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:
To design an ESP32-based system that can differentiate between short and long button presses, and perform separate actions for each type. The feedback of the press type and output status should be displayed on an OLED screen.

Introduction:
In embedded systems, different input durations can be used to perform multiple tasks using a single button.
This project implements such logic by measuring how long the button is pressed.
A short press toggles an LED, while a long press turns ON a buzzer.
An OLED display is used to give the user immediate visual feedback about the type of press detected.

Hardware Components Required:

1. ESP32 DevKitC V4
2. OLED Display (128x64 I2C)
3. Push Button × 1
4. LED × 1
5. Buzzer × 1
6. Resistor × 1 (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
- Adafruit_SSD1306 and Adafruit_GFX libraries for OLED display
- Wokwi Simulator for virtual testing

Working Principle:

1. The ESP32 constantly monitors the button input.
2. When the button is pressed, the current time is stored using millis().
3. When the button is released, the code checks how long it was pressed.
4. If the press duration is less than 1 second, it is counted as a short press.
   → LED toggles ON or OFF.
5. If the press duration is greater than 1.5 seconds, it is a long press.
   → The buzzer turns ON for a short time.
6. The OLED shows messages such as “Short Press Detected” or “Long Press Detected.”
7. The system then resets and waits for the next input.

Code Explanation:

- millis() is used to calculate time differences between button press and release.
- The logic differentiates actions using if-else statements.
- digitalWrite() controls LED and buzzer states.
- OLED messages are updated dynamically using Adafruit functions.
- Debounce delay ensures the button press is accurately detected.
- Serial Monitor is used for debugging and confirmation of press duration.

Testing Steps:

1. Upload the program to ESP32.
2. Open Serial Monitor to check message output.
3. Press and quickly release the button – the LED toggles.
4. Press and hold the button for more than 1.5 seconds – the buzzer activates.
5. OLED displays the corresponding message for each press type.
6. Repeat multiple times to confirm accuracy.

Results:

- The ESP32 accurately detects both short and long presses.
- The OLED display provides clear feedback for each type of press.
- The LED and buzzer operate without delay or false triggers.
- The system works properly both in hardware and in Wokwi simulation.

Wokwi Simulation Link:
https://wokwi.com/projects/445776781415259137

Learning Outcomes:

- Learned to use the millis() function for real-time event measurement.
- Understood timing-based input detection and logic implementation.
- Practiced displaying system feedback through an OLED screen.
- Developed better understanding of button debouncing and condition handling.

Conclusion:
This project successfully implements a dual-function button using timing logic.
It is a great example of how a single input can control multiple outputs depending on press duration.

The concept is useful in IoT-based control panels, automation menus, and user interface systems where different press actions are required.