

Assignment 1 — LED and Button Control with OLED Display

Name: Noor Fatima

Registration Number: 23-NTU-CS-1083

Course: Embedded Systems / Microcontroller Programming

Date: October 2025

1. Title

LED Mode Controller and Single Button Press-Type Detection with OLED Display

2. Objective

- Implement a microcontroller-based system using ESP32 to control LEDs, a buzzer, and an OLED display with push buttons.
 - **Demonstrate two major functions:**
 1. Cycle through multiple LED modes using one button and reset using a second button.
 2. Detect short and long presses with a single button and provide corresponding feedback.
-

3. Components Required

Component	Quantity	Description
ESP32 Board	1	Microcontroller
LEDs	3	LED1, LED2 (Mode indication), LED3 (PWM fade)
Push Buttons	2	Button 1 → Mode cycle, Button 2 → Reset
Active Buzzer	1	Audio feedback for button events
OLED Display (128x64)	1	Visual feedback for modes and events
Jumper Wires	—	Connections between components
Breadboard	1	For easy prototyping

4. Circuit Diagram

4.1 Connections Table (Task A)

Component	ESP32 Pin	Description
LED1	GPIO 2	Mode indication LED

Component	ESP32 Pin	Description
LED2	GPIO 4	Mode indication LED
LED3	GPIO 5	PWM fade LED
BUZZER	GPIO 27	Active buzzer
BTN_MODE	GPIO 12	Mode cycle button
BTN_RESET	GPIO 14	Reset button
OLED SDA	GPIO 21	I2C data
OLED SCL	GPIO 22	I2C clock

4.2 Connections Table (Task B)

Component	ESP32 Pin	Description
LED	GPIO 2	Toggle LED
BUZZER	GPIO 27	Play tone
BUTTON	GPIO 12	Single button input
OLED SDA	GPIO 21	I2C data

Component	ESP32 Pin	Description
OLED SCL	GPIO 22	I2C clock

4.3 Wokwi Simulation

- JSON file [Wokwi - Online ESP32, STM32, Arduino Simulator](#)
 - Simulates LEDs, PWM fading, OLED messages, and buzzer feedback.
-

5. Functional Description

5.1 Task A — LED Mode Controller (Two Buttons)

❖ MODE Button (GPIO 12):

- Cycles through 4 LED modes:
 1. Mode 0 → Both LEDs OFF
 2. Mode 1 → Alternate Blink
 3. Mode 2 → Both ON
 4. Mode 3 → PWM Fade on LED3
- Displays mode number on OLED.
- Buzzer provides short feedback.

❖ RESET Button (GPIO 14):

- Resets mode to 0 (All OFF).
- OLED displays “Reset → Mode 0: OFF”.

- Buzzer plays a lower tone.

❖ Technical Details:

- Interrupts: Used for button press detection.
 - Software Debounce: Prevents false triggering.
 - PWM: Used to fade LED3 smoothly.
 - OLED Display: Updates real-time mode messages.
-

5.2 Task B — Single Button Press-Type Detection

❖ Button Actions (GPIO 12):

- Short Press ($<1.5s$): Toggle LED ON/OFF.
- Long Press ($\geq 1.5s$): Play buzzer tone for 0.6s.

❖ OLED Feedback:

- Displays “Short Press → LED ON/OFF” or “Long Press → Buzzer”.

❖ Logic:

- Uses `millis()` for timing button presses.
 - Includes small delay for debouncing.
 - Serial monitor prints event logs.
-

6. Software Design & Algorithm

6.1 Task A — LED Mode Controller

- ✚ Initialize GPIO pins for LEDs, buzzer, and buttons.
- ✚ Attach interrupts to MODE and RESET buttons.
- ✚ On MODE press → increment mode counter (0–3).
- ✚ On RESET press → reset to mode 0.
- ✚ Update OLED with the current mode.
- ✚ Play buzzer feedback tone.
- ✚ Control LEDs according to selected mode:
 - Mode 0: All OFF
 - Mode 1: Alternate blinking every 400 ms
 - Mode 2: Both ON
 - Mode 3: PWM fade using `analogWrite()`
- ✚ Use non-blocking timing for smooth operation.




6.2 Task B — Single Button Press Detection

- ✚ Initialize LED, buzzer, button, and OLED.
 - ✚ Detect button state changes.
 - ✚ Measure duration using `millis()`.
 - ✚ $<1.5\text{s}$ → Toggle LED
 - ✚ $\geq 1.5\text{s}$ → Play buzzer tone
 - ✚ Update OLED and Serial Monitor with feedback.
-

7. Code Structure

Section	Description
Pin Definitions	Define LED, buzzer, and button pins
Variables	Store LED state, press timing, debounce flags
OLED Function	showMessage() displays feedback text
Interrupts	Trigger on falling edge with debounce logic
Setup()	Initializes pins, OLED, and serial monitor
Loop()	Handles button logic, LED states, and OLED updates

Libraries Used:

-  Arduino.h — Core functions
-  Wire.h — I²C communication
-  Adafruit_GFX.h and Adafruit_SSD1306.h — OLED graphics

8. Observations / Results


Task A – LED Mode Controller

Mode	LED Behavior	OLED Display
0	All OFF	“Mode 0: Both OFF”
1	Alternate blink	“Mode 1: Alternate Blink”

Mode	LED Behavior	OLED Display
2	Both ON	“Mode 2: Both ON”
3	PWM Fade	“Mode 3: PWM Fade”

Task B – Single Button Detection

Press Type	Output	OLED Message
Short (<1.5s)	LED toggles ON/OFF	“Short Press → LED ON/OFF”
Long (≥1.5s)	Buzzer tone (0.6s)	“Long Press → Buzzer”

 All test cases matched expected behavior.

9. Wokwi Simulations

- Task A: <https://wokwi.com/projects/445853794784446465>
- Task B: <https://wokwi.com/projects/445858202457393153>

10. Conclusion

- Successfully implemented LED and button control using ESP32, buzzer, and OLED.
- Task A demonstrated multi-mode LED control with interrupts and PWM fading.

- Task B demonstrated press-type detection using a single button and timing logic.
 - OLED provides real-time visual feedback.
 - Meets all embedded systems objectives and shows understanding of interrupts, debouncing, and event-driven programming.
-

12. References

- Wokwi Simulator: <https://wokwi.com>
- Adafruit GFX & SSD1306 Libraries Documentation
- ESP32 Technical Reference Manual