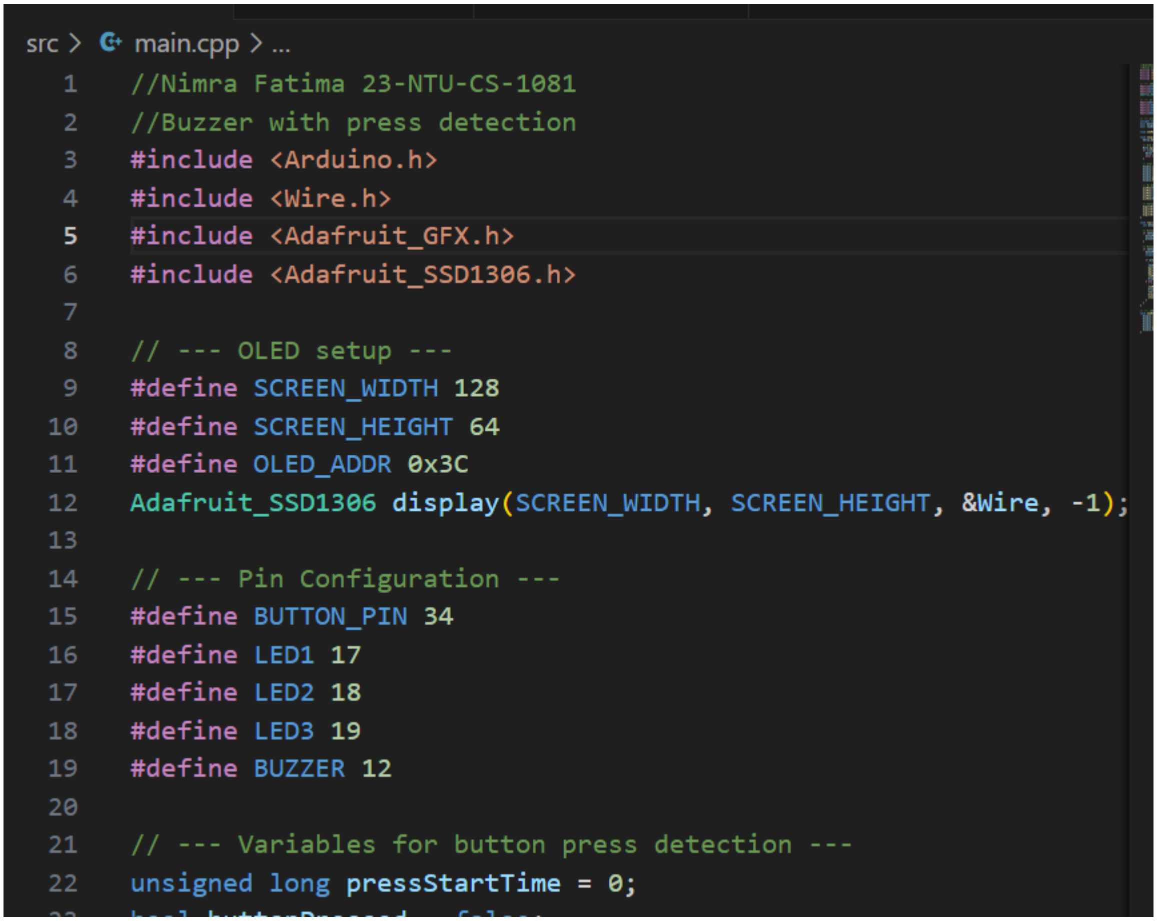
National Textile University, Faisalabad

Department of Computer Science

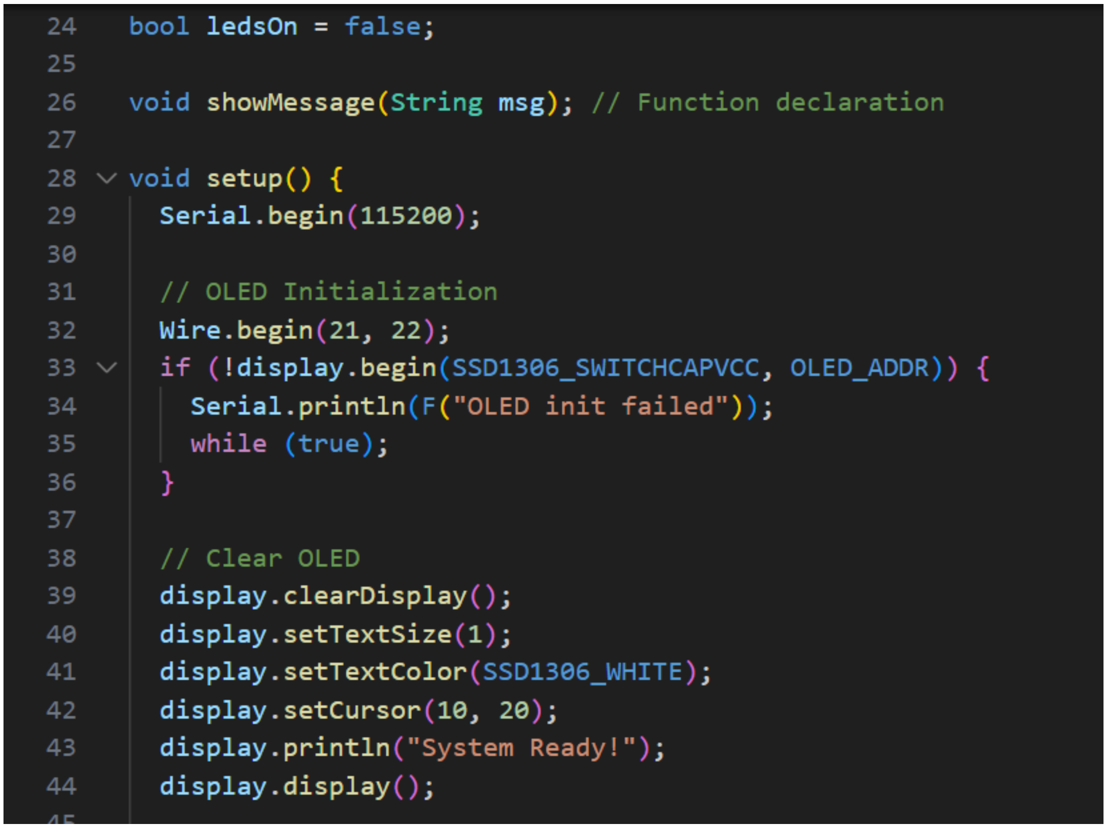
Name	Nimra Fatima
Reg no	23-NTU-CS-1081
Section	BSCS 5th -B
Subject	Embedded & IoT Systems
Assignment	1-Task(A)
Submitted to	Sir Nasir

Task B

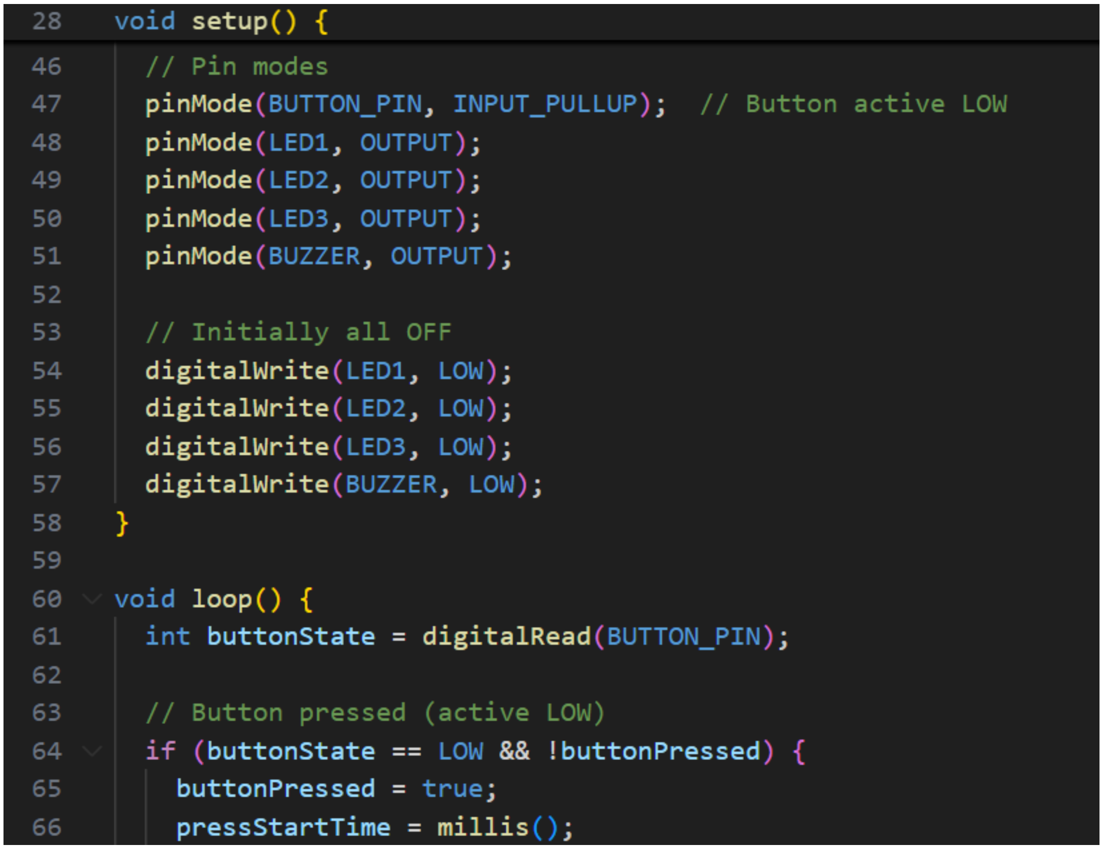
Task B — Coding: Use a single button with press-type detection (display the event on the OLED): • Short press → toggle LED • Long press (> 1.5 s) → play a buzzer tone.



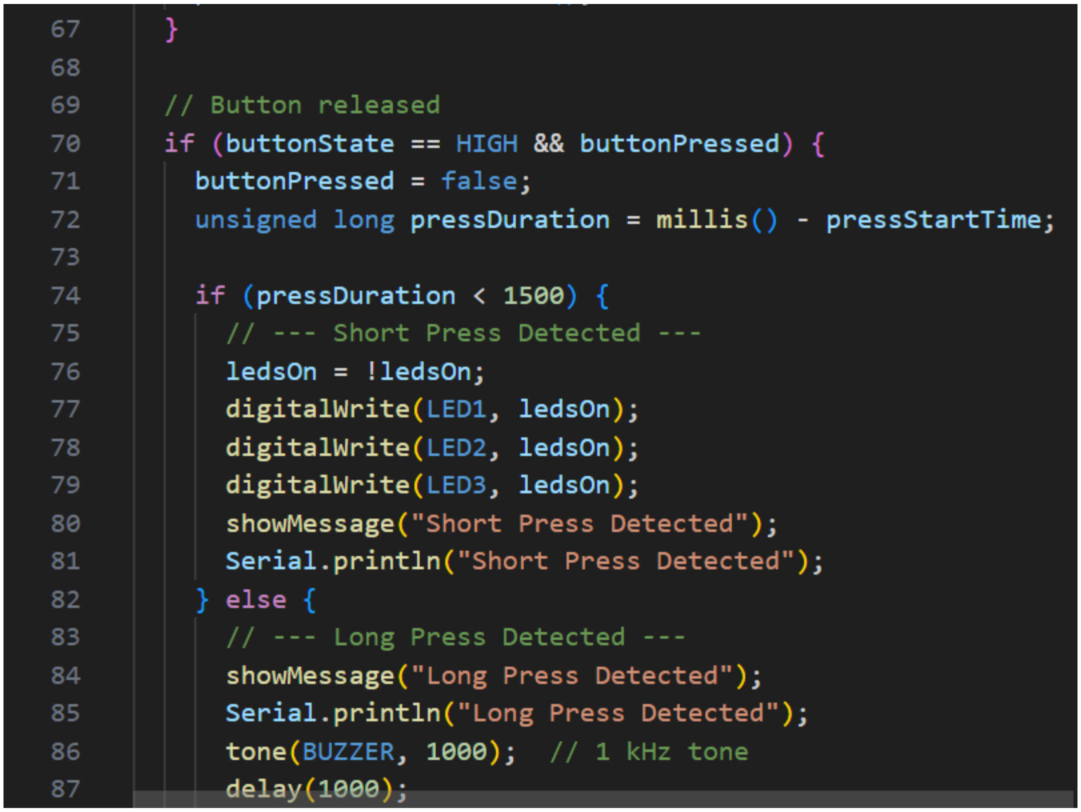
```
//Nimra Fatima 23-NTU-CS-1081
//Buzzer with press detection
#include <Arduino.h>
#include <Wire.h>
#include <Adafruit_GFX.h>
#include <Adafruit_SSD1306.h>

// --- 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 BUTTON_PIN 34
#define LED1 17
#define LED2 18
#define LED3 19
#define BUZZER 12

// --- Variables for button press detection ---
unsigned long pressStartTime = 0;
bool buttonPressed = false;
```



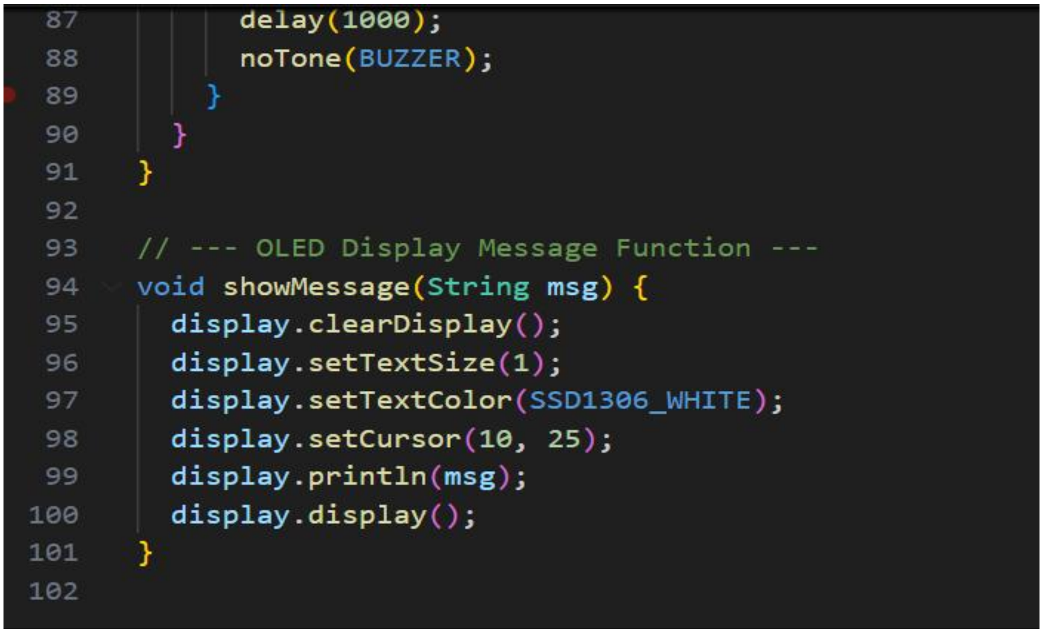
```
bool ledsOn = false;

void showMessage(String msg); // Function declaration

void setup() {
  Serial.begin(115200);

  // OLED Initialization
  Wire.begin(21, 22);
  if (!display.begin(SSD1306_SWITCHCAPVCC, OLED_ADDR)) {
    Serial.println(F("OLED init failed"));
    while (true);
  }

  // Clear OLED
  display.clearDisplay();
  display.setTextSize(1);
  display.setTextColor(SSD1306_WHITE);
  display.setCursor(10, 20);
  display.println("System Ready!");
  display.display();
```



```
void setup() {
  // Pin modes
  pinMode(BUTTON_PIN, INPUT_PULLUP);  // Button active LOW
  pinMode(LED1, OUTPUT);
  pinMode(LED2, OUTPUT);
  pinMode(LED3, OUTPUT);
  pinMode(BUZZER, OUTPUT);

  // Initially all OFF
  digitalWrite(LED1, LOW);
  digitalWrite(LED2, LOW);
  digitalWrite(LED3, LOW);
  digitalWrite(BUZZER, LOW);
}

void loop() {
  int buttonState = digitalRead(BUTTON_PIN);

  // Button pressed (active LOW)
  if (buttonState == LOW && !buttonPressed) {
    buttonPressed = true;
    pressStartTime = millis();
```

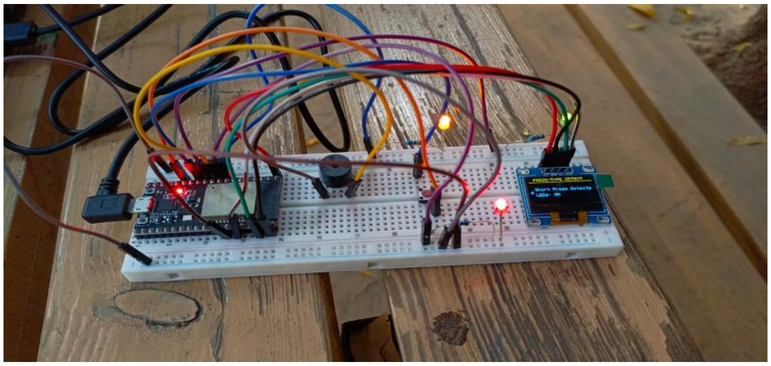


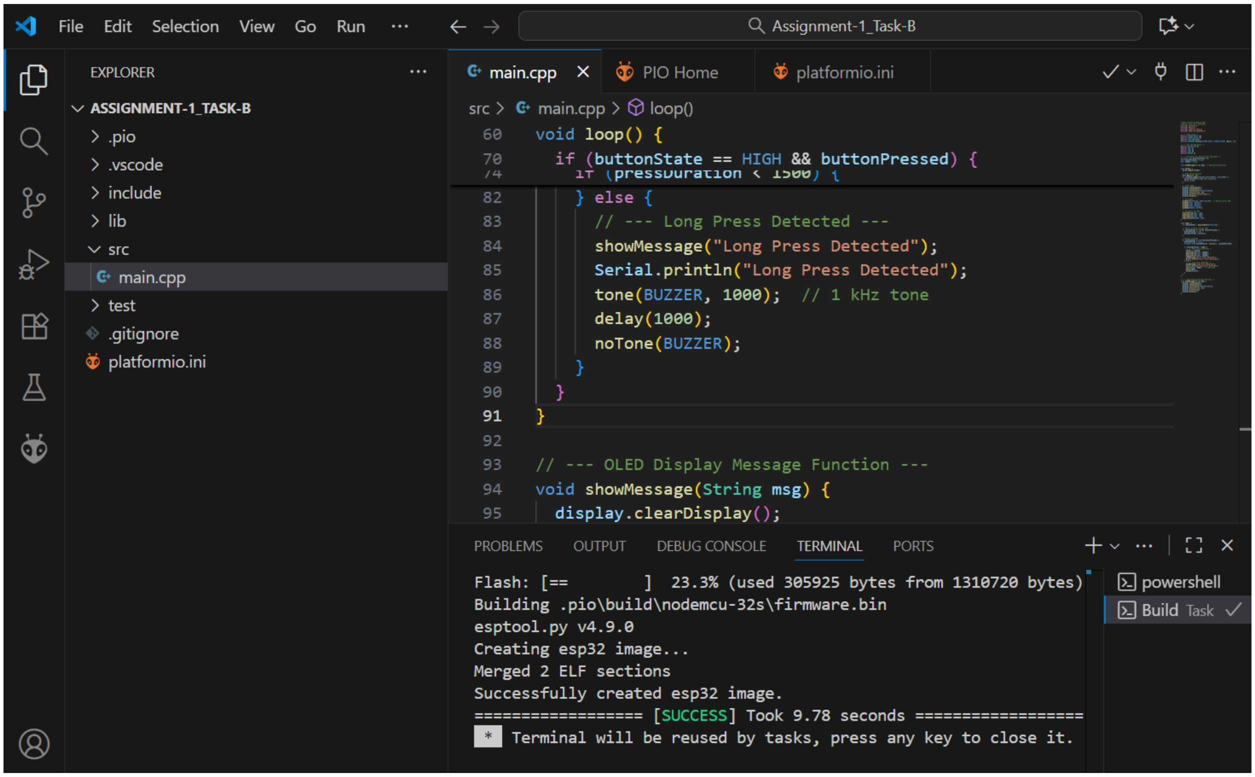
```
  }

  // Button released
  if (buttonState == HIGH && buttonPressed) {
    buttonPressed = false;
    unsigned long pressDuration = millis() - pressStartTime;

    if (pressDuration < 1500) {
      // --- Short Press Detected ---
      ledsOn = !ledsOn;
      digitalWrite(LED1, ledsOn);
      digitalWrite(LED2, ledsOn);
      digitalWrite(LED3, ledsOn);
      showMessage("Short Press Detected");
      Serial.println("Short Press Detected");
    } else {
      // --- Long Press Detected ---
      showMessage("Long Press Detected");
      Serial.println("Long Press Detected");
      tone(BUZZER, 1000);  // 1 kHz tone
      delay(1000);
```



```
      delay(1000);
      noTone(BUZZER);
    }
  }
}

// --- OLED Display Message Function ---
void showMessage(String msg) {
  display.clearDisplay();
  display.setTextSize(1);
  display.setTextColor(SSD1306_WHITE);
  display.setCursor(10, 25);
  display.println(msg);
  display.display();
}
```

Output:

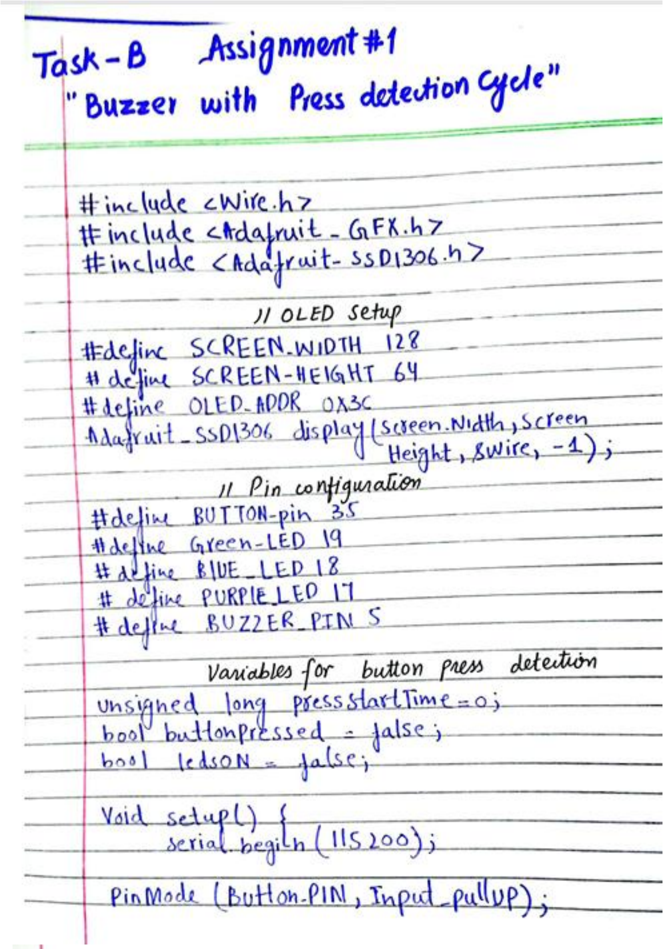




```
void loop() {
  if (buttonState == HIGH && buttonPressed) {
    if (pressDuration < 1500) {
    } else {
      // --- Long Press Detected ---
      showMessage("Long Press Detected");
      Serial.println("Long Press Detected");
      tone(BUZZER, 1000);  // 1 kHz tone
      delay(1000);
      noTone(BUZZER);
    }
  }
}

// --- OLED Display Message Function ---
void showMessage(String msg) {
  display.clearDisplay();
```

```
Flash: [==        ]  23.3% (used 305925 bytes from 1310720 bytes)
Building .pio\build\nodemcu-32s\firmware.bin
esptool.py v4.9.0
Creating esp32 image...
Merged 2 ELF sections
Successfully created esp32 image.
================== [SUCCESS] Took 9.78 seconds ==================
 *  Terminal will be reused by tasks, press any key to close it.
```

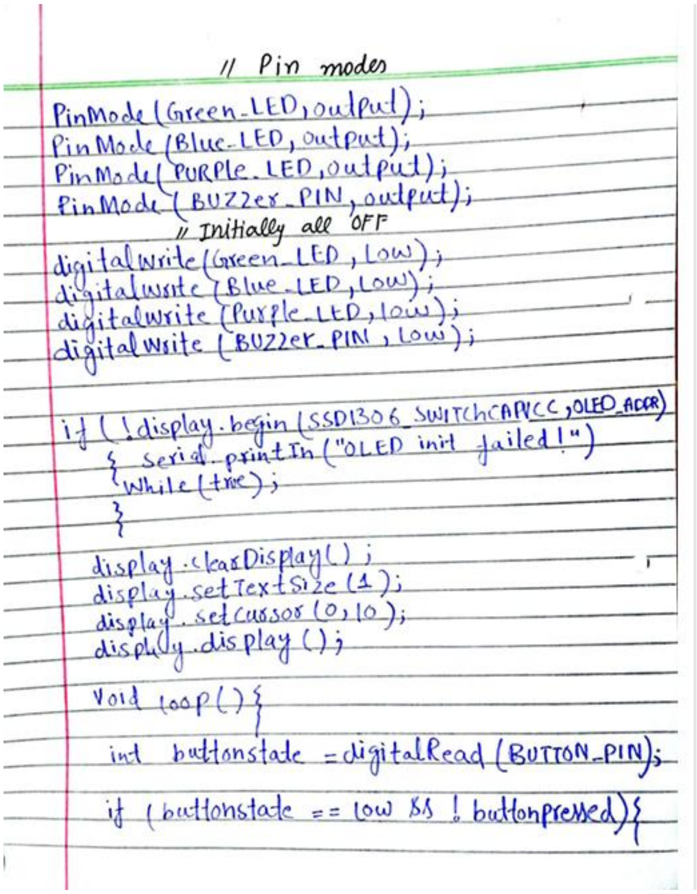
Handwritten code:



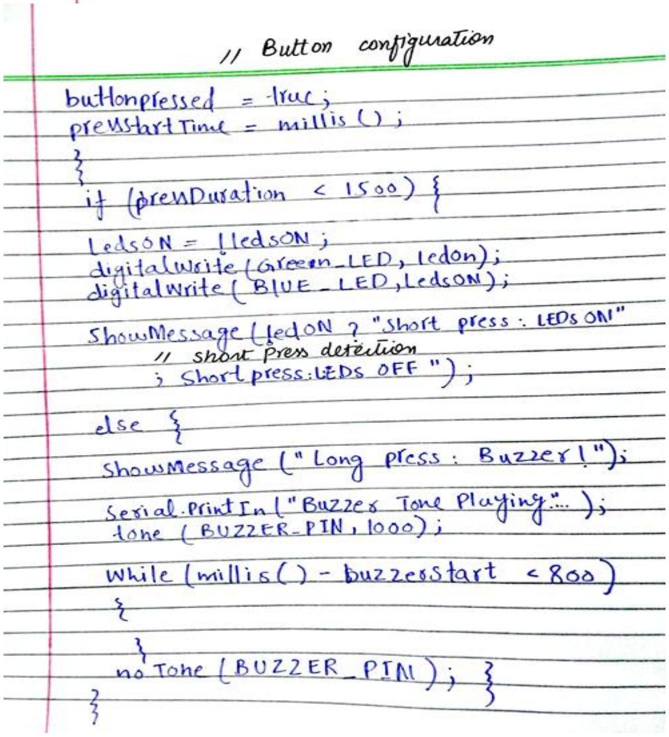
Task-B Assignment #1

"Buzzer with Press detection Cycle"

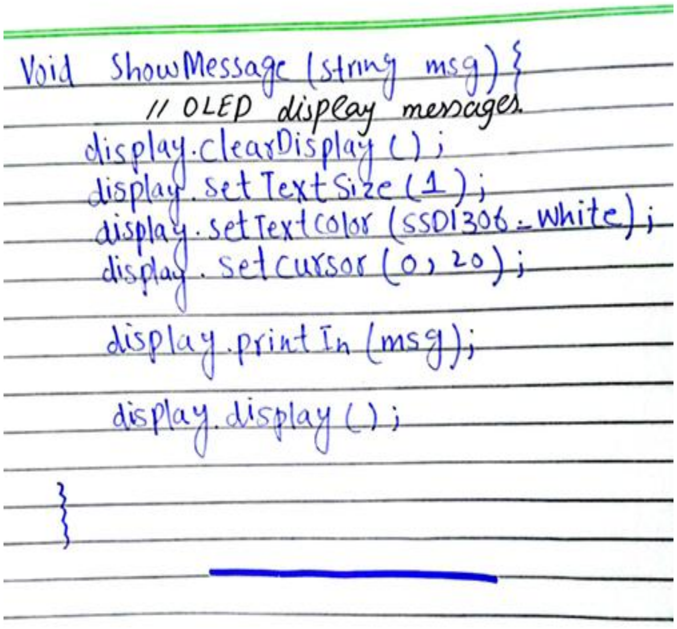
```
#include <Wire.h>
#include <Adafruit_GFX.h>
#include <Adafruit-SSD1306.h>

            // 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 BUTTON-pin 35
#define Green-LED 19
# define BIUE_LED 18
# define PURPLE_LED 17
# define BUZZER_PIN 5

        Variables for button press detection
unsigned long pressStartTime = 0;
bool buttonpressed = false;
bool ledsON = false;

Void setup() {
    serial.begin(115200);

PinMode (Button-PIN, Input_pullUP);
```



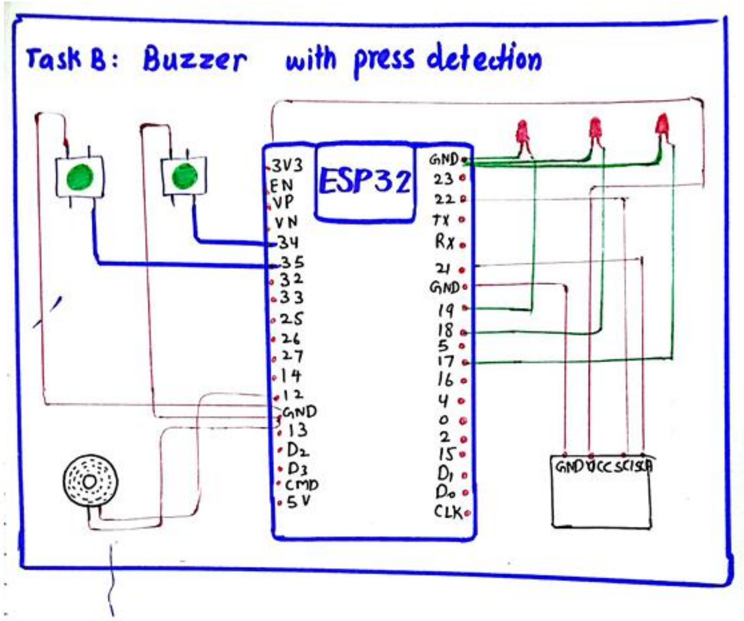
```
            // Pin modes
PinMode(Green-LED, output);
PinMode (Blue-LED, output);
PinMode( PURPle.LED, output);
PinMode ( BUZZer_PIN, output);
        // Initially all OFF
digitalwrite(Green_LED , Low);
digitalwrite (Blue_LED, LOW);
digitalwrite (Purple_LED, low);
digitalwrite ( BUZZer_PIN , Low);


if ( !display.begin (SSD1306_SWITCHCAPVCC, OLED_ADDR)
   { serial.printIn ("OLED init failed!")
   While(true);
   }

 display.clearDisplay() ;
 display.setTextSize (1);
 display.setCursor (0,10);
 display.display ();

 Void loop() {

  int buttonstate = digitalRead (BUTTON_PIN);

  if (buttonstate == low && ! buttonpressed){
```



```
        // Button configuration
buttonpressed = true;
pressStartTime = millis ( ) ;
}
 if (pressDuration < 1500) {

 LedsON = !ledsON ;
 digitalwrite (Green_LED, ledon);
 digitalwrite ( BIUE_LED, LedsON);

 ShowMessage (ledON ? "Short press : LEDS ON"
      // short press detection
      ; Short press:LEDS OFF ");

 else {
  ShowMessage ("Long press : Buzzer!");

  Serial.PrintIn("Buzzer Tone Playing"... );
  tone ( BUZZER-PIN, 1000);

  While (millis() - buzzerStart < 800)
   {

    }
   noTone (BUZZER_PIN); }
}
```



```
Void ShowMessage (string msg) {
     // OLED display messages
 display.clearDisplay ();
 display.setTextSize (1);
 display.setTextColor (SSD1306_White);
 display.setCursor (0, 20);

 display.printIn (msg);

 display.display ();

}
```



Task B: Buzzer with press detection