



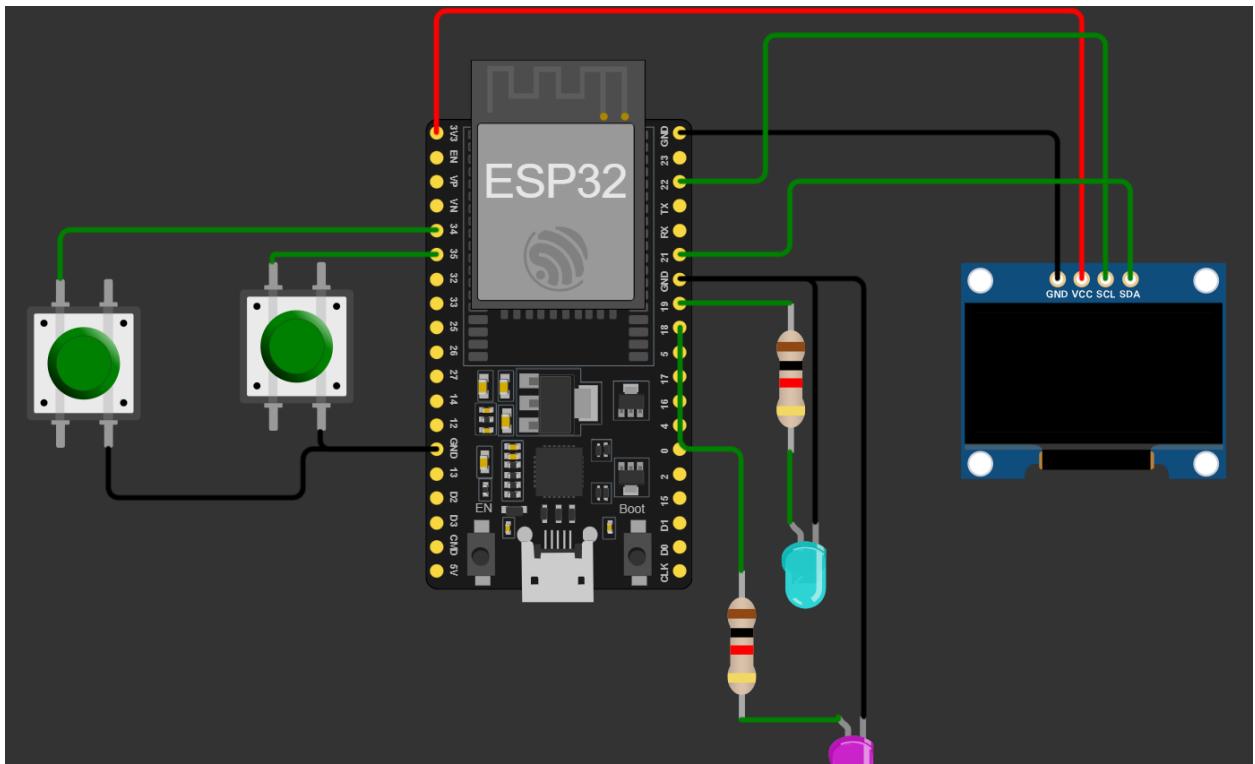
<b>Name:</b>	Muhammad Ibtisam Butt
<b>Reg No:</b>	23-NTU-CS-1269
<b>Section:</b>	BSAI <sup>4TH</sup>
<b>Subject</b>	Embedded IOT
<b>Submitted To:</b>	Sir. Nasir Mehmood

## Assignment-01

### Question 3

# Task A:

1. Wokwi: <https://wokwi.com/projects/445901865090999297>
2. Circuit Diagrams:



### 3. Code Snippets:

The screenshot shows the PlatformIO IDE interface with the main.cpp file open. The code is a C++ program for an Arduino project. It includes headers for Arduino.h, Wire.h, Adafruit\_GFX.h, and Adafruit\_SSD1306.h. It defines constants for SCREEN\_WIDTH (128), SCREEN\_HEIGHT (64), LED1 (pin 18), LED2 (pin 19), and two button pins (pin 35 for BTN\_MODE and pin 34 for BTN\_RESET). It also defines a pin for SDA (pin 21) and SCL (pin 22). The code includes a helper function showMode() which clears the display, sets text color to white, and prints the current mode. The setup() function initializes the display and calls showMode(). The loop() function checks the button states and updates the LED modes accordingly.

```
src > main.cpp > BTN_RESET
1  /*
2   * --- Task A: LED Mode Controller with Fade ---
3   * Name: Ibtisam Butt
4   * Reg. No: 23-NTU-CS-1269
5   * Description: Two push buttons control LED modes and show the state on OLED
6  */
7
8  #include <Arduino.h>
9  #include <Wire.h>
10 #include <Adafruit_GFX.h>
11 #include <Adafruit_SSD1306.h>
12
13 #define SCREEN_WIDTH 128
14 #define SCREEN_HEIGHT 64
15 Adafruit_SSD1306 display(SCREEN_WIDTH, SCREEN_HEIGHT, &Wire, -1);
16
17 // --- Pin Configuration ---
18 #define LED1 18
19 #define LED2 19
20 #define BTN_MODE 35
21 #define BTN_RESET 34
22 // SDA + 21, SCL + 22
23
24 int mode = 0;
25 unsigned long previousMillis = 0;
26 bool ledState = false;
27
28 // --- Helper Function to show text on OLED ---
29 void showMode(String text) {
30     display.clearDisplay();
31     display.setTextSize(1);
32     display.setTextColor(SSD1306_WHITE);
33     display.setCursor(10, 25);
34     display.print("Mode: ");
35     display.println(text);
36     display.display();
37 }
38
39 // --- Setup ---
40 void setup() {
41     display.begin();
42     display.setContrast(0);
43     display.display();
44     delay(1000);
45 }
46
47 void loop() {
48     if (millis() - previousMillis > 1000) {
49         previousMillis = millis();
50         if (mode == 0) {
51             mode++;
52         } else {
53             mode = 0;
54         }
55         ledState = !ledState;
56         display.clearDisplay();
57         display.setTextSize(1);
58         display.setTextColor(SSD1306_WHITE);
59         display.setCursor(10, 25);
60         display.print("Mode: ");
61         display.println(mode);
62         display.display();
63     }
64 }
```

Ln 21, Col 21 Spaces: 2 UTF-8 CRLF {} C++ Go Live PlatformIO

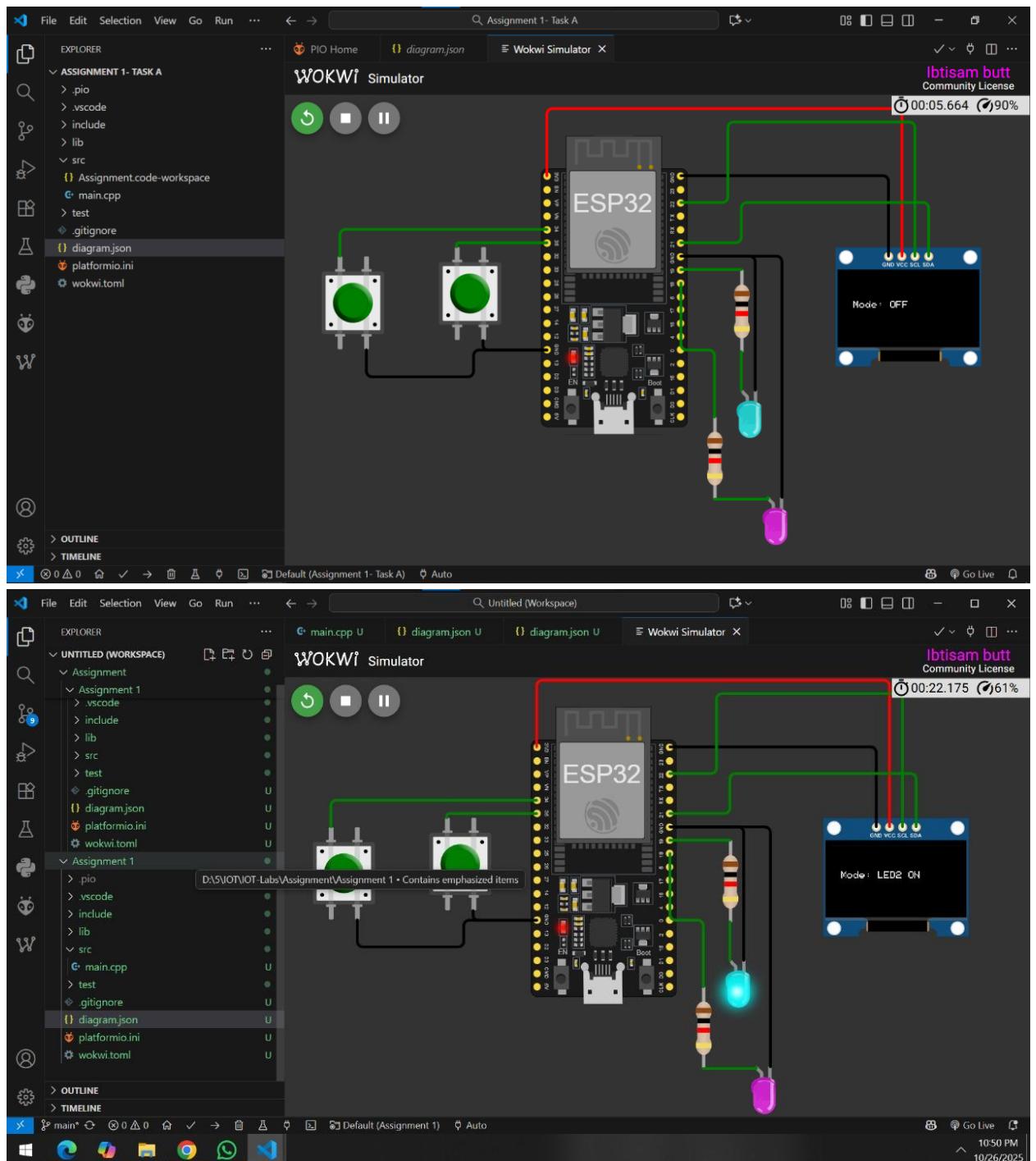
```
src > main.cpp > BTN_RESET
39 // --- Setup ---
40 void setup() {
41     pinMode(LED1, OUTPUT);
42     pinMode(LED2, OUTPUT);
43     pinMode(BTN_MODE, INPUT);
44     pinMode(BTN_RESET, INPUT);
45
46     if (!display.begin(SSD1306_SWITCHCAPVCC, 0x3C)) {
47         for (;;) // halt if OLEO not found
48     }
49
50     display.clearDisplay();
51     showMode("OFF");
52 }
53
54 // --- Main Loop ---
55 void loop() {
56     // --- Button Handling ---
57     if (digitalRead(BTN_MODE) == HIGH) {
58         mode = (mode + 1) % 4; // 0-3 modes
59         delay(300);           // debounce
60     }
61
62     if (digitalRead(BTN_RESET) == HIGH) {
63         mode = 0;
64         delay(300);
65     }
66
67     // --- LED Modes ---
68     switch (mode) {
69         case 0: // Both OFF
70             digitalWrite(LED1, LOW);
71             digitalWrite(LED2, LOW);
72             showMode("OFF");
73             break;
74
75         case 1: // Alternate Blink
76             showMode("ALT BLINK");
77             if (millis() - previousMillis > 500) { // toggle every 0.5s
78                 previousMillis = millis();
79             }
80             ledState = !ledState;
81             digitalWrite(LED1, ledState ? HIGH : LOW);
82             digitalWrite(LED2, ledState ? LOW : HIGH);
83             break;
84
85         case 2: // Both ON
86             digitalWrite(LED1, HIGH);
87             digitalWrite(LED2, HIGH);
88             showMode("BOTH ON");
89             break;
90
91         case 3: // PWM Fade
92             showMode("FADE");
93             for (int brightness = 0; brightness <= 255; brightness += 5) {
94                 analogWrite(LED1, brightness);
95                 analogWrite(LED2, brightness);
96                 delay(20);
97             }
98             for (int brightness = 255; brightness >= 0; brightness -= 5) {
99                 analogWrite(LED1, brightness);
100                analogWrite(LED2, brightness);
101                delay(20);
102            }
103        }
104    }
105 }
```

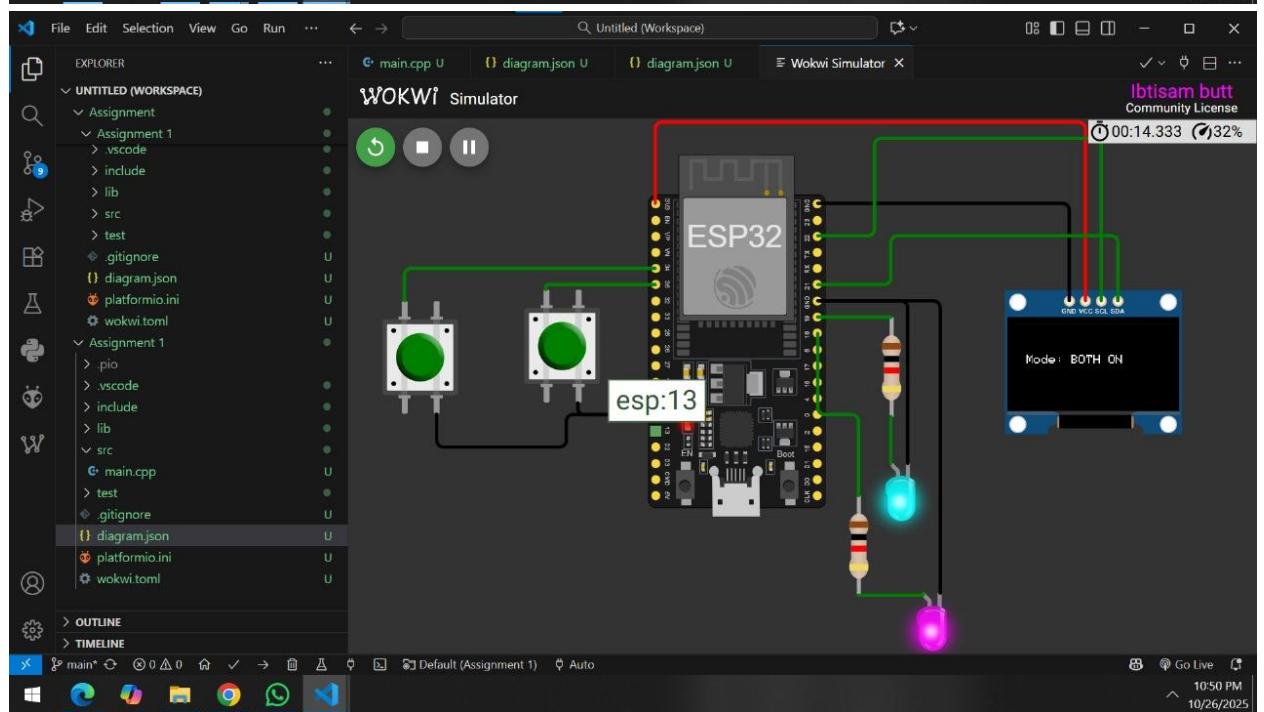
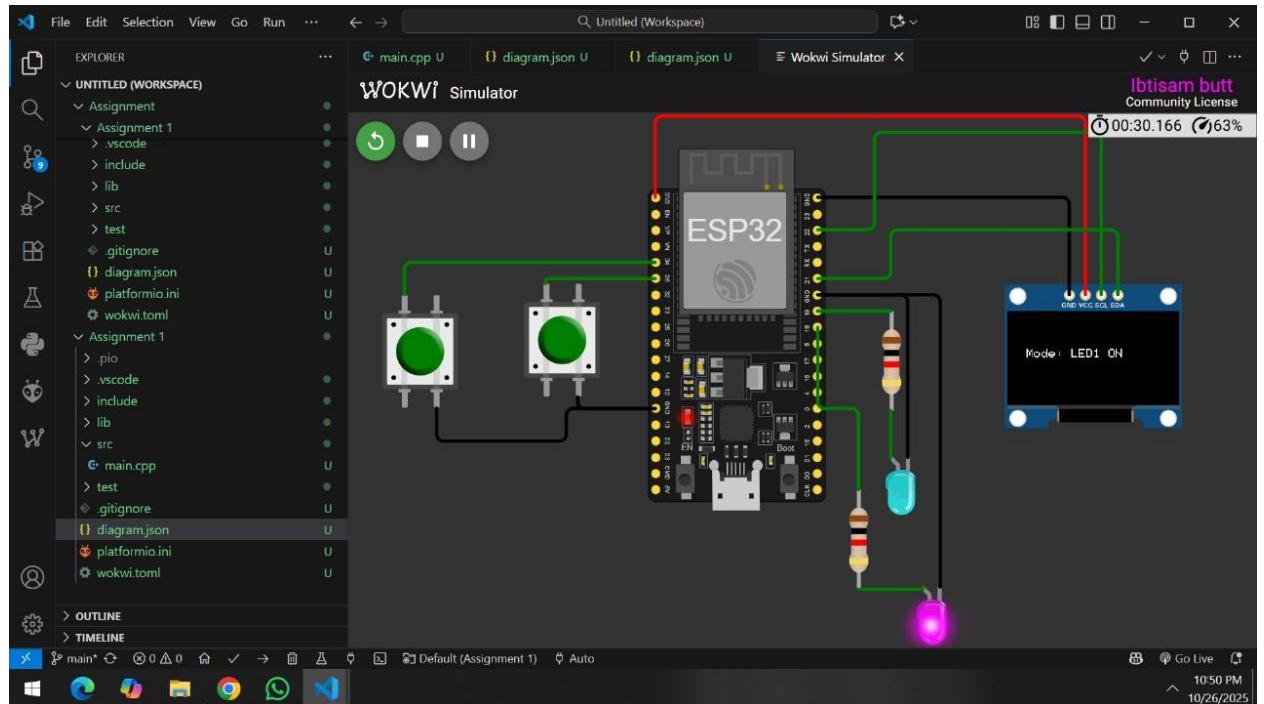
```
src > main.cpp > BTN_RESET
55 void loop() {
56     switch (mode) {
57         case 1: // Alternate Blink
58             showMode("ALT BLINK");
59             if (millis() - previousMillis > 500) { // toggle every 0.5s
60                 previousMillis = millis();
61             }
62             ledState = !ledState;
63             digitalWrite(LED1, ledState ? HIGH : LOW);
64             digitalWrite(LED2, ledState ? LOW : HIGH);
65             break;
66
67         case 2: // Both ON
68             digitalWrite(LED1, HIGH);
69             digitalWrite(LED2, HIGH);
70             showMode("BOTH ON");
71             break;
72
73         case 3: // PWM Fade
74             showMode("FADE");
75             for (int brightness = 0; brightness <= 255; brightness += 5) {
76                 analogWrite(LED1, brightness);
77                 analogWrite(LED2, brightness);
78                 delay(20);
79             }
80             for (int brightness = 255; brightness >= 0; brightness -= 5) {
81                 analogWrite(LED1, brightness);
82                 analogWrite(LED2, brightness);
83                 delay(20);
84             }
85         }
86     }
87 }
```

## 4. Pin Map:

```
## Pin Map
- LED1 → GPIO 18
- LED2 → GPIO 19
- Button1 → GPIO 35
- Button2 → GPIO 34
- OLED → SDA 21, SCL 22
```

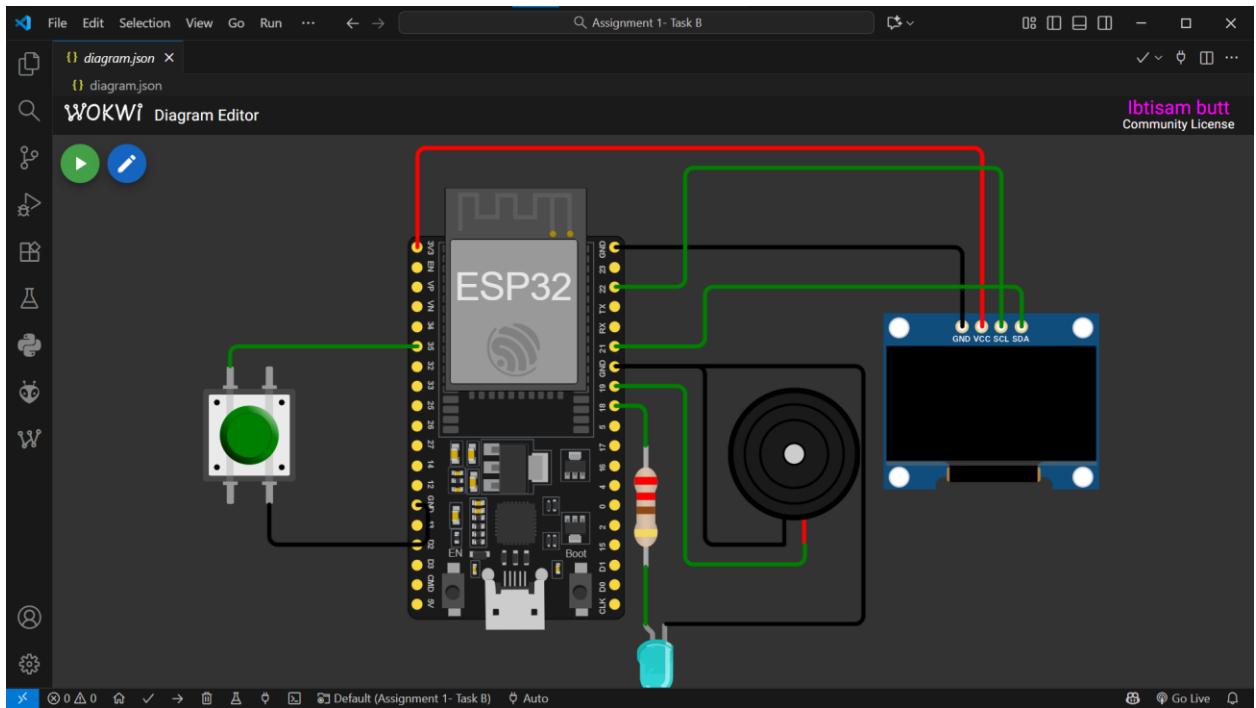
## 5. Screenshots:



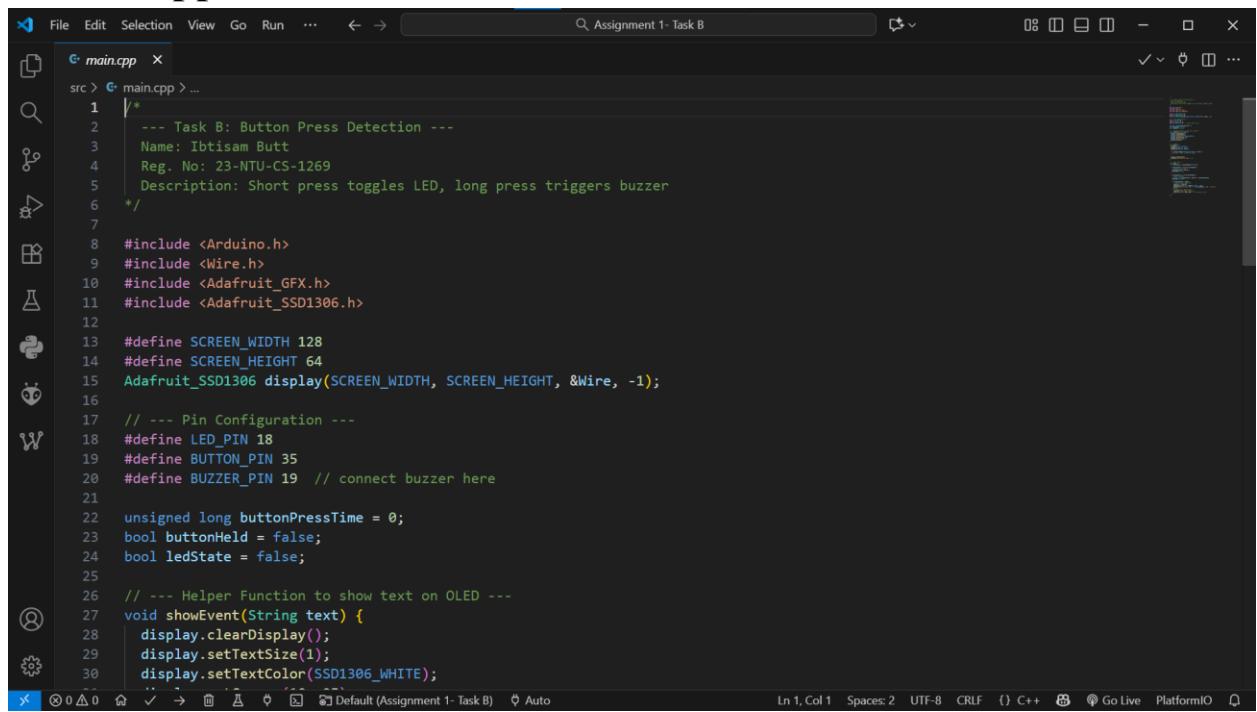


# Task B:

1. Wokwi: <https://wokwi.com/projects/445898275769959425>
2. Circuit Diagrams:



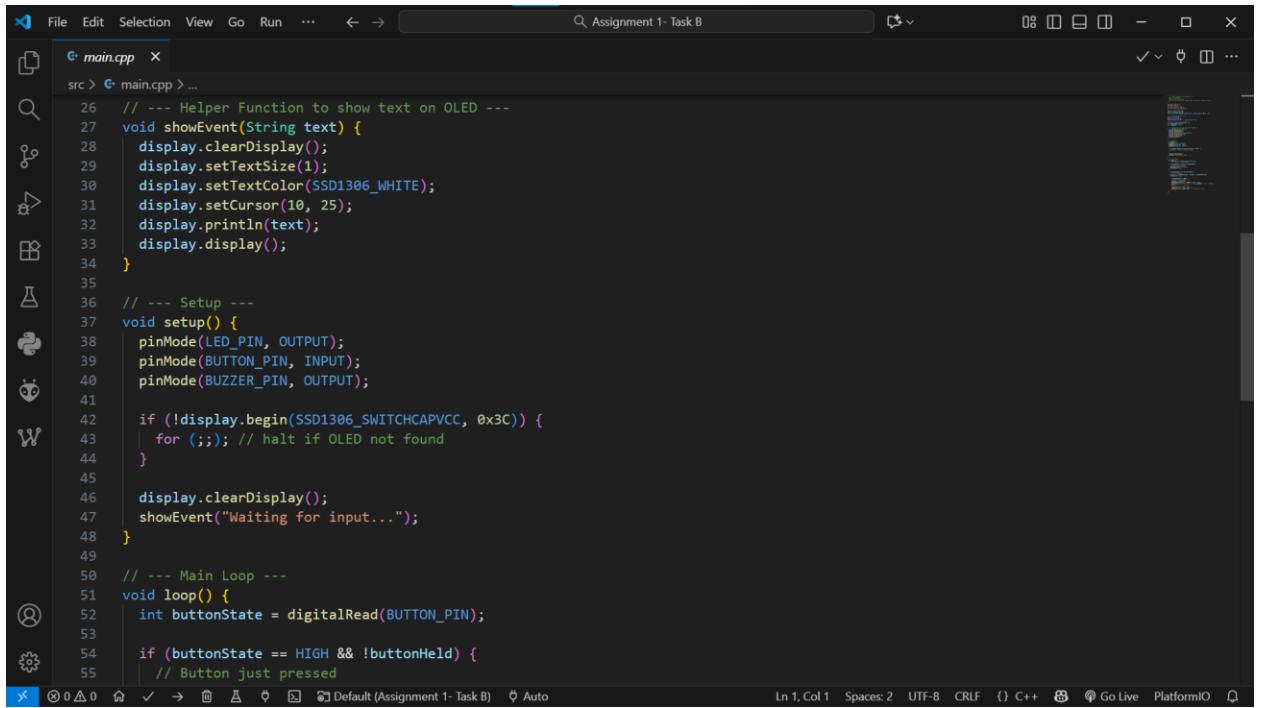
### 3. Code Snippets:



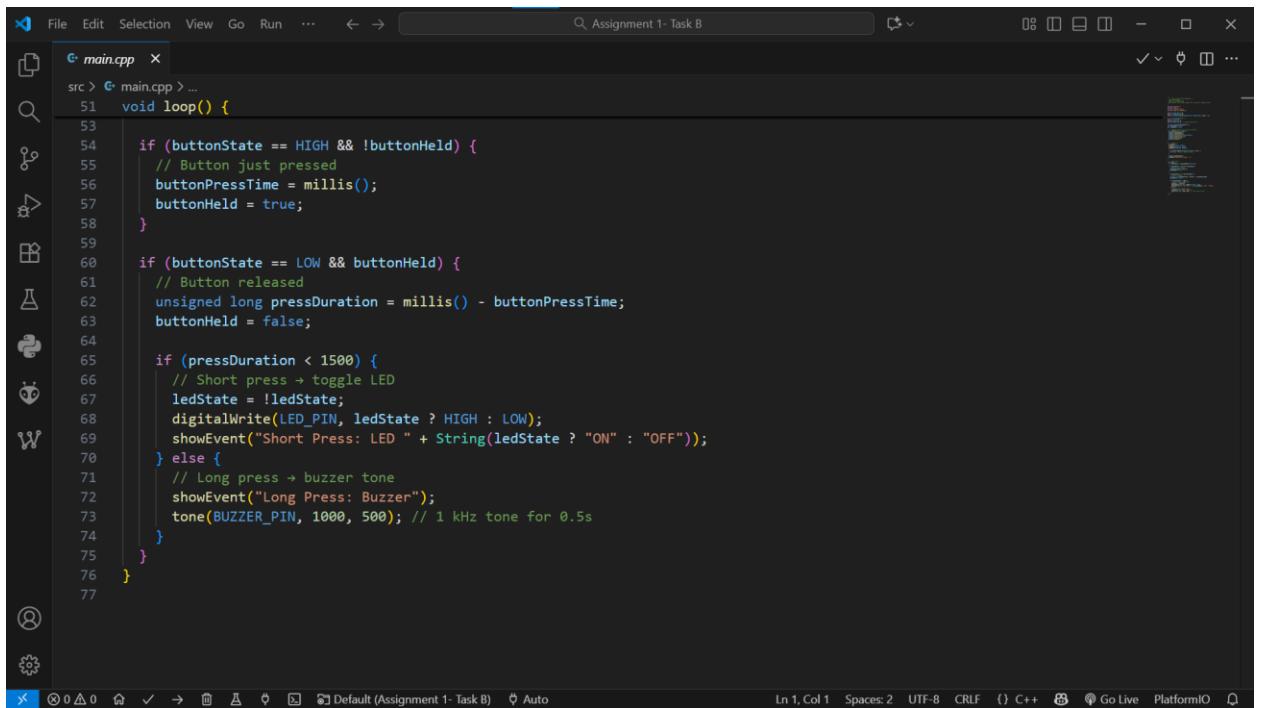
The screenshot shows a code editor window titled "Assignment 1- Task B". The file being edited is "main.cpp". The code is as follows:

```
src > main.cpp > ...
1 /*
2  --- Task B: Button Press Detection ---
3  Name: IbtiSam Butt
4  Reg. No: 23-NTU-CS-1269
5  Description: Short press toggles LED, long press triggers buzzer
6 */
7
8 #include <Arduino.h>
9 #include <Wire.h>
10 #include <Adafruit_GFX.h>
11 #include <Adafruit_SSD1306.h>
12
13 #define SCREEN_WIDTH 128
14 #define SCREEN_HEIGHT 64
15 Adafruit_SSD1306 display(SCREEN_WIDTH, SCREEN_HEIGHT, &Wire, -1);
16
17 // --- Pin Configuration ---
18 #define LED_PIN 18
19 #define BUTTON_PIN 35
20 #define BUZZER_PIN 19 // connect buzzer here
21
22 unsigned long buttonPressTime = 0;
23 bool buttonHeld = false;
24 bool ledState = false;
25
26 // --- Helper Function to show text on OLED ---
27 void showEvent(String text) {
28     display.clearDisplay();
29     display.setTextSize(1);
30     display.setTextColor(SSD1306_WHITE);
31 }
```

The code is a C++ program for an Arduino. It includes headers for Arduino, Wire, Adafruit\_GFX, and Adafruit\_SSD1306. It defines constants for screen dimensions and pin numbers (LED\_PIN 18, BUTTON\_PIN 35, BUZZER\_PIN 19). It also defines variables for button press time, button held status, and led state. A helper function "showEvent" is defined to clear the display, set text size to 1, and set text color to white. The code is annotated with comments explaining the task: "Task B: Button Press Detection", "Name: IbtiSam Butt", "Reg. No: 23-NTU-CS-1269", and "Description: Short press toggles LED, long press triggers buzzer".



```
src > main.cpp > ...
26 // --- Helper Function to show text on OLED ---
27 void showEvent(String text) {
28     display.clearDisplay();
29     display.setTextSize(1);
30     display.setTextColor(SSD1306_WHITE);
31     display.setCursor(10, 25);
32     display.println(text);
33     display.display();
34 }
35
36 // --- Setup ---
37 void setup() {
38     pinMode(LED_PIN, OUTPUT);
39     pinMode(BUTTON_PIN, INPUT);
40     pinMode(BUZZER_PIN, OUTPUT);
41
42     if (!display.begin(SSD1306_SWITCHCAPVCC, 0x3C)) {
43         for (;;); // halt if OLED not found
44     }
45
46     display.clearDisplay();
47     showEvent("Waiting for input...");
48 }
49
50 // --- Main Loop ---
51 void loop() {
52     int buttonState = digitalRead(BUTTON_PIN);
53
54     if (buttonState == HIGH && !buttonHeld) {
55         // Button just pressed
```



```
src > main.cpp > ...
51 void loop() {
52
53     if (buttonState == HIGH && !buttonHeld) {
54         // Button just pressed
55         buttonPressTime = millis();
56         buttonHeld = true;
57     }
58
59     if (buttonState == LOW && buttonHeld) {
60         // Button released
61         unsigned long pressDuration = millis() - buttonPressTime;
62         buttonHeld = false;
63
64         if (pressDuration < 1500) {
65             // Short press → toggle LED
66             ledState = !ledState;
67             digitalWrite(LED_PIN, ledState ? HIGH : LOW);
68             showEvent("Short Press: LED " + String(ledState ? "ON" : "OFF"));
69         } else {
70             // Long press → buzzer tone
71             showEvent("Long Press: Buzzer");
72             tone(BUZZER_PIN, 1000, 500); // 1 kHz tone for 0.5s
73         }
74     }
75 }
76 }
```

## 4. Pin Map: ## Pin Map

- LED1 → GPIO 18
- Button → GPIO 35
- Buzzer → GPIO 19
- OLED → SDA 21, SCL 22

## 5. Screenshots:

