

Assignment 1:

Name: Laiba Fatima

Registration Number: 23-ntu-cs-1257

Course: Embedded IOT Systems

Submission Date: 25-10-2025

Objective:

The objective of this assignment is to design and implement an ESP32-based circuit with LEDs, push buttons, a buzzer, and an OLED display.

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

Components List:

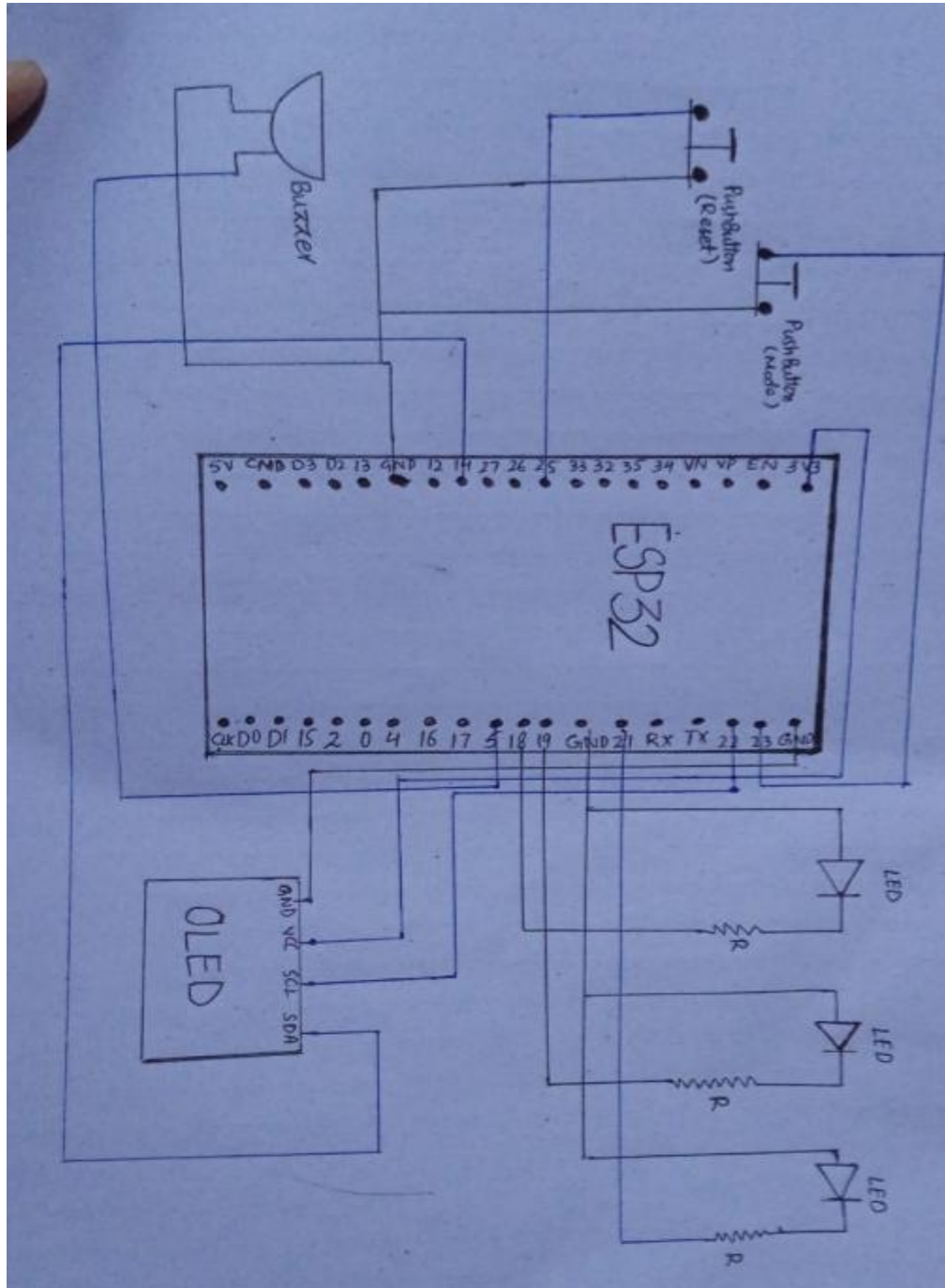
Component	Quantity	Notes
ESP32	1	Microcontroller
LEDs	3	Any color
Push Buttons	2	For mode change and reset

Component	Quantity	Notes
Buzzer	1	Active buzzer
OLED Display	1	I2C 128x64
Resistors	3	For LEDs (100Ω)

PIN Mapping:

Component	ESP32 Pin
LED1	18
LED2	19
LED3	21
Button1	23
Button2	25
Buzzer	5
OLED SDA	14
OLED SCL	22
OLED VCC	3V3
OLED GND	GND

Circuit Diagram:



CODE:

// Name: Laiba Fatima

```
// Reg# 23-ntu-cs-1257
```

```
// Question3_TaskA(Assignment1)
```

```
// Subject: Embedded IOT Systems
```

```
#include <Arduino.h>      // Basic Arduino functions
```

```
#include <Wire.h>         // For I2C communication with OLED
```

```
#include <Adafruit_GFX.h>  // Graphics library for OLED
```

```
#include <Adafruit_SSD1306.h> // OLED driver library
```

```
// OLED setup
```

```
#define SCREEN_WIDTH 128  // OLED width in pixels
```

```
#define SCREEN_HEIGHT 64  // OLED height in pixels
```

```
Adafruit_SSD1306 display(SCREEN_WIDTH, SCREEN_HEIGHT, &Wire, -1); // Create OLED object
```

```
// Pin mapping
```

```
#define OLED_SDA 14      // OLED SDA pin
```

```
#define OLED_SCL 22      // OLED SCL pin
```

```
#define LED1 18           // LED1 pin
```

```
#define LED2 19           // LED2 pin
```

```
#define LED3 21           // LED3 pin
```

```
#define MODE_BUTTON 23    // Button to change mode
```

```
#define RESET_BUTTON 25   // Button to reset mode
```

```
#define BUZZER 5          // Buzzer pin
```

```
// Variables
```

```
int mode = 0;          // 0=OFF, 1=Alternate Blink, 2=Both ON, 3=PWM Fade
int lastModeBtn = HIGH; // Stores previous state of MODE button
int lastResetBtn = HIGH; // Stores previous state of RESET button
```

```
// Function to show current mode on OLED
```

```
void showMode() {
    display.clearDisplay(); // Clear previous text
    display.setTextSize(1); // Set text size
    display.setTextColor(SSD1306_WHITE); // White color text
    display.setCursor(10, 10); // Set cursor position
    display.print("Mode "); // Print "Mode "
    display.print(mode); // Print current mode number
    display.setCursor(10, 30); // Move to next line
```

```
// Show mode description
```

```
switch (mode) {
    case 0: display.print("Both OFF"); break;
    case 1: display.print("Alternate Blink"); break;
    case 2: display.print("Both ON"); break;
    case 3: display.print("PWM Fade"); break;
}
```

```
display.display(); // Update OLED screen
```

```
}
```

```
void setup() {
```

```
    // Set pins mode
```

```
pinMode(LED1, OUTPUT);
pinMode(LED2, OUTPUT);
pinMode(LED3, OUTPUT);
pinMode(MODE_BUTTON, INPUT_PULLUP); // Button with internal pull-up
pinMode(RESET_BUTTON, INPUT_PULLUP);
pinMode(BUZZER, OUTPUT);
```

```
// Initialize OLED
```

```
Wire.begin(OLED_SDA, OLED_SCL); // Start I2C for OLED
```

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

```
    Serial.println("SSD1306 allocation failed"); // Stop if OLED not working
```

```
    for(;;);
```

```
}
```

```
showMode(); // Show initial mode on OLED
```

```
}
```

```
void loop() {
```

```
    int modeBtn = digitalRead(MODE_BUTTON); // Read MODE button
```

```
    int resetBtn = digitalRead(RESET_BUTTON); // Read RESET button
```

```
// MODE button pressed (short press)
```

```
if (modeBtn == LOW && lastModeBtn == HIGH) {
```

```
    mode++;          // Go to next mode
```

```
    if (mode > 3) mode = 0; // Wrap around after mode 3
```

```
    tone(BUZZER, 1000, 100); // Beep buzzer shortly
```

```

    showMode();          // Update OLED
    delay(200);          // Debounce delay
}

lastModeBtn = modeBtn;   // Save current button state


// RESET button pressed
if (resetBtn == LOW && lastResetBtn == HIGH) {
    mode = 0;            // Reset mode to 0
    tone(BUZZER, 500, 200); // Beep buzzer
    showMode();          // Update OLED
    delay(200);          // Debounce delay
}

lastResetBtn = resetBtn; // Save current button state


// Behaviour for each mode
switch (mode) {
    case 0: // All LEDs OFF
        digitalWrite(LED1, LOW);
        digitalWrite(LED2, LOW);
        digitalWrite(LED3, LOW);
        break;

    case 1: // Alternate blink
        digitalWrite(LED1, HIGH); // LED1 ON
        digitalWrite(LED2, LOW);  // LED2 OFF
        digitalWrite(LED3, LOW);  // LED3 OFF

```

```
delay(300);          // Wait 0.3 sec
digitalWrite(LED1, LOW); // LED1 OFF
digitalWrite(LED2, HIGH); // LED2 ON
digitalWrite(LED3, HIGH); // LED3 ON
delay(300);
break;
```

case 2: // Both ON

```
digitalWrite(LED1, HIGH);
digitalWrite(LED2, HIGH);
digitalWrite(LED3, HIGH);
break;
```

case 3: // PWM Fade for all LEDs

```
for (int i = 0; i <= 255; i++) { // Increase brightness
    analogWrite(LED1, i);
    analogWrite(LED2, i);
    analogWrite(LED3, i);
    delay(5);
}
for (int i = 255; i >= 0; i--) { // Decrease brightness
    analogWrite(LED1, i);
    analogWrite(LED2, i);
    analogWrite(LED3, i);
    delay(5);
}
```



```

break;

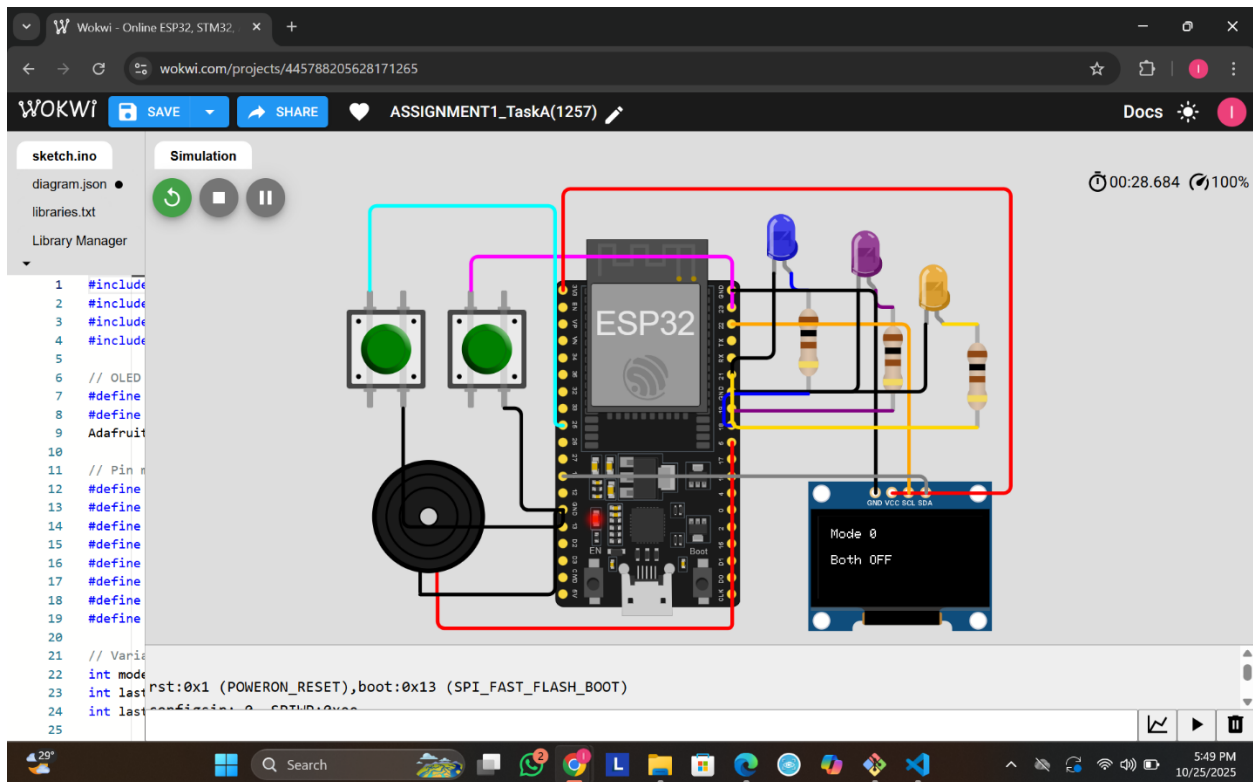
}

}

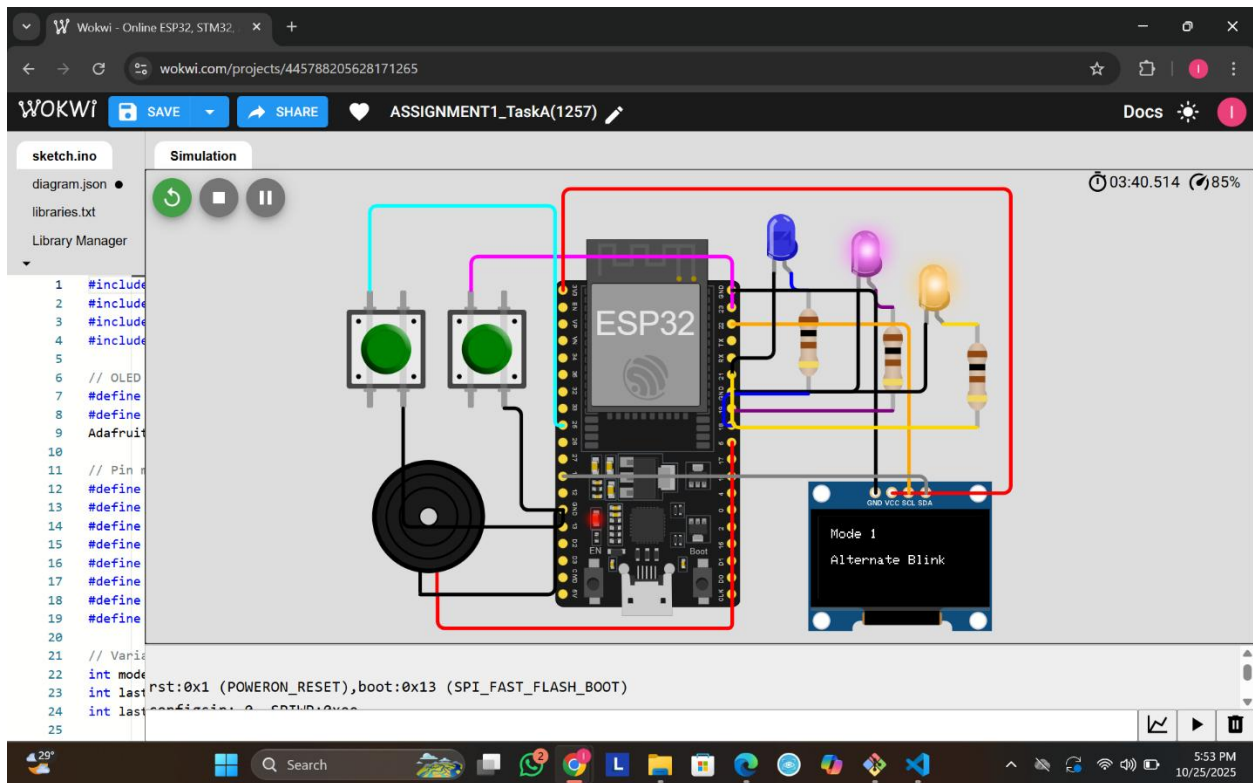
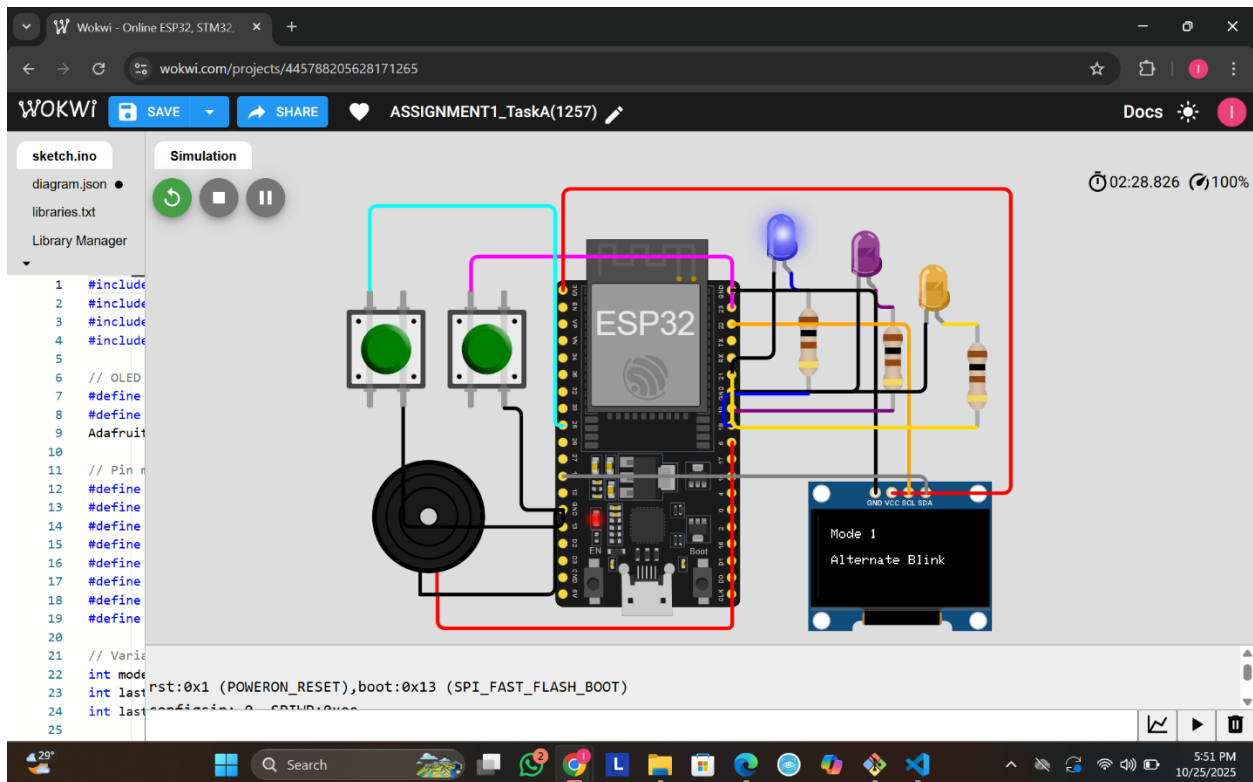
```

CODE OUTPUT:

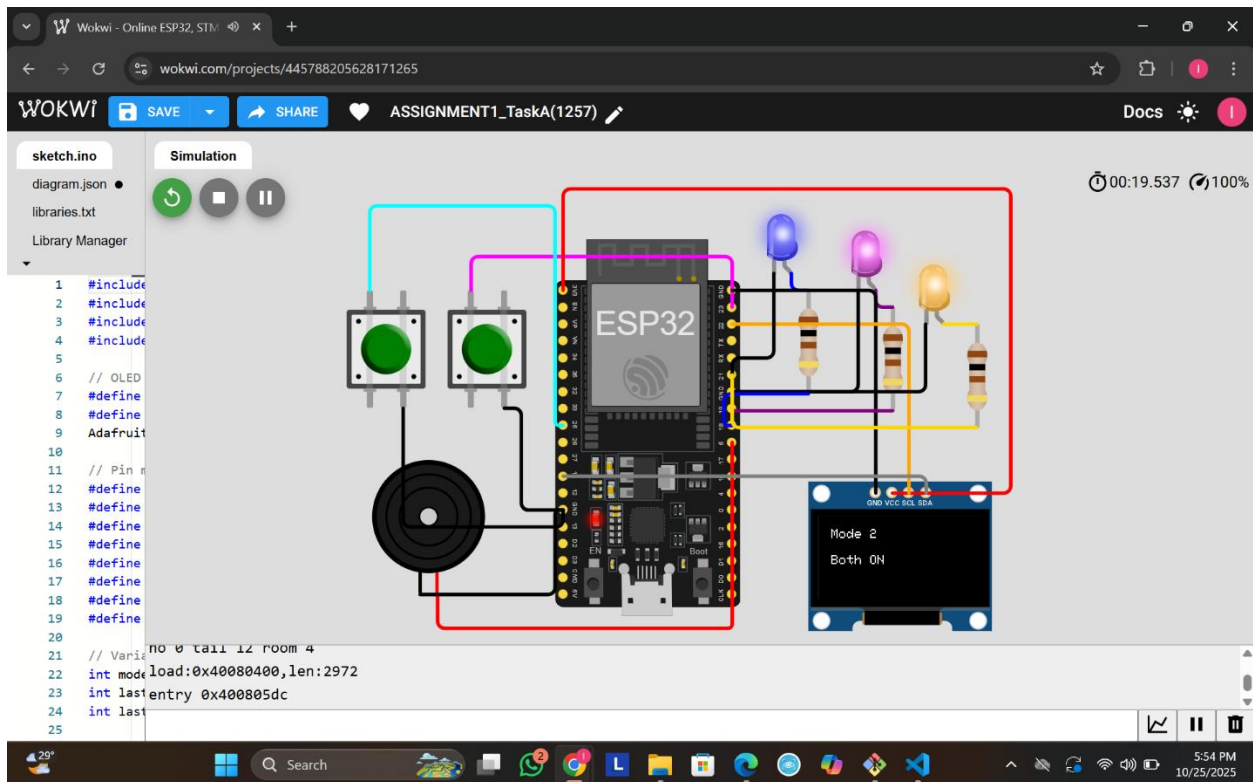
AT MODE0:



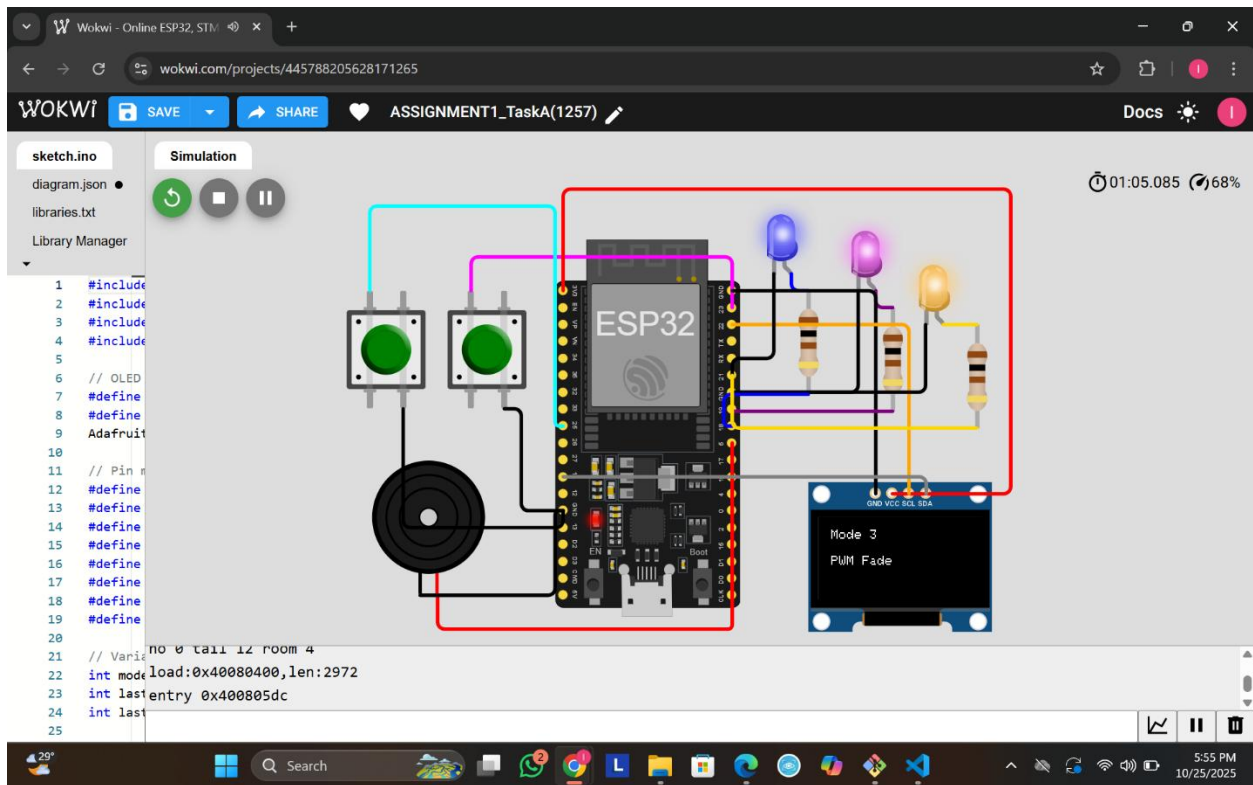
AT MODE1:

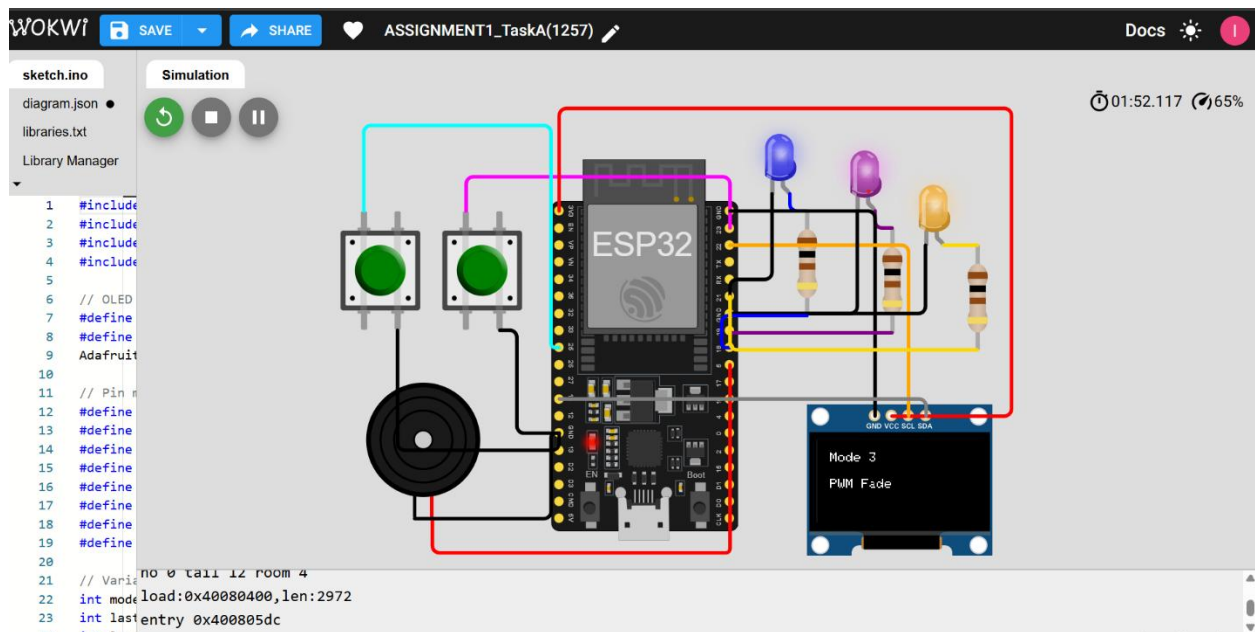


AT MODE2:



AT MODE3:





Code Overview

Structure of the Code:

1. Header / Libraries:

- Arduino.h → Basic Arduino functions
- Wire.h → For I2C communication with OLED
- Adafruit_GFX.h and Adafruit_SSD1306.h → To control and display text/graphics on OLED

2. Pin Definitions:

- LEDs → 3 pins (LED1, LED2, LED3)
- Buttons → 2 pins (Mode button & Reset button)
- Buzzer → 1 pin
- OLED → SDA & SCL pins

3. Variables:

- mode → stores current LED mode (0–3)
- lastModeBtn and lastResetBtn → track previous button state for detecting presses

4. Function showMode()

- Clears OLED
- Shows **current mode number**
- Displays **description** like “Both OFF” or “PWM Fade”

5. setup()

- Sets pin modes (INPUT/OUTPUT)
- Initializes OLED
- Shows the initial mode

6. loop()

- Reads buttons
- **Mode button pressed:** cycles through LED modes (0→3)
- **Reset button pressed:** resets mode to 0
- For each mode:
 - **0:** All LEDs OFF
 - **1:** Alternate blink
 - **2:** All LEDs ON
 - **3:** PWM fade

7. Delays and buzzer are used for debouncing and feedback

WORKING EXPLANATION:

1. There are **3 LEDs** and **2 buttons**.

2. **Button 1 (Mode Button):**

- Every time we press it, the LED behavior changes.
- There are **4 modes**:
 1. **All LEDs OFF** → nothing lights up
 2. **Alternate Blink** → LEDs blink one after another

3. **All LEDs ON** → all lights stay ON

4. **PWM Fade** → LEDs slowly brighten and dim like a fading effect

3. **Button 2 (Reset Button):**

- Resets the LED mode to **Mode 0 (All OFF)**

4. **OLED Display:**

- Shows which mode is currently active so we can **see it on screen**

5. **Buzzer:**

- Beeps shortly whenever a button is pressed to give **feedback**

WOKWI PROJECT LINK(TaskA):

<https://wokwi.com/projects/445788205628171265>

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

Components List:

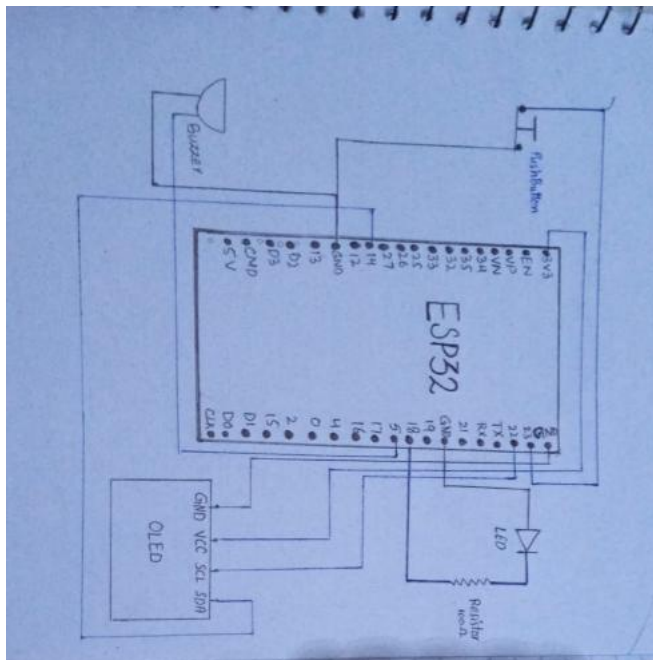
COMPONENTS	QUANTITY
Push Button	1
Resistor	1
LED	1
OLED	1

Buzzer	1
ESP32	1

Pin Mapping:

COMPONENTS Esp32	
Push Button	23
LED	18
OLED SDA	14
OLED SCL	22
OLED VCC	3V3
OLED GND	GND
Buzzer	5

Circuit Diagram:



Code:

```
// Name: Laiba Fatima
```

```
// Reg# 23-ntu-cs-1257
```

```
// Question3_TaskB(Assignment1)
```

// Subject: Embedded IOT Systems

```
#include <Arduino.h>      // Basic Arduino functions
```

```
#include <Wire.h>         // For communication with OLED
```

```
#include <Adafruit_GFX.h>  // Helps draw on OLED
```

```
#include <Adafruit_SSD1306.h> // OLED display driver
```

// OLED setup

```
#define SCREEN_WIDTH 128  // OLED width
```

```
#define SCREEN_HEIGHT 64  // OLED height
```

```
Adafruit_SSD1306 display(SCREEN_WIDTH, SCREEN_HEIGHT, &Wire, -1); // Create OLED object
```

// Pins

```
#define BUTTON 23         // Button pin
```

```
#define LED1 18           // LED pin
```

```
#define BUZZER 5          // Buzzer pin
```

// Variables

```
bool ledState = false;    // LED ON or OFF
```

```
unsigned long buttonPressTime = 0; // When button was pressed
```

```
bool buttonHeld = false;  // Is button being held?
```

```
const unsigned long LONG_PRESS_TIME = 1500; // 1.5 seconds for long press
```

```
void setup() {
```

```
    pinMode(BUTTON, INPUT_PULLUP); // Button input with pull-up resistor
```



```

pinMode(LED1, OUTPUT);    // LED output
pinMode(BUZZER, OUTPUT);  // Buzzer output

Wire.begin(14, 22);       // Start OLED I2C: SDA=14, SCL=22

// Initialize OLED
if(!display.begin(SSD1306_SWITCHCAPVCC, 0x3C)) {
    Serial.println("OLED failed"); // Stop if OLED not working
    for(;;);
}

// Show "Ready..." on OLED
display.clearDisplay();
display.setTextSize(1);
display.setTextColor(SSD1306_WHITE);
display.setCursor(10, 10);
display.println("Ready...");
display.display();
}

void loop() {
    int buttonState = digitalRead(BUTTON); // Read button state

    // Button pressed now
    if (buttonState == LOW && !buttonHeld) {
        buttonPressTime = millis(); // Remember time of press
    }
}

```

```
    buttonHeld = true;    // Mark button as held
}

// Button released now

if (buttonState == HIGH && buttonHeld) {
    unsigned long pressDuration = millis() - buttonPressTime; // How long button was
    pressed

    buttonHeld = false;    // Reset held flag

    display.clearDisplay(); // Clear OLED
    display.setCursor(10, 10); // Set text position

    if (pressDuration >= LONG_PRESS_TIME) {
        // Long press

        display.println("Long press detected"); // Show message
        display.display();

        tone(BUZZER, 1000, 500);    // Play sound for 0.5 sec
    } else {
        // Short press

        display.println("Short press detected"); // Show message
        display.display();

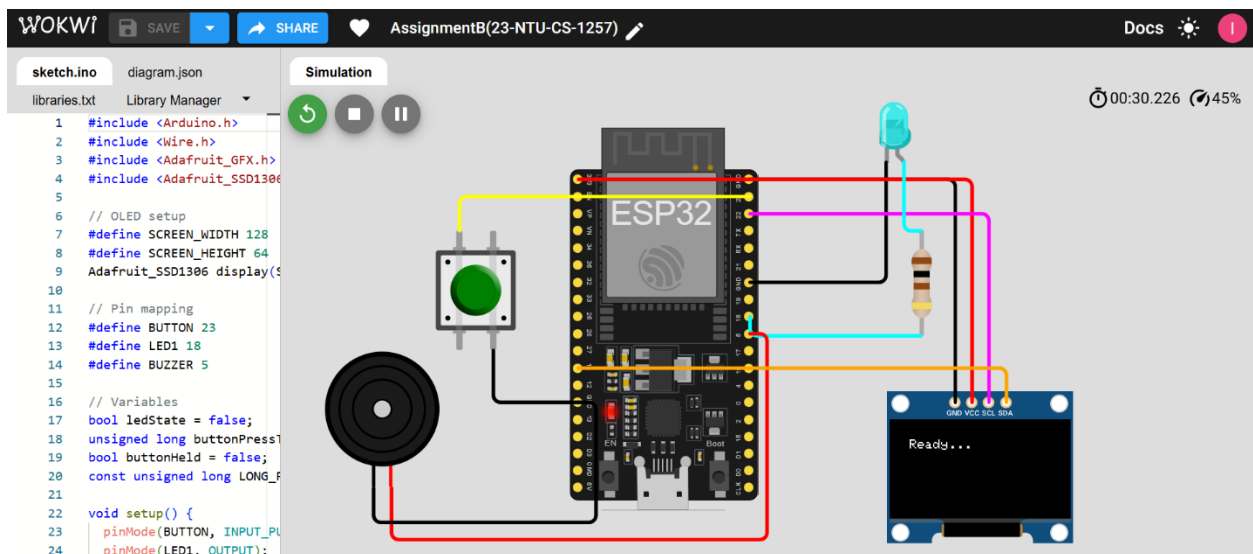
        ledState = !ledState;    // Switch LED ON/OFF
        digitalWrite(LED1, ledState);    // Apply new LED state
    }

    delay(200); // Small wait to avoid double detection
}
```

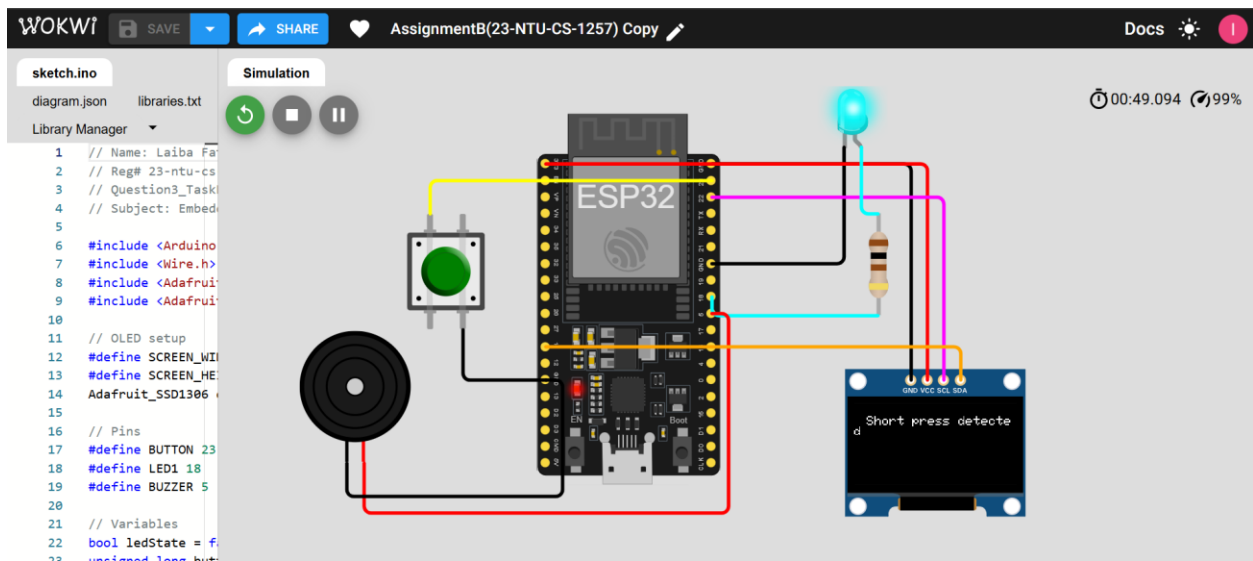
```
}  
  
}
```

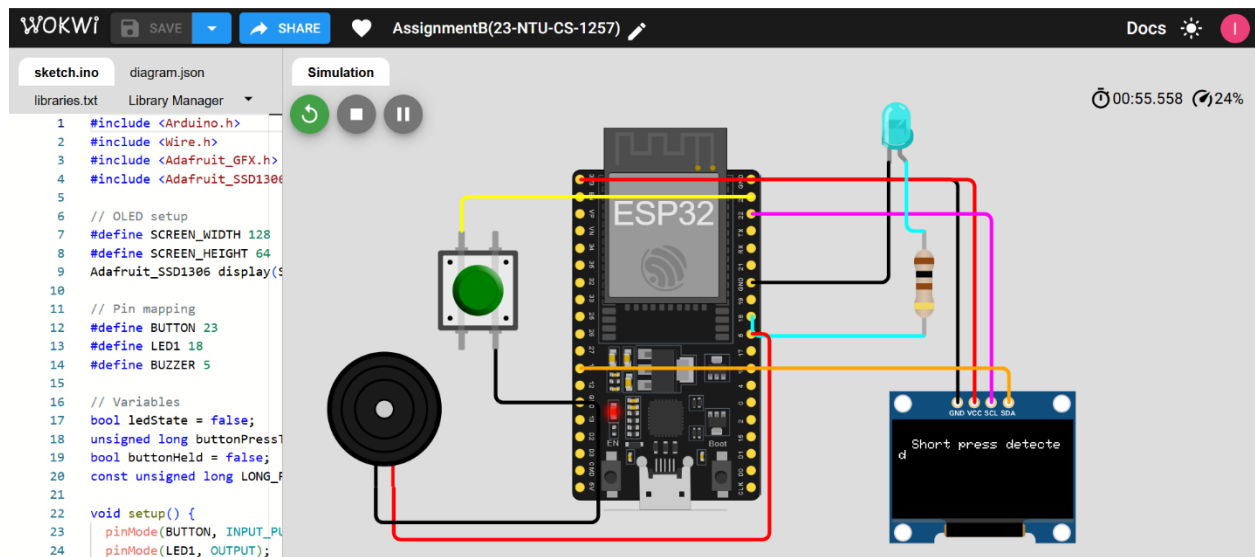
CODE OUTPUT:

ReadyToDetect:

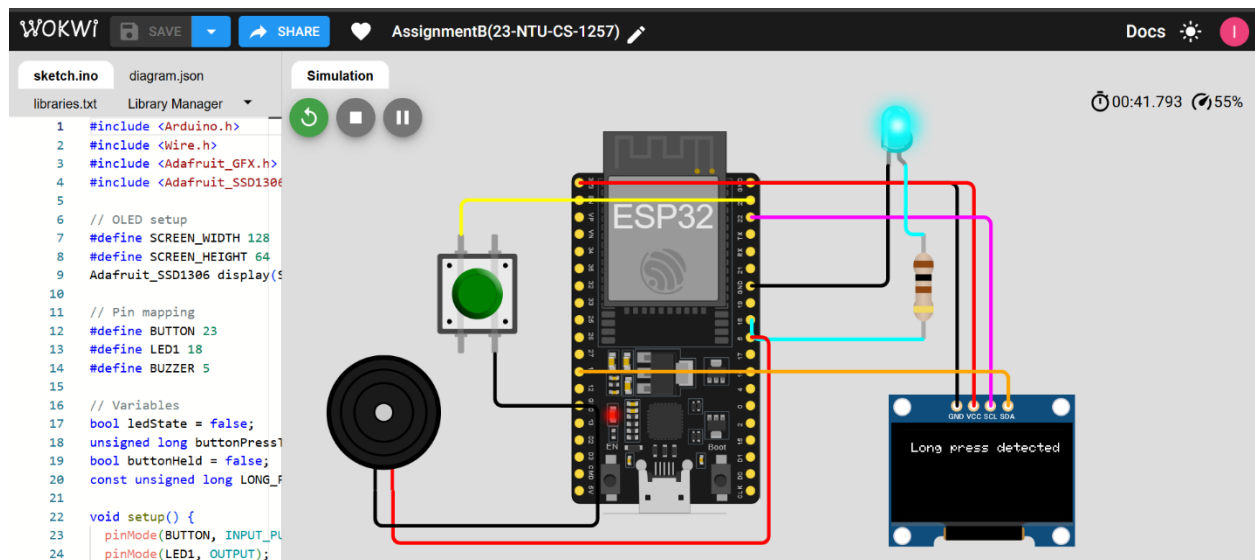


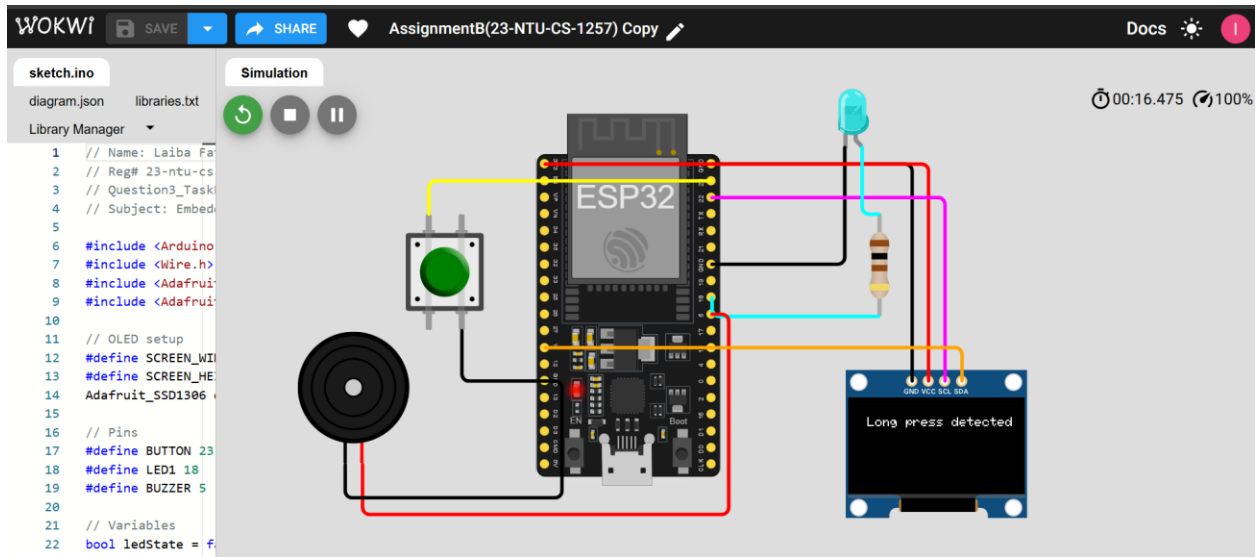
Short Press:





Long Press:





Code Overview:

Structure of the Code:

1. Header / Libraries:

- `Arduino.h` → Basic Arduino functions
- `Wire.h` → I2C communication with OLED
- `Adafruit_GFX.h` and `Adafruit_SSD1306.h` → For OLED display

2. Pin Definitions:

- **BUTTON** → 1 pin (GPIO 23)
- **LED1** → 1 pin (GPIO 18)
- **BUZZER** → 1 pin (GPIO 5)
- **OLED SDA/SCL** → 14/22

3. Variables:

- `ledState` → tracks if LED is ON or OFF
- `buttonPressTime` → stores time when button was pressed
- `buttonHeld` → checks if button is being held
- `LONG_PRESS_TIME` → 1.5 seconds threshold for long press

4. setup()

- Sets pins as INPUT/OUTPUT
- Initializes OLED
- Displays “Ready...” on OLED

5. loop()

- Reads the **button state**
- Detects **when button is pressed** and **released**
- Measures **press duration**
- Decides:
 - **Short press** → **toggle LED**
 - **Long press** → **play buzzer**
- Shows the action on OLED

WORKING EXPLANATION:

1. **There is 1 button, 1 LED, 1 buzzer, and an OLED display.**
2. **When we press the button:**
 - If we **press shortly (<1.5 sec)** → LED turns ON if it was OFF, or OFF if it was ON
 - OLED shows “**Short press detected**”
3. **When we press and hold the button (>1.5 sec):**
 - Buzzer beeps for half a second
 - OLED shows “**Long press detected**”
4. **Every time the button is pressed:**
 - The program measures **how long we held it**
 - Decides whether it's short or long press
 - Gives **visual feedback on OLED** and **sound feedback from buzzer**

WOKWI PROJECT LINK(TASK B):

<https://wokwi.com/projects/445794379500897281>

WOKWI PROJECT LINK(TASK A):

<https://wokwi.com/projects/445788205628171265>

Conclusion:

In this assignment, I used an ESP32 with 3 LEDs, 2 buttons, a buzzer, and an OLED screen.

For Task A, one button changes how the LEDs light up and the other button turns them off. The screen shows which mode is on.

For Task B, one button can do two things: a short press turns the LED on or off, and a long press makes the buzzer sound. The screen shows what happened.

This project helped me understand how buttons, LEDs, buzzers, and a screen can work together in a simple circuit. I also learned how to show information on the OLED.