National Textile University, Faisalabad

Department of Computer Science

Name:	**Zainab sultan**
Class:	BSCS_B 5th Semester
Registration No:	23-NTU-CS-1097
Assignment No:	01
Course Name:	Embedded & IOT system
Submitted To:	**Sir Nasir Mahmood**
Submission Date:	25/10/2025

Embedded IoT Systems (CSE-3080)

Assignment 1

Question 3:

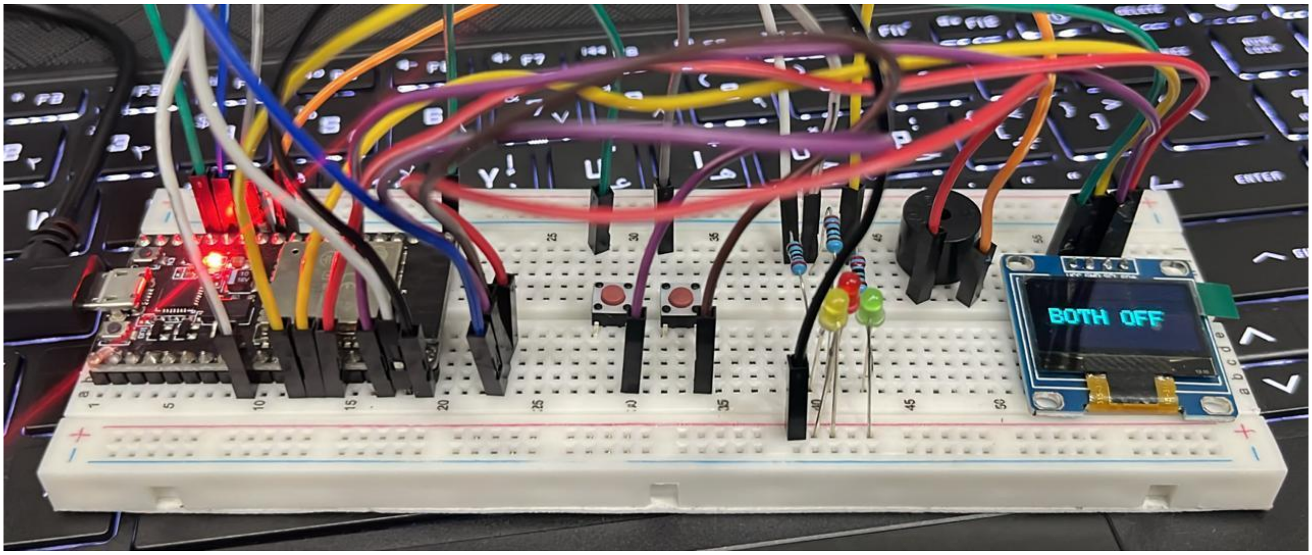
TASK A:

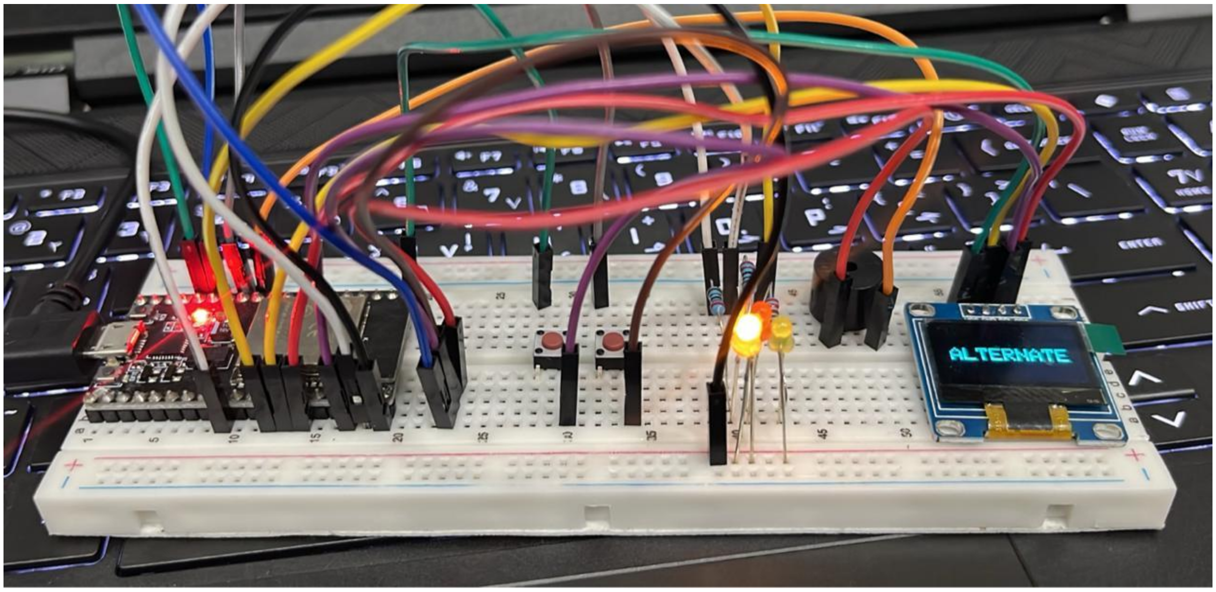
Coding: Use one button to cycle through LED modes (display the current state on the OLED):

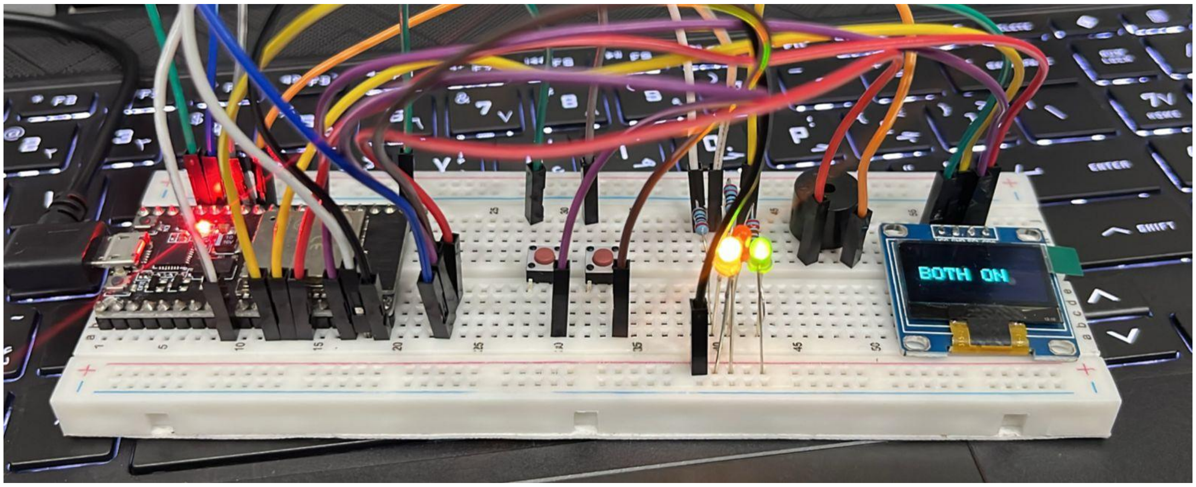
1. **Both OFF**
2. **Alternate blink**
3. **Both ON**
4. **PWM fade**

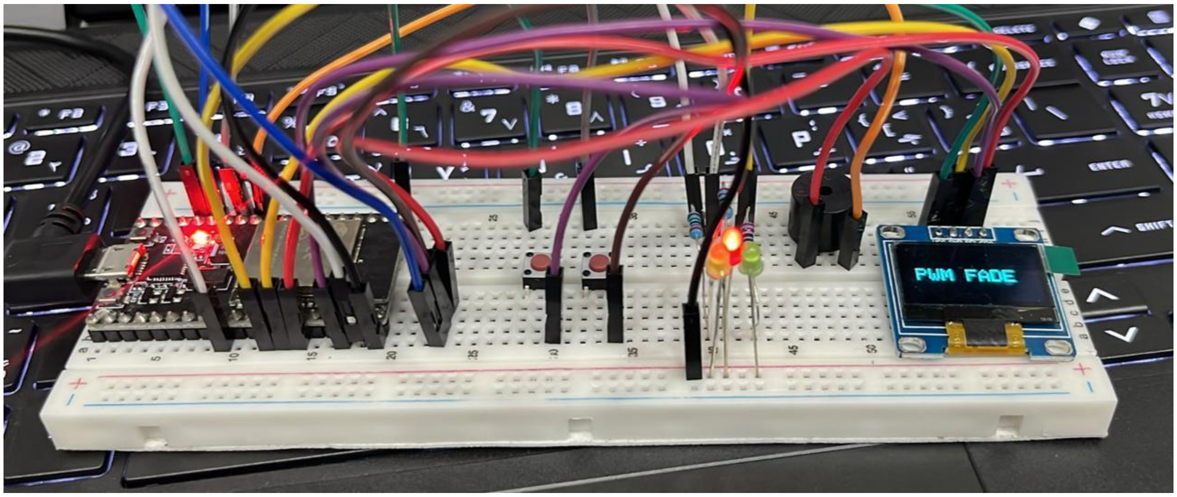
Use the second button to reset to OFF.

- **KIT:**

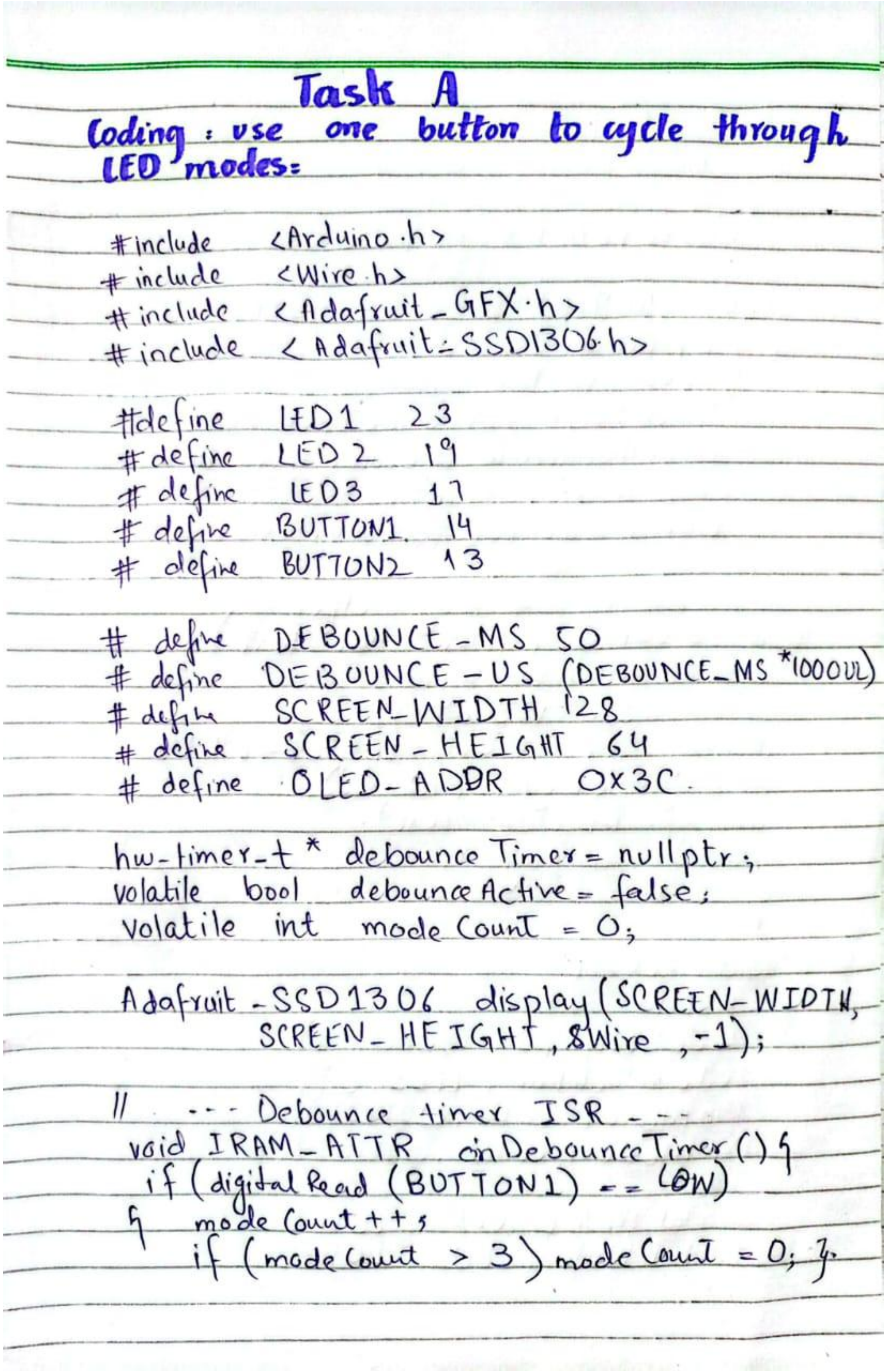



ALTERNATE


BOTH ON


PWM FADE

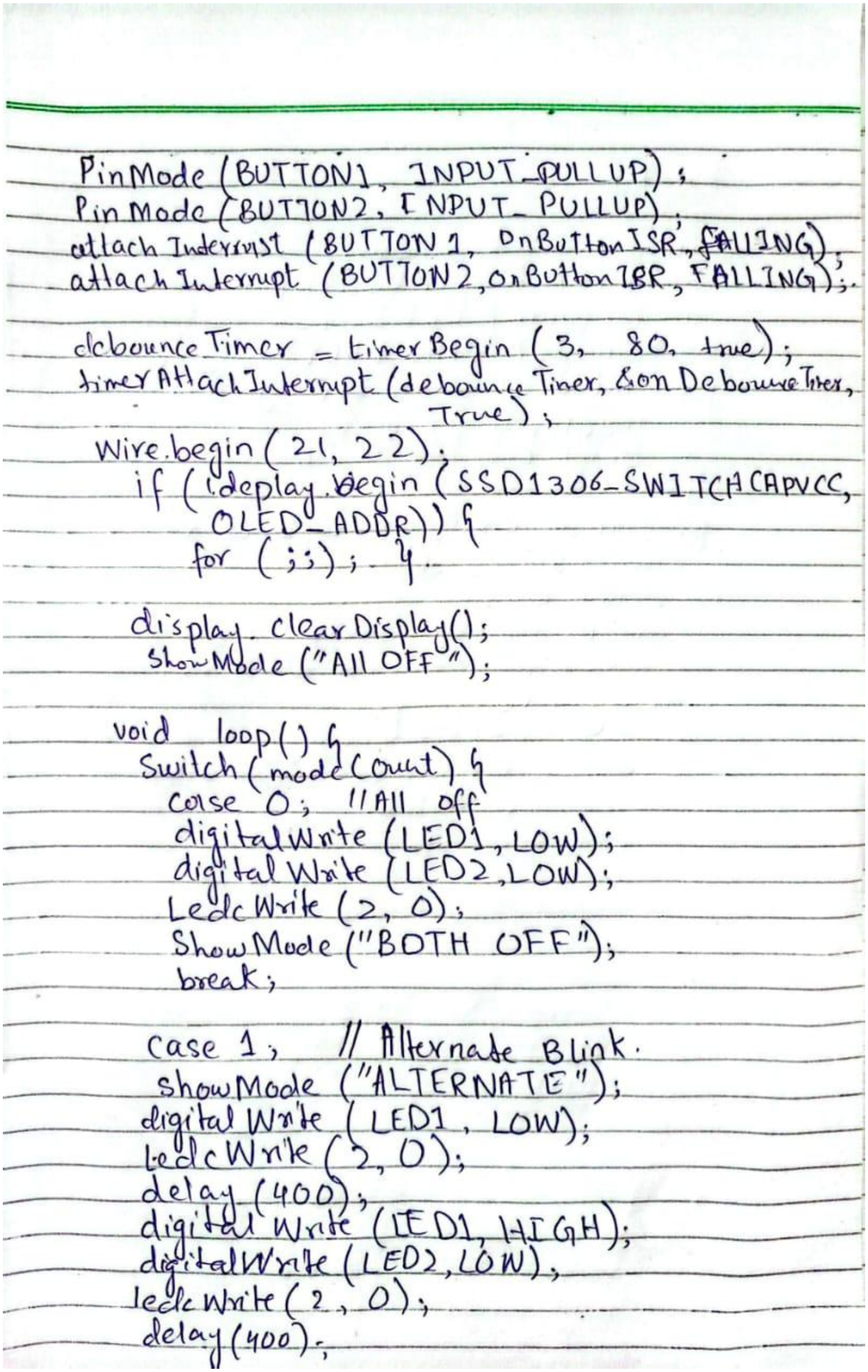
- **Handwritten code:**



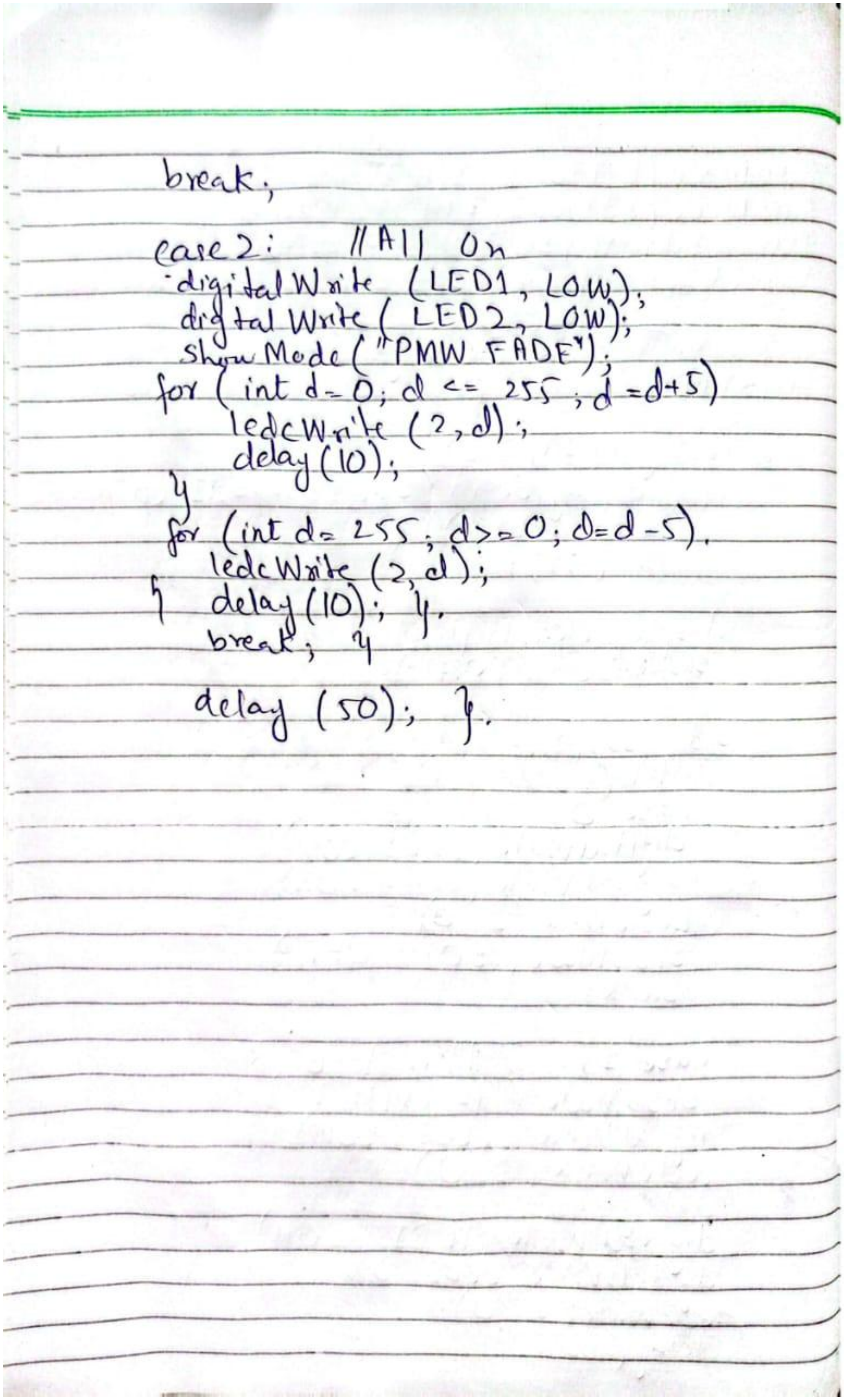
Task A

Coding : use one button to cycle through LED modes:

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

#define LED1 23
#define LED2 19
#define LED3 17
#define BUTTON1 14
#define BUTTON2 13

#define DEBOUNCE-MS 50
#define DEBOUNCE-US (DEBOUNCE_MS *1000UL)
#define SCREEN_WIDTH 128
#define SCREEN-HEIGHT 64
#define OLED-ADDR 0x3C.

hw-timer-t * debounceTimer = nullptr;
volatile bool debounceActive = false;
volatile int modeCount = 0;

Adafruit-SSD1306 display(SCREEN-WIDTH,
         SCREEN_HEIGHT, &Wire , -1);

// --- Debounce timer ISR ---
void IRAM_ATTR onDebounceTimer() {
 if (digitalRead (BUTTON1) == LOW)
 { modeCount ++;
   if (modeCount > 3) modeCount = 0; }.
if (digital Read (BUTTON2) == LOW)
{   mode Count = 0; }.

   debounce Active = false; }.

void IRAM-ATTR onButtonISR()
{  if (!debounce Active)
   { debounceActive = true;
     timer Write (debounce Timer, 0);
     timer AlarmWrite (debounce Timer, DEBOUNCE_US,
                         false);
    timer Alarm Enable (debounceTimer); }}.

   // ---- Display helper ----
void ShowMode (const char* text) {
  display. Clear Display();
  display. Set Text Size (2);
  display. Set Text Color (SSD1306-WHITE);
  display. Set Cursor (10, 26);
  display. PrintIn (text);
   display. display ();
 }

void setup()
{  //PWM  setup
   ledcSetup (2, 5000, 8);
   ledc AttachPin (LED3, 2);
   PinMode ( LED1, OUTPUT);
  Pin Mode (LED2, OUTPUT);

  digital Write ( LED1, LOW);
  digital Write (LED2, LOW);
```

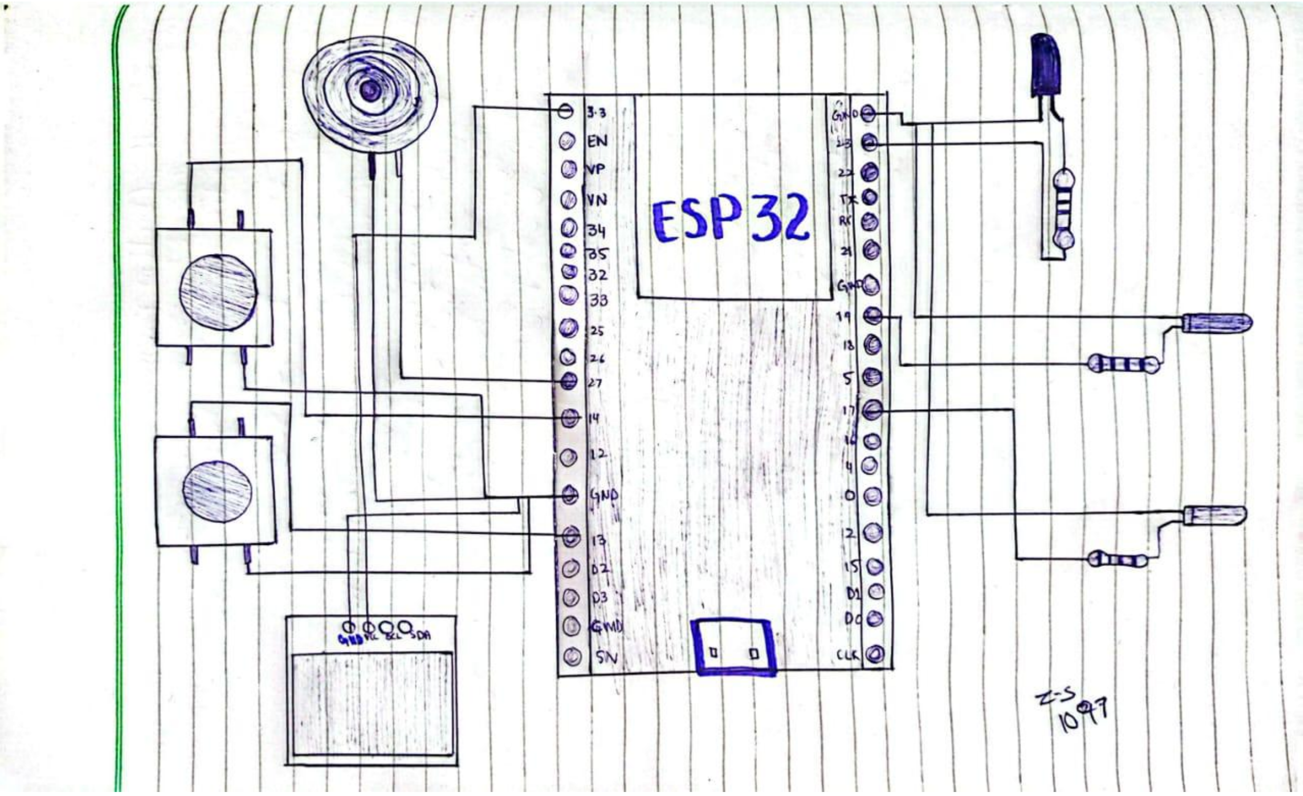


```
PinMode (BUTTON1, INPUT_PULLUP);
Pin Mode (BUTTON2, INPUT_PULLUP);
attach Inderrust (BUTTON 1, OnButtonISR, FALLING);
attach Interrupt (BUTTON2, OnButtonISR, FALLING);.

debounce Timer = timer Begin (3, 80, true);
timer Attach Interrupt (debounce Tiner, &on Debounce Tiner,
                        True);
 Wire.begin(21, 22);
   if (!deplay.begin(SSD1306_SWITCHCAPVCC,
       OLED_ADDR)) {
       for (;;); }

   display. clear Display();
   Show Mode ("All OFF");

  void  loop() {
   Switch (mode Count) {
    case 0; //All off
    digitalWrite (LED1, LOW);
    digital Write (LED2, LOW);
    Ledc Write (2, 0);
    Show Mode ("BOTH OFF");
    break;

   case 1; // Alternate Blink.
    show Mode ("ALTERNATE");
   digital Write (LED1, LOW);
   Ledc Write (2, 0);
   delay (400);
   digital Write (LED1, HIGH);
   digitalWrite (LED2, LOW);
   ledc Write (2, 0);
   delay(400);
```



```
break;

case 2:   // All On
 digitalWrite (LED1, LOW);
 digtal Write ( LED2, LOW);
 ShowMode ("PMW FADE");
for ( int d = 0; d <= 255 ; d = d+5)
    ledcWrite (2, d);
    delay (10);
}
for (int d = 255; d>=0; d=d-5).
   ledcWrite (2, d);
{  delay (10); }.
   break; }

  delay (50); }.
```

- **Wokwi diagram:**



- **Wokwi link:**

  https://wokwi.com/projects/445485546359644161

- **GitHub link:**

  https://github.com/zainab873/embedded-iot-zainab/tree/main/23-NTU-CS-FL-

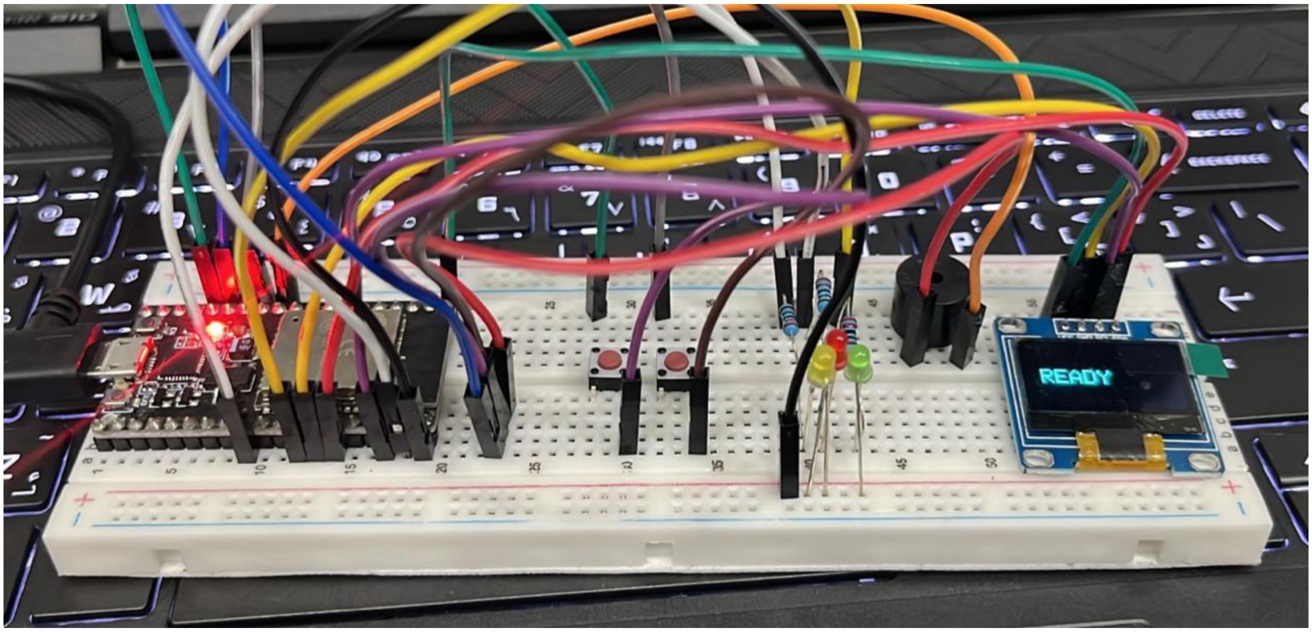
1097%20%20Assignment1/Task%20A%20--23-NTU-CS-FL-1097%20Zainab%20Sultan

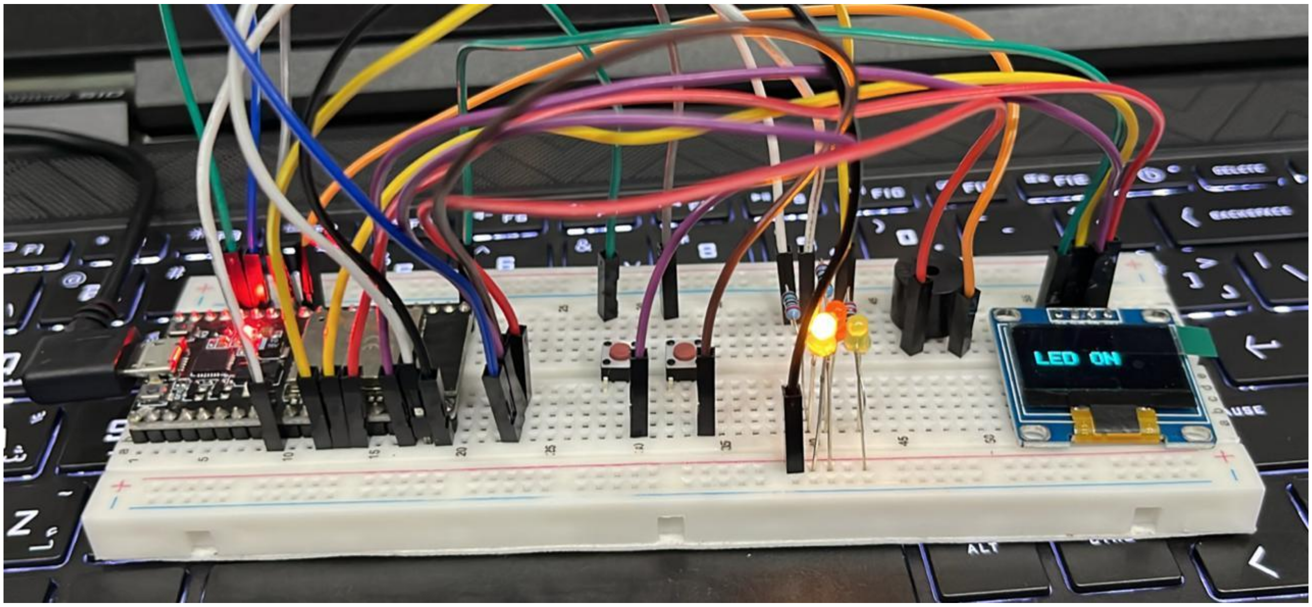
Task B

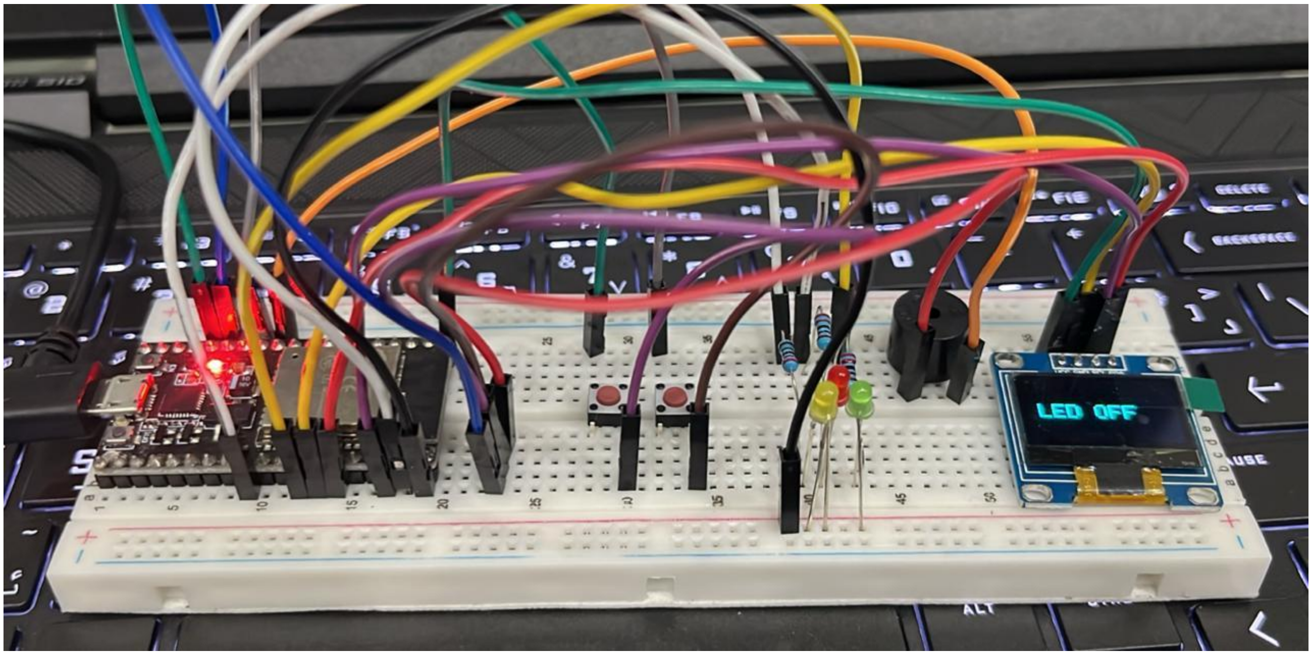
Coding: Use a single button with press-type detection (display the event on the OLED):

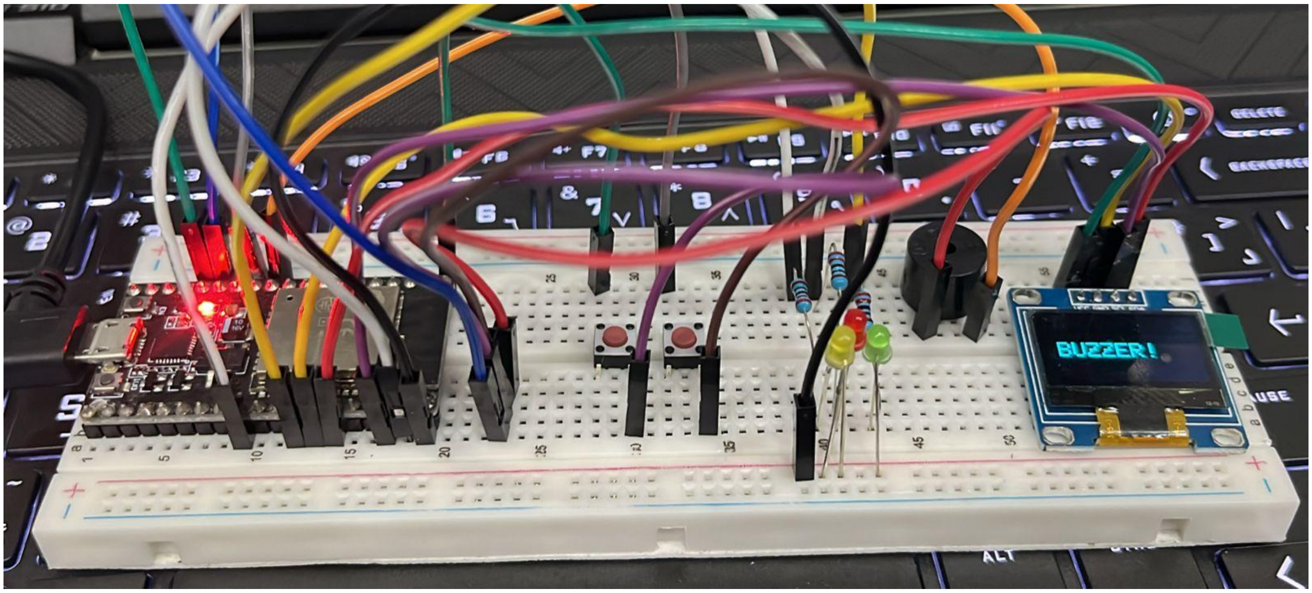
- **Short press → toggle LED**
- **Long press (> 1.5 s) → play a buzzer tone**

- KIT:

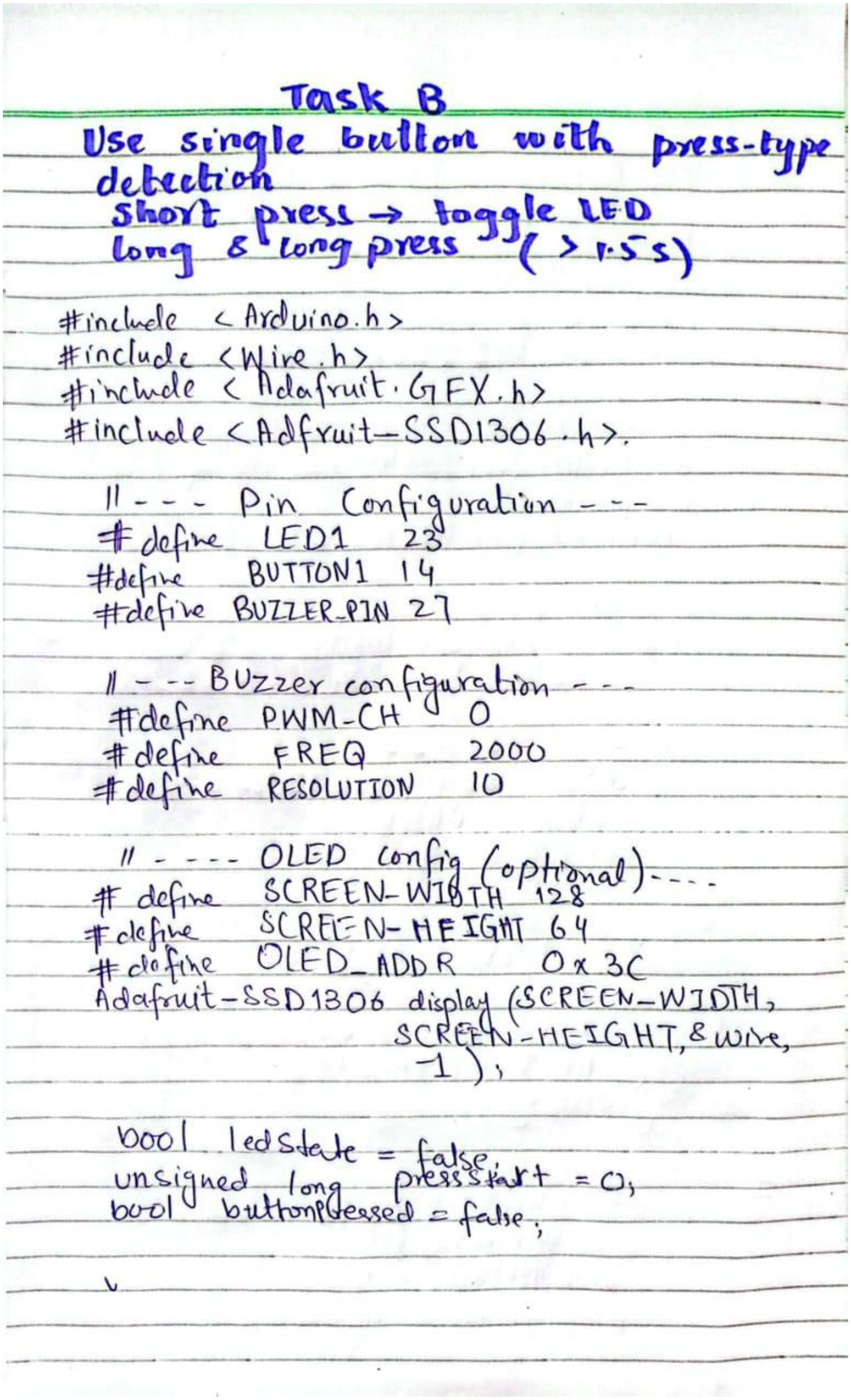



LED ON


LED OFF



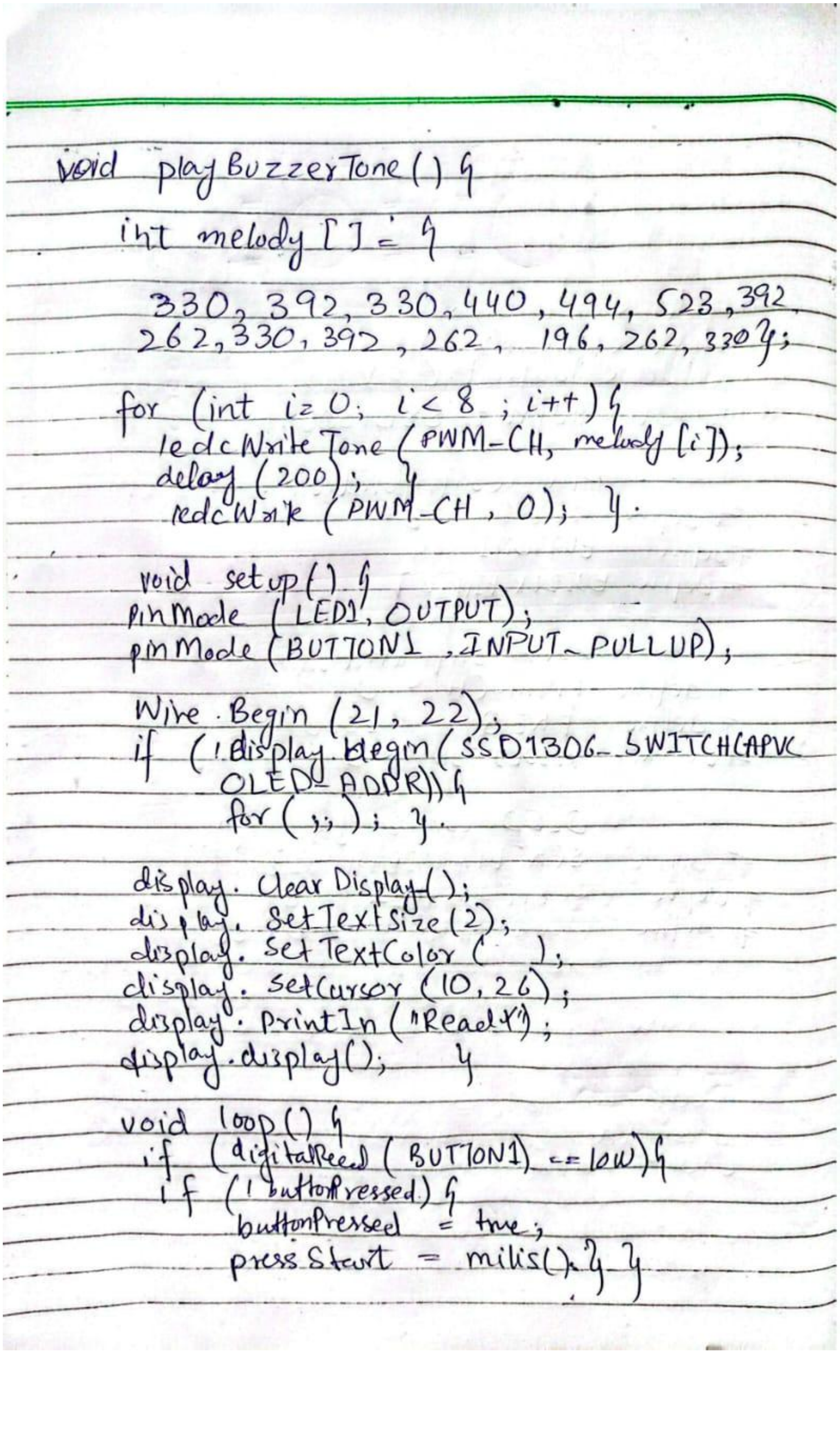
- **Handwritten code:**



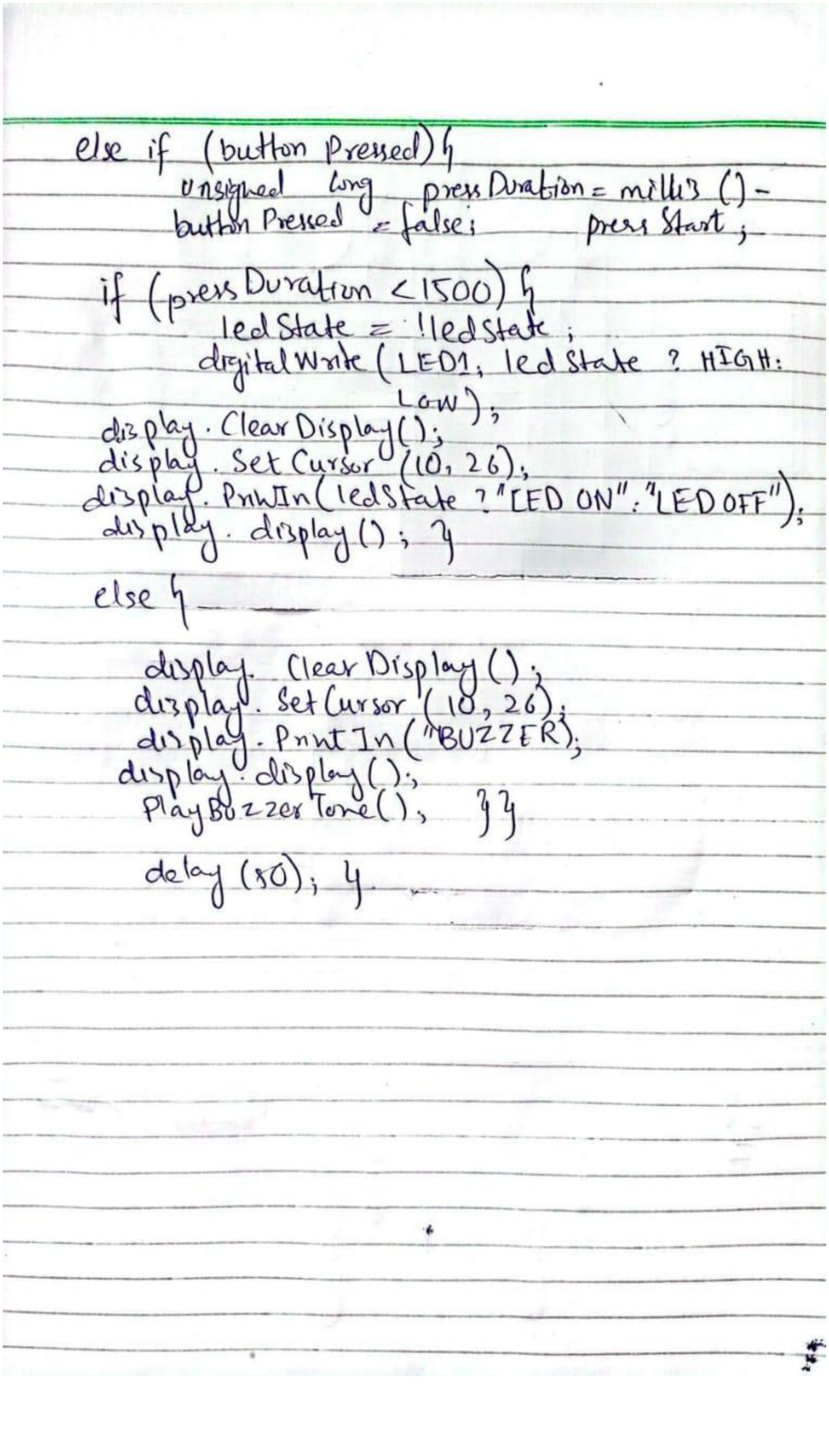
Task B

Use single button with press-type detection
Short press → toggle LED
long & long press (> 1.5s)

```
#include <Arduino.h>
#include <Wire.h>
#include < Adafruit.GFX.h>
#include <Adfruit-SSD1306.h>.

 // --- Pin Configuration ---
 #define LED1  23
#define  BUTTON1 14
#define BUZZER-PIN 27

 // --- BUzzer configuration ---
 #define PWM-CH    0
 #define FREQ      2000
#define RESOLUTION  10

  // ---- OLED config (optional)----
 # define  SCREEN-WIDTH 128
# define  SCREEN-HEIGHT 64
# define  OLED_ADDR     0x3C
Adafruit-SSD1306 display (SCREEN-WIDTH,
                   SCREEN-HEIGHT, &wire,
                   -1 );

 bool  ledState = false,
 unsigned long  pressStart = 0;
 bool  buttonPressed = false;
```

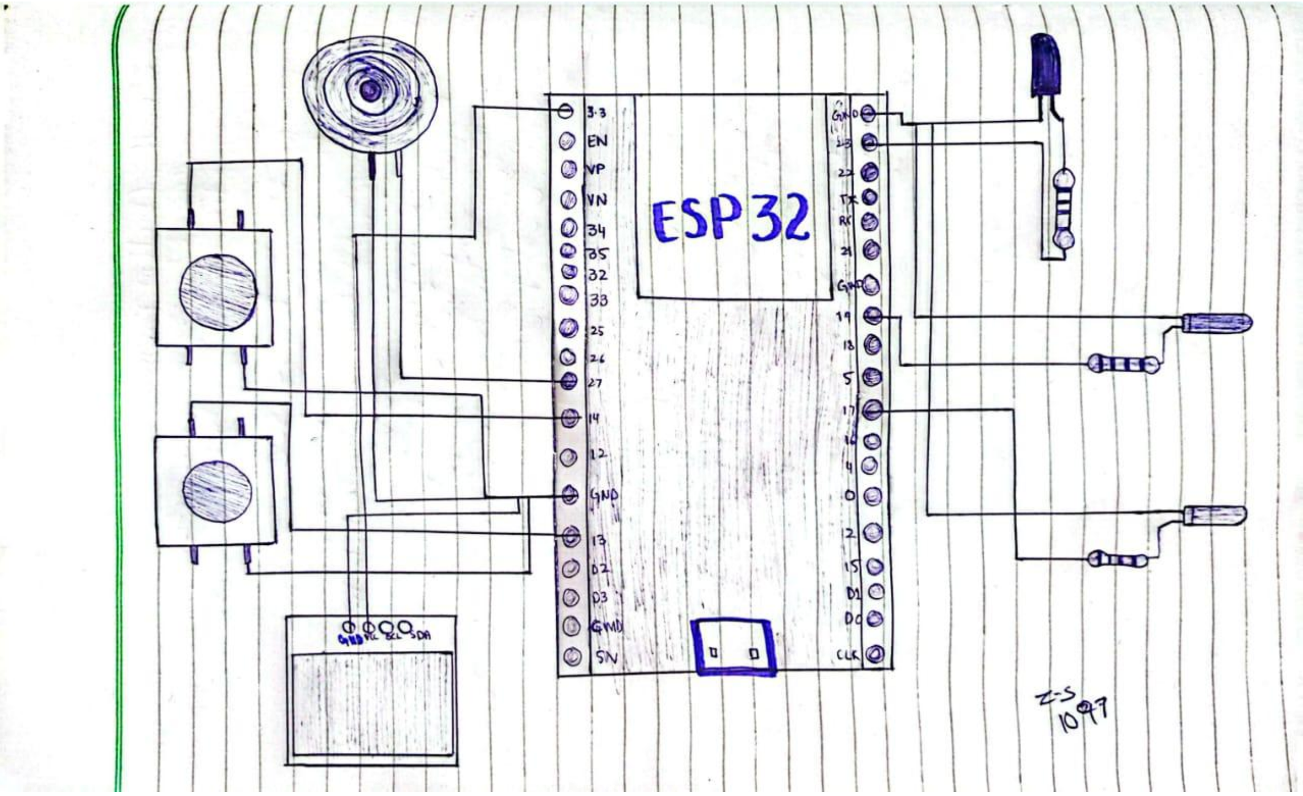


```
void playBuzzerTone() {
    int melody[] = {
      330, 392, 330, 440, 494, 523, 392
      262, 330, 392, 262, 196, 262, 330};

    for (int i=0; i<8 ; i++) {
      ledcWriteTone (PWM-CH, melody[i]);
      delay (200);   }
      ledcWrite (PWM-CH, 0);  }.

    void setup() {
    pinMode (LED1, OUTPUT);
    pinMode (BUTTON1 , INPUT_PULLUP);

    Wire.Begin (21, 22);
    if (!display.begin(SSD1306_SWITCHCAPVC
         OLED_ADDR)) {
         for ( ;; ); }.

    display. ClearDisplay();
    display. SetTextSize(2);
    display. SetTextColor (  );
    display. SetCursor (10, 26);
    display. PrintIn ("Ready");
    display.display();      }

    void loop() {
     if (digitalRead ( BUTTON1) == LOW) {
      if (!buttonPressed) {
            buttonPressed = true;
            pressStart = milis(); } }
```



```
else if (button Pressed) {
        unsigned long pressDuration = millis () -
        button Pressed = false;          press Start ;

  if (press Duration <1500) {
        led State = !led State ;
      digital Write ( LED1, led State ? HIGH:
                      LOW);
  display . Clear Display();
  display . Set Cursor (10, 26);
  display. PrintIn(ledState ?"LED ON": "LED OFF");
  display . display () ; }

  else {

     display. Clear Display () ;
     display. Set Cursor (10, 26);
     display. Print In ("BUZZER);
    display. display();
     PlayBuzzer Tone(),     }}

     delay (80); }
```

- **Wokwi diagram:**



- **Wokwi link:**

https://wokwi.com/projects/445711469091759105

- **Github link:**

  https://github.com/zainab873/embedded-iot-zainab/tree/main/23-NTU-CS-FL-1097%20%20Assignment1/Task%20B%20--23-NTU-CS-FL-1097%20Zainab%20Sultan

- **Assignment GitHub link:**

  https://github.com/zainab873/embedded-iot-zainab/tree/main/23-NTU-CS-FL-1097%20%20Assignment1