National Textile University, Faisalabad

Department of Computer Science

Name:	Muhammad Umar Mushtaq
Class:	BSCS_B 5th Semester
Registration No:	23-NTU-CS-1077
Assignment:	01
Course Name:	Embedded Iot System
Submitted To:	**Sir Nasir Mehmood**
Submission Date:	25/10/2025

TASK 2 – Button Press Duration Detection (Short and Long Press Detection with OLED)

Objective:
The goal of this task is to make a system using ESP32 that can detect both short and long button presses and perform different actions for each type. The OLED display will show messages to indicate which type of press was detected.

Introduction:
In many embedded systems, one button can perform multiple tasks depending on how long it is pressed.
In this project, we use that concept.
A short press will toggle an LED ON or OFF, and a long press will turn ON a buzzer for a short time.
An OLED display is used to show whether a short or long press was detected.

Hardware Components Required:

1. **ESP32 DevKitC V4**
2. **OLED Display (128x64 I2C)**
3. **One Push Button**
4. **One LED**
5. **One Buzzer**
6. **One Resistor (for LED)**
7. **Breadboard and jumper wires**

Pin Connections:
LED → GPIO 5
Button → GPIO 25
Buzzer → GPIO 18
OLED SDA → GPIO 21
OLED SCL → GPIO 22

Software and Libraries Used:

- **Arduino IDE for writing and uploading the code**
- **Adafruit_SSD1306 and Adafruit_GFX libraries for OLED display**
- **Wokwi Simulator for virtual testing**

Working Principle:

1. **The ESP32 continuously checks the button's input.**
2. **When the button is pressed, the current time is recorded using the millis() function.**
3. **When the button is released, the system calculates how long the button was held.**
4. **If the press time is less than 1 second, it is a short press, and the LED toggles.**
5. **If the press time is more than 1.5 seconds, it is a long press, and the buzzer turns ON for a short time.**
6. **The OLED display shows messages such as "Short Press Detected" or "Long Press Detected."**
7. **After that, the system resets and waits for the next button press.**

Code Explanation:

- **The millis() function is used to measure how long the button is pressed.**
- **Using if–else statements, the program decides which action to perform (LED toggle or buzzer ON).**
- **digitalWrite() controls the LED and buzzer.**
- **The OLED screen updates automatically to show which type of press was detected.**
- **A small delay is used for debouncing to prevent false readings.**

- **Serial Monitor helps in debugging and shows press duration in real-time.**

Testing Steps:

1. **Upload the program to ESP32.**
2. **Open Serial Monitor to see press timing.**
3. **Press and quickly release the button → LED will toggle.**
4. **Press and hold the button for more than 1.5 seconds → Buzzer will turn ON.**
5. **OLED screen will display “Short Press” or “Long Press.”**
6. **Repeat to confirm accurate detection.**

Results:

- **The ESP32 successfully detects both short and long presses.**
- **The OLED clearly shows the type of press.**
- **LED and buzzer work properly without delay or false triggers.**
- **The same code works fine in both actual hardware and the Wokwi simulator.**

Learning Outcomes:

- **Learned how to use millis() for time-based logic.**
- **Understood how to detect press duration using software logic.**
- **Learned how to display real-time messages on the OLED.**
- **Practiced using debouncing techniques and timing conditions.**

Conclusion:

**This project shows how one button can perform multiple tasks based on press duration.
It is a practical concept used in IoT systems, automation controls, and menu-based devices.
The system works efficiently and provides clear visual and sound feedback through the OLED and buzzer.**