

National Textile University, Faisalabad



Department of Computer Science

Name	Nimra Fatima
Reg no	23-NTU-CS-1081
Section	BSCS 5 th -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.

```
src > main.cpp > ...
1  //Nimra Fatima 23-NTU-CS-1081
2  //Buzzer with press detection
3  #include <Arduino.h>
4  #include <Wire.h>
5  #include <Adafruit_GFX.h>
6  #include <Adafruit_SSD1306.h>
7
8  // --- OLED setup ---
9  #define SCREEN_WIDTH 128
10 #define SCREEN_HEIGHT 64
11 #define OLED_ADDR 0x3C
12 Adafruit_SSD1306 display(SCREEN_WIDTH, SCREEN_HEIGHT, &Wire, -1);
13
14 // --- Pin Configuration ---
15 #define BUTTON_PIN 34
16 #define LED1 17
17 #define LED2 18
18 #define LED3 19
19 #define BUZZER 12
20
21 // --- Variables for button press detection ---
22 unsigned long pressStartTime = 0;
```

```

24   bool ledsOn = false;
25
26   void showMessage(String msg); // Function declaration
27
28   void setup() {
29       Serial.begin(115200);
30
31       // OLED Initialization
32       Wire.begin(21, 22);
33       if (!display.begin(SSD1306_SWITCHCAPVCC, OLED_ADDR)) {
34           Serial.println(F("OLED init failed"));
35           while (true);
36       }
37
38       // Clear OLED
39       display.clearDisplay();
40       display.setTextSize(1);
41       display.setTextColor(SSD1306_WHITE);
42       display.setCursor(10, 20);
43       display.println("System Ready!");
44       display.display();
45

```

```

28   void setup() {
46       // Pin modes
47       pinMode(BUTTON_PIN, INPUT_PULLUP); // Button active LOW
48       pinMode(LED1, OUTPUT);
49       pinMode(LED2, OUTPUT);
50       pinMode(LED3, OUTPUT);
51       pinMode(BUZZER, OUTPUT);
52
53       // Initially all OFF
54       digitalWrite(LED1, LOW);
55       digitalWrite(LED2, LOW);
56       digitalWrite(LED3, LOW);
57       digitalWrite(BUZZER, LOW);
58   }
59
60   void loop() {
61       int buttonState = digitalRead(BUTTON_PIN);
62
63       // Button pressed (active LOW)
64       if (buttonState == LOW && !buttonPressed) {
65           buttonPressed = true;
66           pressStartTime = millis();

```

```

67     }
68
69     // Button released
70     if (buttonState == HIGH && buttonPressed) {
71         buttonPressed = false;
72         unsigned long pressDuration = millis() - pressStartTime;
73
74         if (pressDuration < 1500) {
75             // --- Short Press Detected ---
76             ledsOn = !ledsOn;
77             digitalWrite(LED1, ledsOn);
78             digitalWrite(LED2, ledsOn);
79             digitalWrite(LED3, ledsOn);
80             showMessage("Short Press Detected");
81             Serial.println("Short Press Detected");
82         } else {
83             // --- Long Press Detected ---
84             showMessage("Long Press Detected");
85             Serial.println("Long Press Detected");
86             tone(BUZZER, 1000); // 1 kHz tone
87             delay(1000);

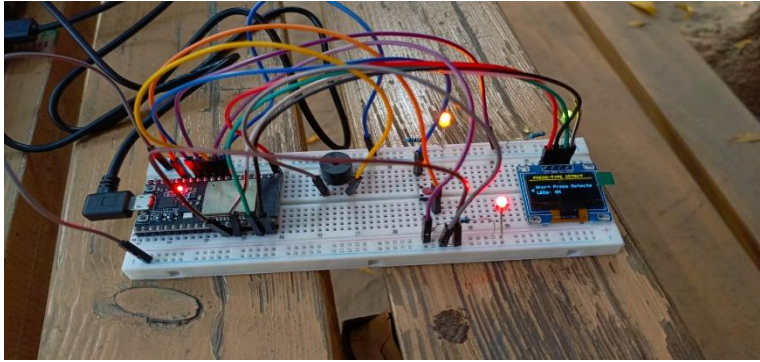
```

```

87         delay(1000);
88         noTone(BUZZER);
89     }
90 }
91
92
93 // --- OLED Display Message Function ---
94 void showMessage(String msg) {
95     display.clearDisplay();
96     display.setTextSize(1);
97     display.setTextColor(SSD1306_WHITE);
98     display.setCursor(10, 25);
99     display.println(msg);
100    display.display();
101 }
102

```

Output:



```
File Edit Selection View Go Run ... Assignment-1_Task-B
```

EXPLORER

- ASSIGNMENT-1_TASK-B
 - .pio
 - .vscode
 - include
 - lib
 - src
 - main.cpp
 - test
 - .gitignore
 - platformio.ini

main.cpp

```
src > main.cpp > loop()
60 void loop() {
70   if (buttonState == HIGH && buttonPressed) {
74     if (pressDuration < 1500) {
82     } else {
83       // --- Long Press Detected ---
84       showMessage("Long Press Detected");
85       Serial.println("Long Press Detected");
86       tone(BUZZER, 1000); // 1 kHz tone
87       delay(1000);
88       noTone(BUZZER);
89     }
90   }
91 }
92
93 // --- OLED Display Message Function ---
94 void showMessage(String msg) {
95   display.clearDisplay();
```

TERMINAL

```
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 BLUE_LED 18
#define PURPLE_LED 17
#define BUZZER_PIN 5

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

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

    pinMode(BUTTON_PIN, INPUT_PULLUP);
```

```
    // Button configuration
    buttonPressed = true;
    pressStartTime = millis();
}

if (pressDuration < 1500) {
    ledOn = !ledOn;
    digitalWrite(GREEN_LED, ledOn);
    digitalWrite(BLUE_LED, ledOn);

    showMessage(ledOn ? "Short press: LEDs ON"
        : "Short press: LEDs OFF");
} else {
    showMessage("Long press: Buzzer!");
    Serial.println("Buzzer Tone Playing...");
    tone(BUZZER_PIN, 1000);

    while (millis() - buzzerStart < 800) {
        // Buzzer on
    }
    noTone(BUZZER_PIN);
}
```

```
// 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.println("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) {
```

```
        void showMessage(String msg) {
            // OLED display messages
            display.clearDisplay();
            display.setTextSize(1);
            display.setTextColor(SSD1306_WHITE);
            display.setCursor(0, 20);

            display.println(msg);

            display.display();
        }
    }
}
```

Task B: Buzzer with press detection

