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 1 – Multi-Device Control using ESP32 (LEDs, Buttons, Buzzer & OLED)

Objective:

The main objective of this task is to design a simple hardware control system using ESP32 that can manage multiple devices (LEDs and a buzzer) through button inputs. The system should also display real-time feedback on an OLED screen to enhance user interaction.

Introduction:

In this project, the ESP32 microcontroller is used to control three LEDs and one buzzer

using two push buttons.
An OLED display is interfaced through I2C communication to show the current status of devices.
The aim is to understand the working of digital input/output pins, logic control, and display interfacing.
Such systems are commonly used in smart control panels, IoT dashboards, and device monitoring applications.

Hardware Components Required:

1. ESP32 DevKitC V4 – microcontroller board
2. OLED Display (128x64 I2C) – for display output
3. LEDs (Red, Green, Blue) – 3 pieces
4. Push Buttons – 2 pieces
5. Buzzer – 1 piece
6. Resistors (420 ohms) – 3 pieces
7. Breadboard and jumper wires – for connections

Pin Connections:
LED1 → GPIO 2
LED2 → GPIO 4
LED3 → GPIO 5
Button1 → GPIO 26
Button2 → GPIO 27
Buzzer → GPIO 15
OLED SDA → GPIO 21
OLED SCL → GPIO 22

Software and Libraries Used:

- Arduino IDE for programming and uploading code
- Adafruit_SSD1306 and Adafruit_GFX libraries for OLED display

- Wokwi online simulator for testing the circuit virtually

Working Principle:

1. Two push buttons are connected to the ESP32 inputs.
2. Button 1 is used to control LED1 and LED2. When pressed, these LEDs toggle their state (ON/OFF).
3. Button 2 controls LED3 and the buzzer. Pressing it toggles their state.
4. The OLED display is continuously updated to show which components are ON or OFF.
5. The main loop checks button states using digitalRead() function.
6. When a button is pressed, it changes the output pin's logic using digitalWrite().
7. Debouncing is handled using a short delay or state check.
8. The OLED screen displays messages like "LED1 ON", "Buzzer OFF", etc.

Code Explanation:

1. The Adafruit libraries are initialized to communicate with the OLED.
2. All output devices (LEDs and buzzer) are set as OUTPUT pins.
3. Buttons are configured with INPUT_PULLUP to avoid floating states.
4. Inside the loop, the ESP32 continuously reads both button states.
5. If Button 1 is pressed, the program toggles LED1 and LED2.
6. If Button 2 is pressed, LED3 and buzzer toggle.
7. The OLED is cleared each time and updated to show the latest states.
8. Display messages are shown using functions like display.setCursor(), display.print(), and display.display().

Testing Steps:

1. Connect all components according to the pin map.

2. Upload the code to ESP32 using Arduino IDE.
3. Open Serial Monitor at 115200 baud to observe debugging messages.
4. Press Button 1 and Button 2 separately.
5. Observe LEDs and buzzer switching ON and OFF.
6. Check that the OLED screen updates immediately to reflect current status.
7. Repeat multiple times to ensure stable performance without false triggering.

Results:

- The LEDs and buzzer respond correctly to button presses.
- The OLED display shows real-time device states.
- The circuit operates smoothly both on hardware and Wokwi simulation.

Wokwi Simulation Link:
https://wokwi.com/projects/445223337931397121

Learning Outcomes:

- Learned how to use GPIO pins as input and output on ESP32.
- Understood the use of pull-up resistors with push buttons.
- Implemented real-time status updates using an OLED display.
- Gained experience in interfacing multiple components together.

Conclusion:
This project demonstrates the practical working of an ESP32-based control system where multiple devices can be managed simultaneously using button inputs. It improves the understanding of embedded system interaction and provides a base for smart automation projects.