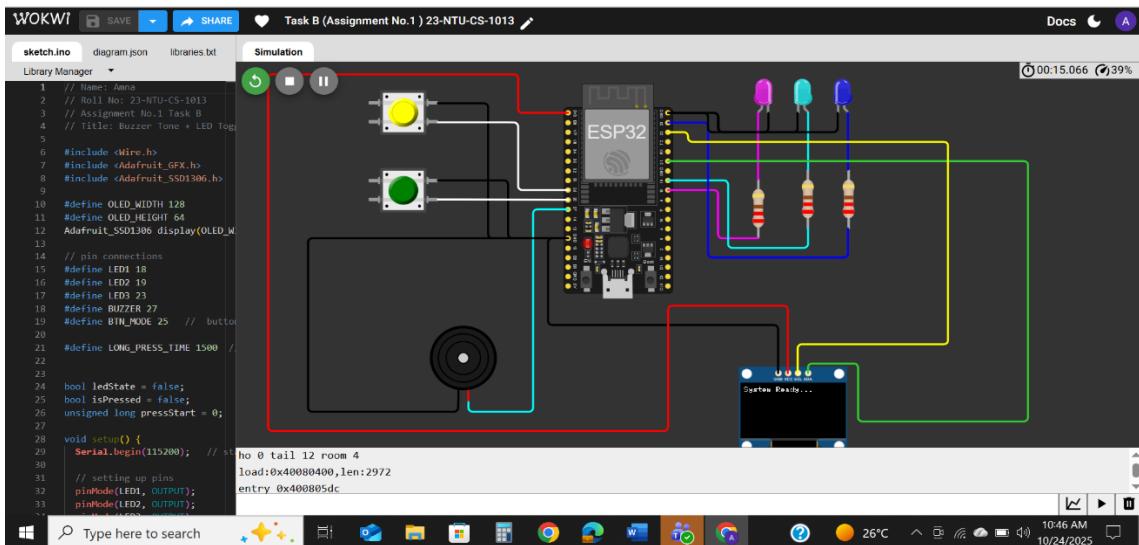
```

86     ledState = !ledState;    // change LED state
87
88     digitalWrite(LED1, ledState);
89     digitalWrite(LED2, ledState);
90     digitalWrite(LED3, ledState);
91
92     display.clearDisplay();
93     display.setCursor(0, 0);
94     display.println("Short Press");
95     display.println(ledState ? "LEDs turned ON" : "LEDs turned OFF");
96     display.display();
97     Serial.println("short press detected");
98 }
99
100 // function for long press
101 // plays buzzer sound for 0.5 second
102 void handleLongPress() {
103     display.clearDisplay();
104     display.setCursor(0, 0);
105     display.println("Long Press");
106     display.println("Buzzer playing...");
107     display.display();
108
109     Serial.println("Long press detected");
110
111     tone(BUZZER, 1000);    // 1kHz tone
112     delay(500);           // play for half second
113     noTone(BUZZER);       // stop buzzer
114 }
115

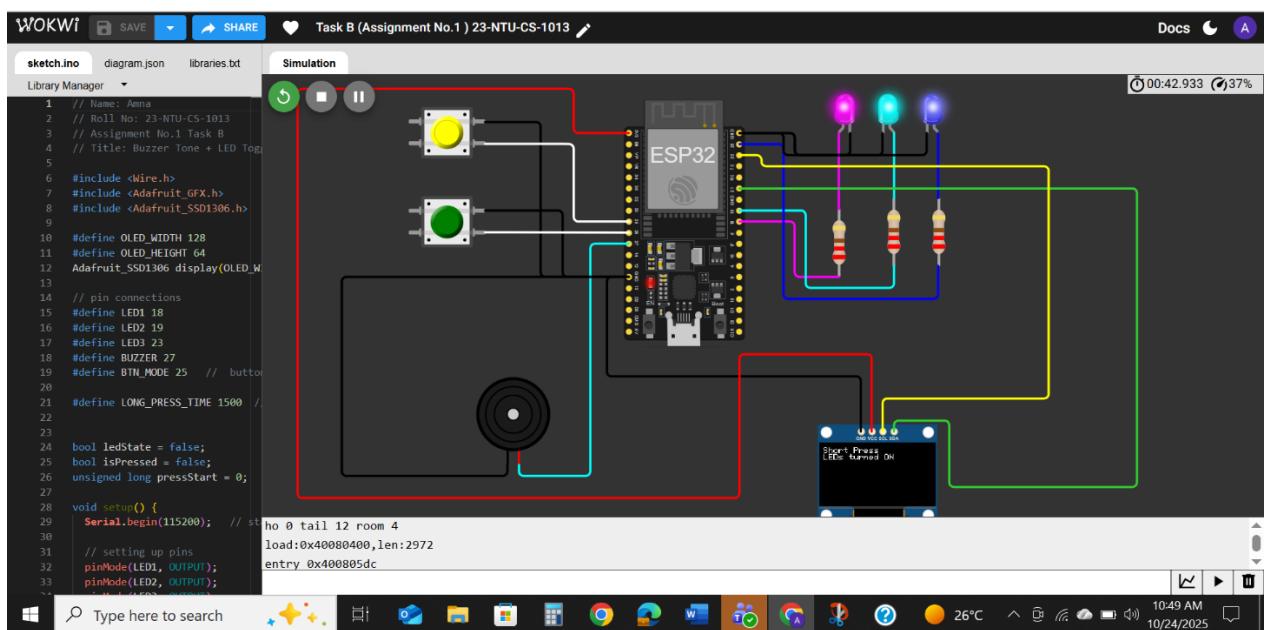
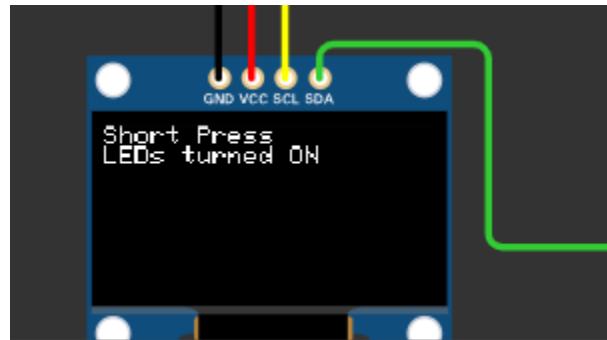
```

## 4.2. Output Screenshots

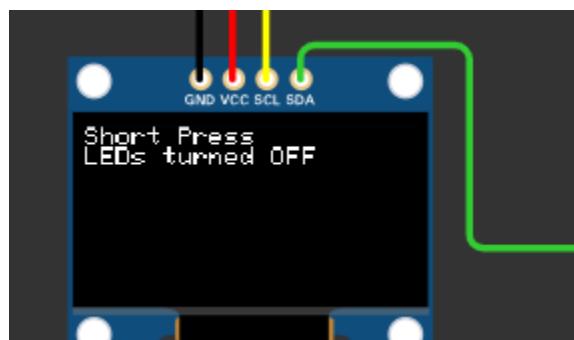
### a) System Ready

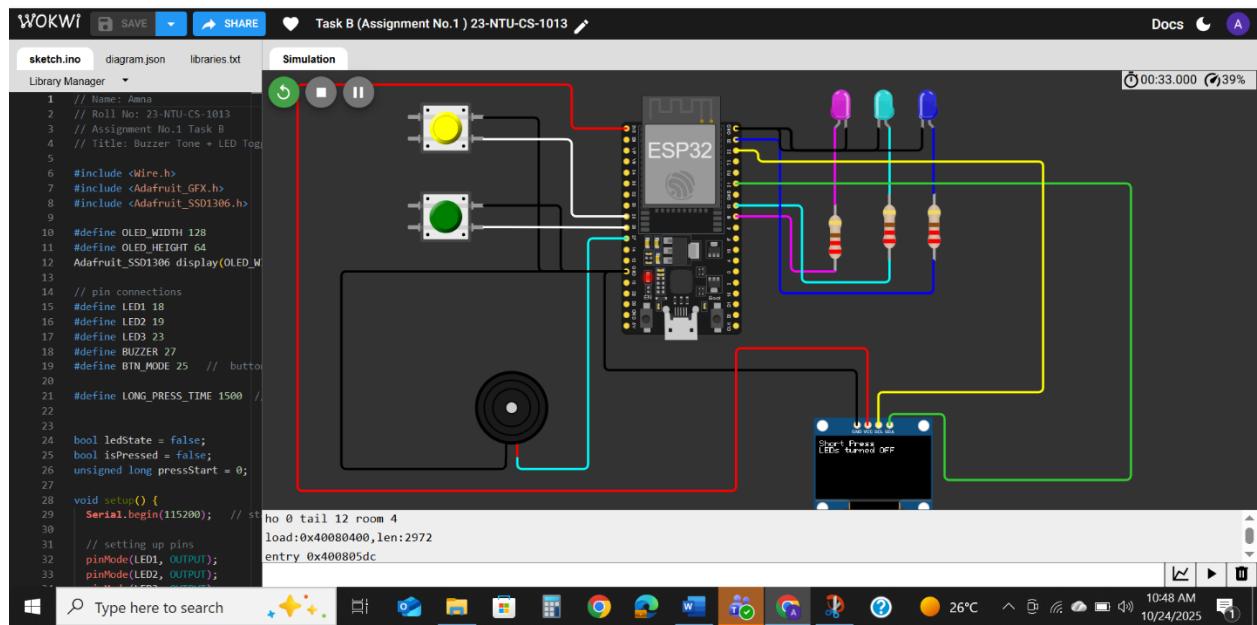


**b) Short press: LEDs Turned ON**



**c) Short press: LEDs Turned OFF**





#### d) Long press: Buzzer playing

