**National Textile University, Faisalabad**

A logo with blue and red squares

AI-generated content may be incorrect.

**Department of Computer Science**

|  |  |
| --- | --- |
| **Name:** | Meerab Zahra |
| **Class:** | BSCS 5th A |
| **Registration No:** | 23-NTU-CS-1049 |
| **Assignment:** | 1 |
| **Course Name:** | Embedded IoT Systems |
| **Submitted To:** | Sir Nasir Mahmood |
| **Submission Date:** | October 26, 2025 |

**Question 3 — Implementation**

Circuit Diagram: Design a Wokwi circuit and draw a neat hand-sketch including:

• 2 push buttons

• 3 LEDs

• 1 buzzer

• 1 OLED

**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

**Code:**

//Meerab Zahra

//23-NTU-CS-1049

//Assignment1Q3\_PartB

#include <Arduino.h>

#include <Wire.h>

#include <Adafruit\_GFX.h>

#include <Adafruit\_SSD1306.h>

// OLED setup

#define SCREEN\_WIDTH 128

#define SCREEN\_HEIGHT 64

Adafruit\_SSD1306 display(SCREEN\_WIDTH, SCREEN\_HEIGHT, &**Wire**, -1);

// Pin definitions

#define BUTTON\_PIN 36

#define LED\_PIN 18

#define BUZZER\_PIN 19

// Variables

bool ledState = false;

unsigned long pressStart = 0;

bool buttonPressed = false;

void setup() {

  pinMode(BUTTON\_PIN, INPUT);   // external pull-down or use INPUT\_PULLUP

  pinMode(LED\_PIN, OUTPUT);

  pinMode(BUZZER\_PIN, OUTPUT);

  // OLED init

  if (!display.begin(SSD1306\_SWITCHCAPVCC, 0x3C)) for(;;);

  display.clearDisplay();

  display.setTextSize(2);

  display.setTextColor(SSD1306\_WHITE);

  display.setCursor(0, 20);

  display.print("Ready");

  display.display();

}

void loop() {

  int buttonState = digitalRead(BUTTON\_PIN);

  // When button is pressed

  if (buttonState == HIGH && !buttonPressed) {

    buttonPressed = true;

    pressStart = millis();  // mark time

  }

  // When button is released

  if (buttonState == LOW && buttonPressed) {

    unsigned long pressDuration = millis() - pressStart;

    buttonPressed = false;

    if (pressDuration < 1500) {

      // Short press → Toggle LED

      ledState = !ledState;

      digitalWrite(LED\_PIN, ledState);

      display.clearDisplay();

      display.setTextSize(2);

      display.setCursor(0, 20);

      if (ledState) {

        // LED turned ON

        display.print("Short Hold");

        display.display();

      } else {

        // LED turned OFF

        display.print("Short Hold");

        display.display();

        delay(1000);

        display.clearDisplay();

        display.setTextSize(2);

        display.setCursor(0, 20);

        display.print("Ready");

        display.display();

      }

    } else {

      // Long press → Play buzzer tone

      display.clearDisplay();

      display.setTextSize(2);

      display.setCursor(0, 20);

      display.print("Long Hold");

      display.display();

      tone(BUZZER\_PIN, 1000, 3000); // buzzer ON for 3 seconds

      delay(2000);                  // wait until buzzer finishes

      // After buzzer stops, return to "Ready"

      display.clearDisplay();

      display.setTextSize(2);

      display.setCursor(0, 20);

      display.print("Ready");

      display.display();

    }

  }

}

**Wokwi Circuit Diagram**

**A computer circuit board with wires

AI-generated content may be incorrect.**

**Project Goal**

* When the circuit has just started, it shows **Ready** on the OLED.
* When the button is pressed for less than 1.5 second, it toggles the LED state. If the LED was off and button is held for a short while (less than 1.5 seconds), it turns LED on and vice versa. Also, OLED displays this event as short hold which appears for a specified time, after that OLED is again set to Ready.
* When the button is pressed for long duration (more than 1.5 seconds), the buzzer beeps for 3 seconds and the OLED displays this event as Long hold. OLED is then updated to display Ready after a specified time of event occurrence.

**Wokwi Circuit Output**

* **Simulation Start**

A circuit board with wires

AI-generated content may be incorrect.

* **Press the button for short duration and LED is on**

**A computer circuit board with wires

AI-generated content may be incorrect.**

* **Press the button again for short duration and LED toggles its state**

**A computer chip with wires

AI-generated content may be incorrect.**

* **Press the button for long duration and buzzer beeps**

**A computer circuit board with wires

AI-generated content may be incorrect.**

**Hardware Circuit Setup**

In the designed circuit, ESP32 has been attached to a breadboard. A jumper wire from voltage pin (3.3V) of ESP32 has been connected to the voltage line of breadboard so voltage can be shared among components. Same has been done to share ground of ESP32 among the components. An OLED display has been connected to ESP32 with its ground wire connected to ground, voltage wire connected to voltage, SCL connected to pin 22 and SDA connected to pin 21. Red LED has been connected in such a way that its negative side is grounded and on positive side, 1 kilo ohm resistor is connected and that resistor is connected to pin 18. There is one push button in the circuit also. To connect button, take any of two pairs and use both pins. One pin is attached to voltage and to the other pin, 10 kilo ohm resistor is attached. The resistor side connected to button pin is then connected to GPIO 36 and the other side of resistor is grounded. The buzzer is attached such that the positive side has been connected to GPIO 19 and negative side is grounded.

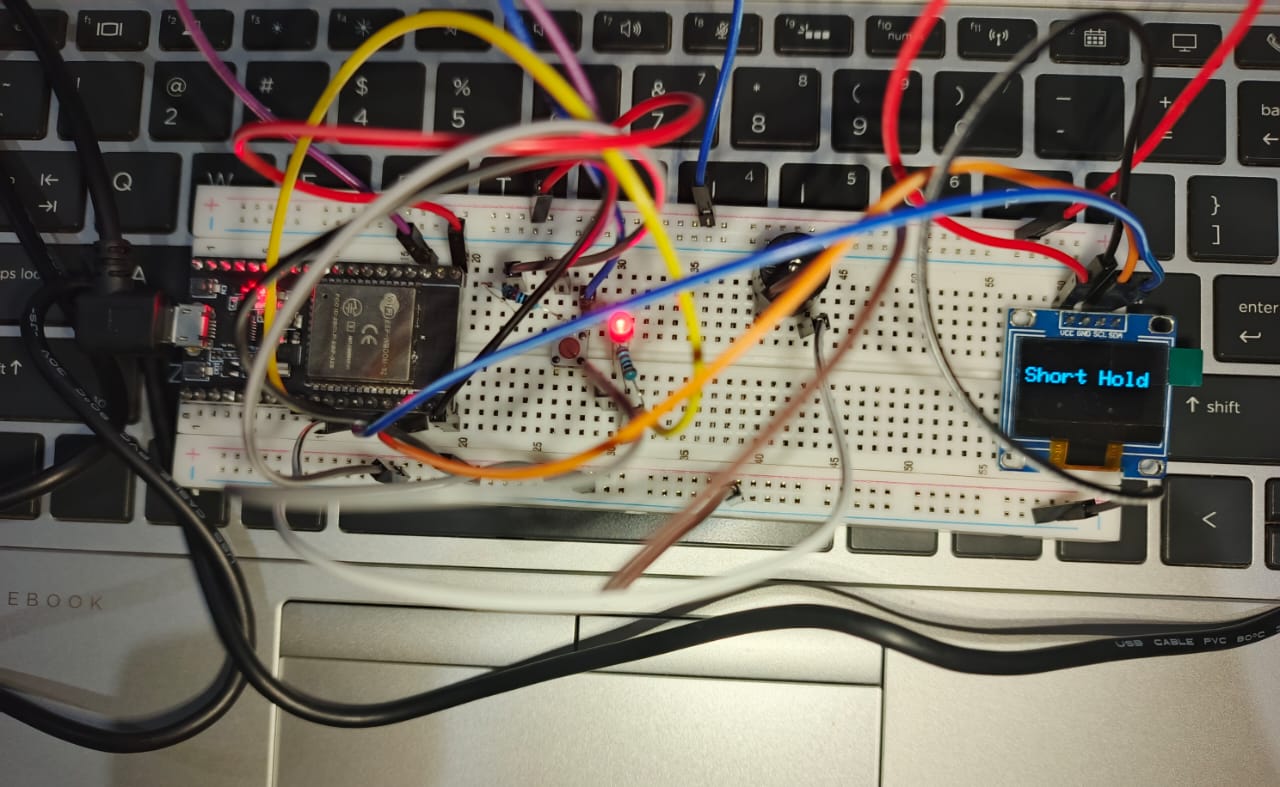
**Hardware Circuit Output**

* **Circuit Simulation**

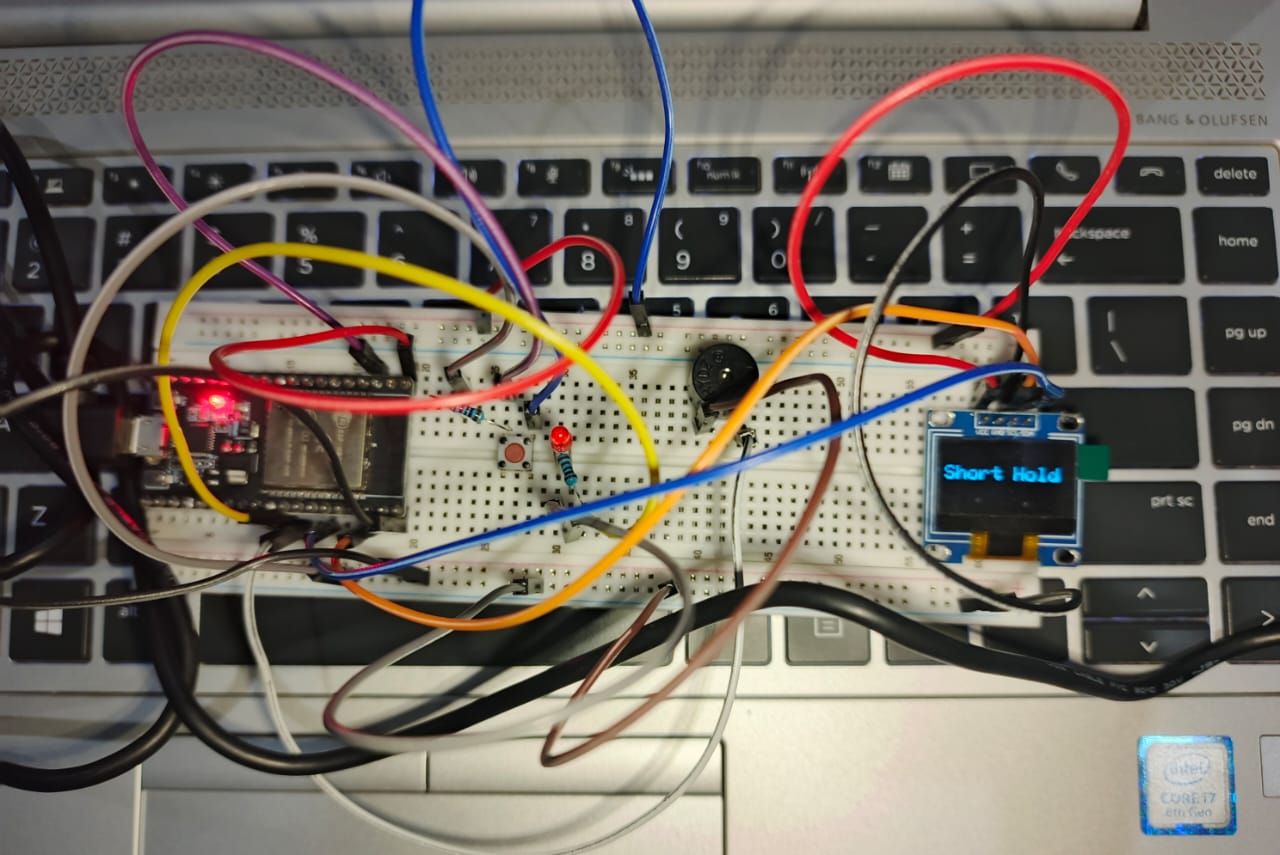
A computer with wires on it

AI-generated content may be incorrect.

* **Short hold turning LED on**



* **Short hold turning LED on**



* **Long hold leading to buzzer beep**

A computer keyboard with wires on it

AI-generated content may be incorrect.

**Note:** In the Wokwi simulation, the buzzer icon may appear active at startup due to simulator behavior, but in actual hardware, the buzzer remains off until a long button press is detected.