# 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 |

## Description

# Task_1

**Multi-Device Control using ESP32 (LEDs, Buttons, OLED, and Buzzer)**

**Description:**

In this task, an **ESP32 DevKit-C V4** is programmed to control **three LEDs**, **two push buttons**, a **buzzer**, and an **OLED display** using the **Adafruit SSD1306** library.

**Working:**

- Each **button press** performs a specific action — such as toggling LEDs or turning ON the buzzer.
- The **OLED display** shows clear text messages for user feedback like “LED ON”, “LED OFF”, or “Buzzer Active”.
- LEDs are connected to **GPIO 2, 4, and 5**, the buzzer to **GPIO 15**, and buttons to **GPIO 26 and 27** with internal pull-ups.
- The **I$^2$C OLED** uses **GPIO 21 (SDA)** and **GPIO 22 (SCL)** with address 0x3C.
- Proper resistors (420Ω) are used with LEDs to limit current.
- The system was simulated on **Wokwi** to test circuit behavior.

**Time & Execution:**

On pressing a button, the ESP32 reads the input, updates the respective LED/buzzer state, and instantly displays the message on the OLED. The whole response time is **less than 1 second**, ensuring real-time feedback.

**Objective:**

To learn how to interface multiple input/output devices with ESP32 and display real-time feedback using an OLED display.

# Task_2

This task demonstrates **button press duration detection** using an **ESP32**, differentiating between **short** and **long presses**.

The system includes a **push button**, **LED**, **buzzer**, and **OLED display** for feedback.

**Working:**

- When the button (GPIO 25) is pressed and released, the code calculates how long it was held using millis().

- If the press duration is **less than 1.5 seconds**, it is treated as a **short press**, and the **LED (GPIO 5)** toggles its state.
- If the press is **longer than 1.5 seconds**, a **buzzer (GPIO 18)** activates for 0.5 seconds to indicate a **long press**.
- The **OLED (I$^2$C: SDA 21, SCL 22)** displays messages like "Short Press → LED Toggle" or "Long Press → Buzzer".
- The system resets automatically after each press for the next detection.

**Time & Execution:**

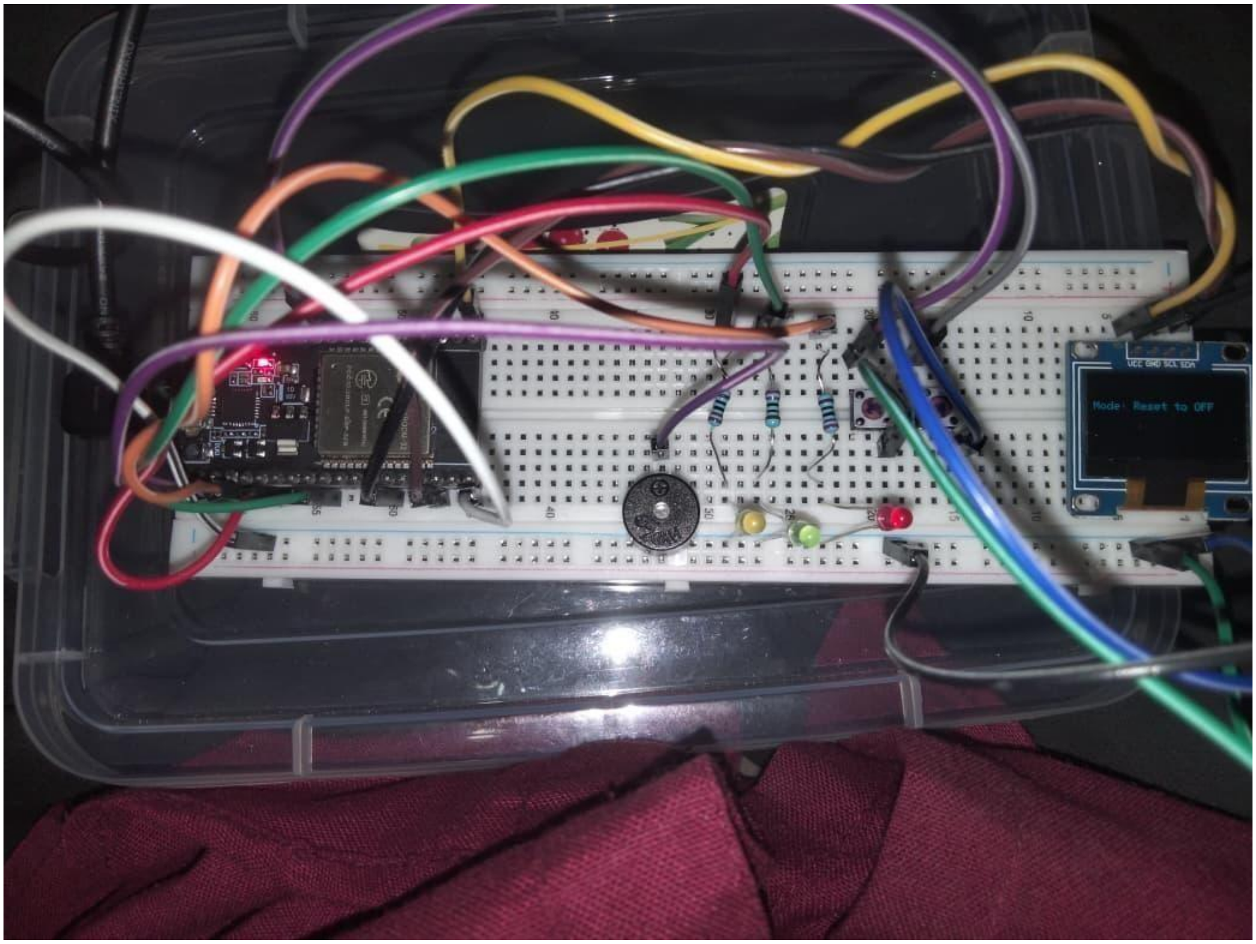
- Short press duration: **< 1.5 seconds**
- Long press duration: **> 1.5 seconds**
- OLED updates instantly after each event with clear text.

**Objective:**

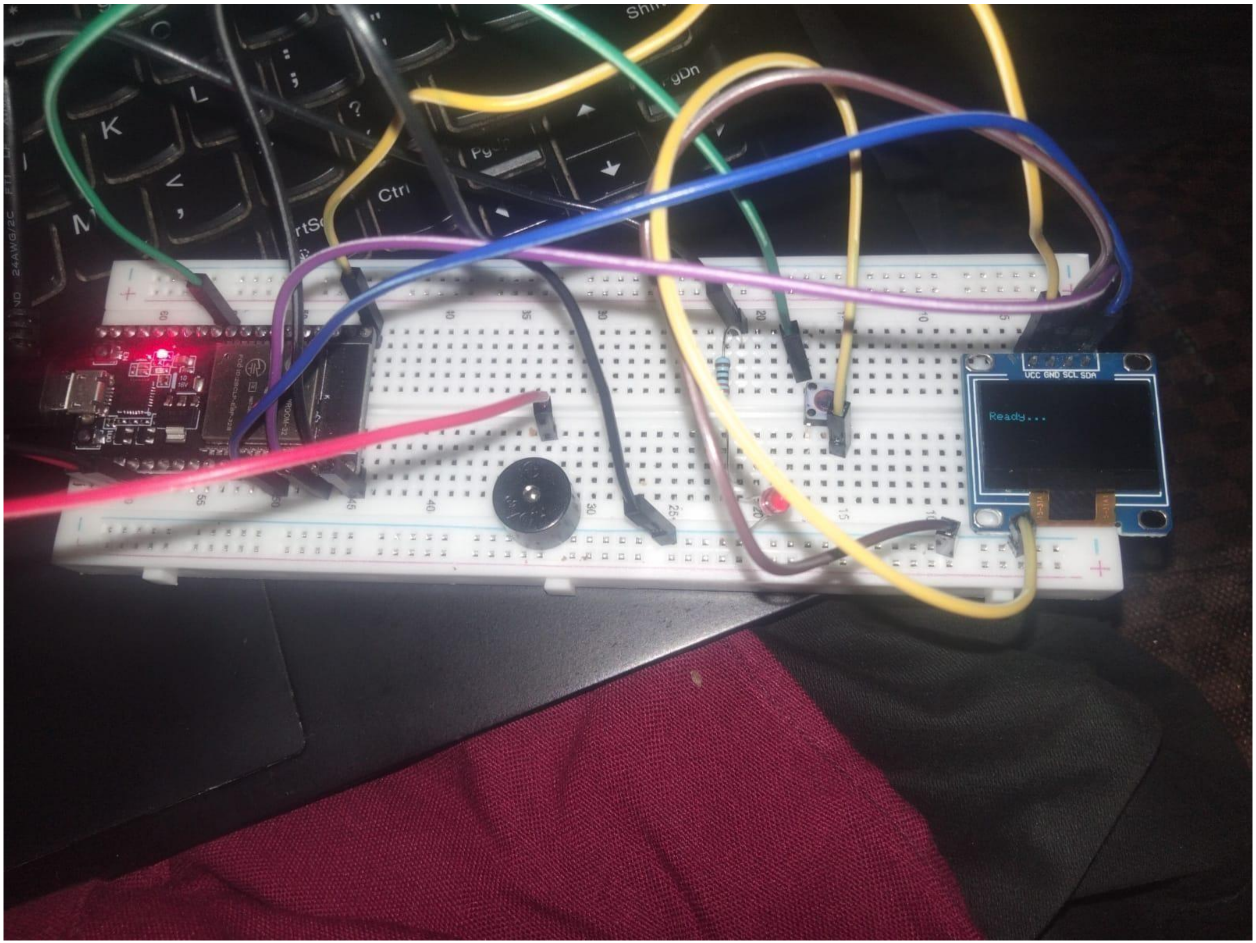
To understand how to measure input duration using **millis()**, differentiate user input types, and provide feedback through both **LEDs**, **buzzer**, and **display**.

## Picture of Kits

Task_01



Task_02



# Handwritten Code

Task_01

Assignment #1

Task#1 code:

```
/* Project: LED Mode controller with OLED
   and buzzer
   Name: Umar Mushtaq
   Reg No: 23-NTU-CS-1077
*/

#include <Ardinuo.h>
# include <wire.h>
# include <Adafruit_GFX.h>
# include <Adafruit_SSD1306>

// --- Pin definations ---
#define LED1 2
#define LED2 4
#define LED3 5
#define BTN_MODE 26
#define BTN-RESET 27
#define BUZZER 15
```

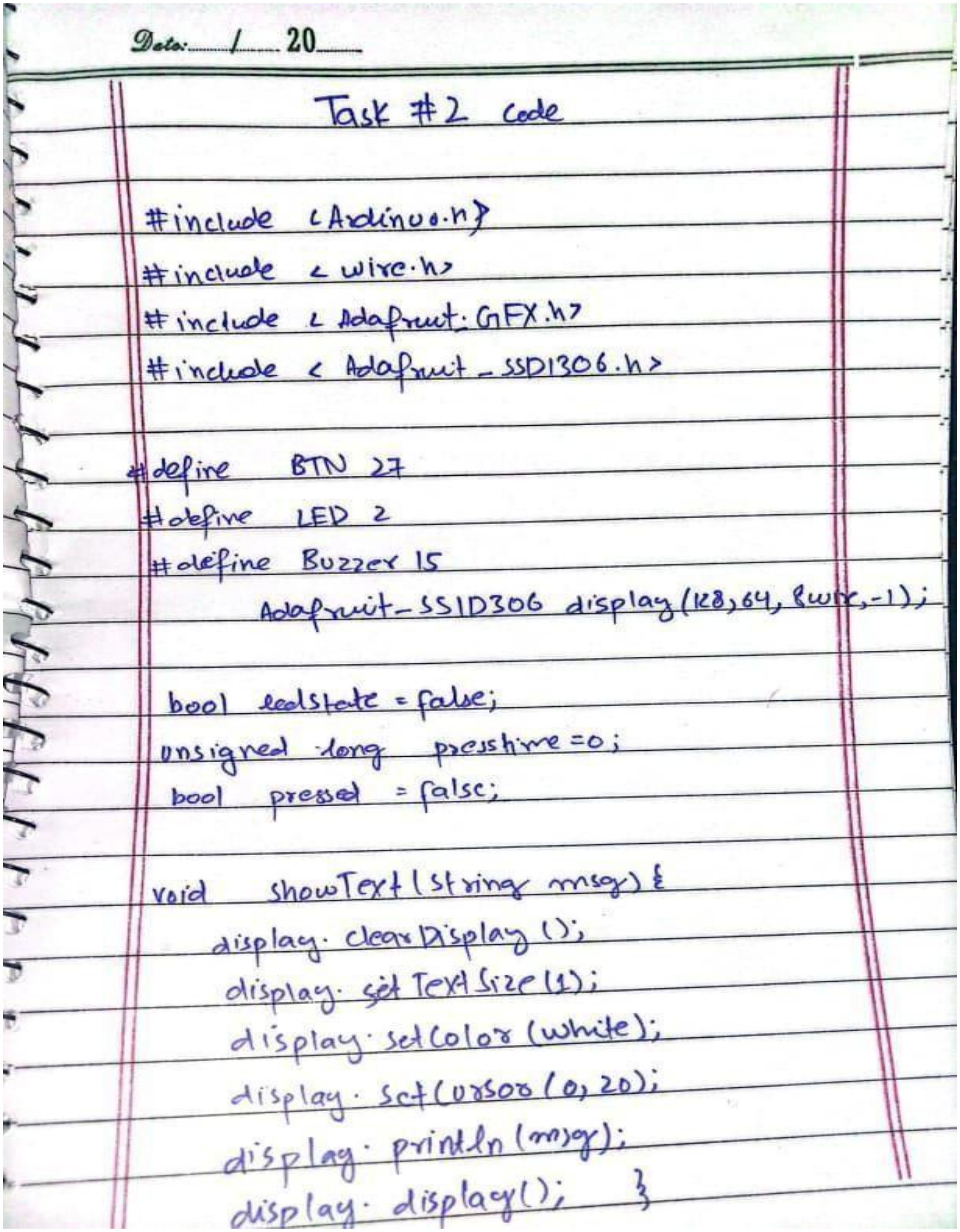
```
// --- object Display on OLED---
Adafruit_SSD1306 display(128, 64, &wire, -1)

// --- variables for mode controll ---
int mode = 0;
unsigned long PreMillis = 0;
bool ledstate = false;

// --- Function: show massage on OLED
void showmsg(string msg){
      display.clearDisplay();
      display.setText(1);
      display.setColor(white);
      display.setCursor(0, 20);
      display.Print("Mode: ");
      display.Println(msg);
      display.display();
}

//- -- function beep sound ---
void beepBuzzer(int freq, int dur){
    tone(Buzzer, freq, dur);
    delay(dur + 50);
    noTone(Buzzer); }
```

```
void setup() {
        Pinmode(LED1,outPut)
        PinMode(LED2, output)
        Pin Mode(LED3, output)
        PinMode(BTN-Mode, output)
        Pin Mode(BTN_RESET, output)
        PinMode(Buzzer, output).

 display.begin(SS1306_SWITCHCAPAVCC, 0X3C)
 display. cleardisplay();
 display.display();
  showmsg("Both off")
 void  loop() {
     if  digitalRead(BTN_Mode)== Low){
     delay(200);
      mode++;
       if(mode>4) mode=1)
 switch (mode) {
     case 1:
       digitalWrite(LED1, Low);
       digital write (LED 2, LOW);
        Showmsg("Both OFF");
        beepBuzzer(800,120);
```

```
break
case 2;
    showMsg("Alternate Blink");
    beepBuzzer(1000, 120);
    break;
case 3;
     digitalWrite (LED1, HIGH)
     digitalWrite(LED2, HiGH)
      showMsg ("Both ON");
      beep Buzzer (1200,120);
      break;
case 4;
       showMsg ("PWM Fade");
       BeepBuzzer ( 1500, 120);
       break;
     }
}
if (digitalRead(BTN_RESET)= = low {
    delay(200);
     mode = 1;
    digitalWrite (LED1, Low);
    digitalWrite (LED2, LOW);
    analogewrite (LED3, 0);
```

```
void setup(){
        PinMode (BTN, INPUT_PULLUP);
        PinMode(LED, OUTPUT);
        Pin Mode (Buzzer, OUTPUT)
display.begin(SSD1306_SWITCHCAVCC, 0X3c)
  showText("Ready..."); }
 void loop() {
 if (digitalRead (BTN)== low 88 ! pressed){
   pressed = true;
    presstime = millis (); }
If (digital Read (BTN)== HIGH 88 pressed){
  unsigned long duration= millis()-presstime;
  pressed = false;
   if (duration > 1500){
    tone (Buzzer, 1000, 500),
    Show text ("long press -> Buzzer"));

else
    ledState = ! state
    digitalWrite (LED, ledstate);
    Show text("short press -> LED
                      toggle),
  }
 }
```

Task_02



Date:____/____ 20____

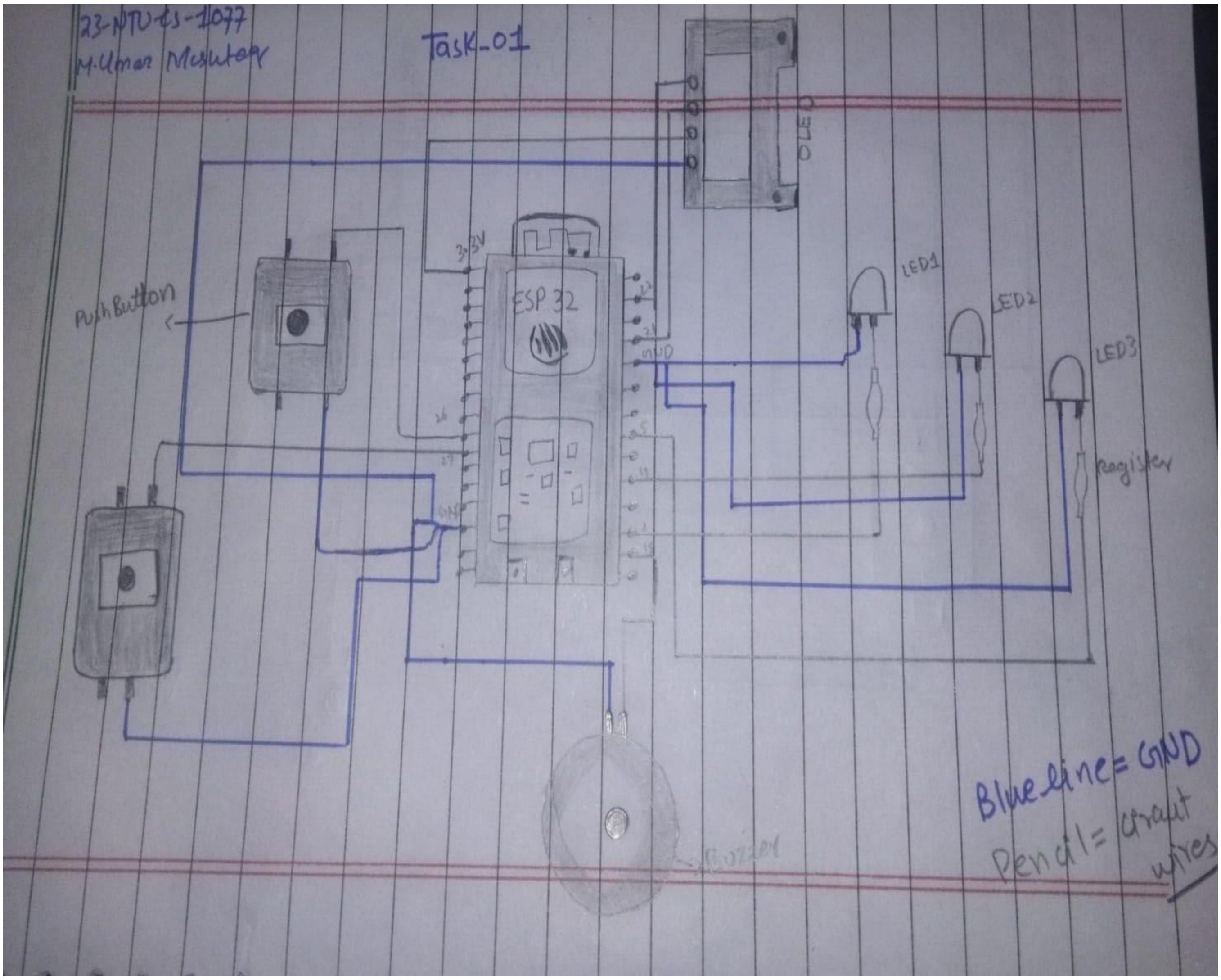
Task #2 code

```
#include <Ardinuo.h>
#include <wire.h>
#include <Adafruit_GFX.h>
#include <Adafruit_SSD1306.h>

#define BTN 27
#define LED 2
#define Buzzer 15
        Adafruit_SSID306 display(128,64,&wire,-1);

bool ledstate = false;
unsigned long presstime =0;
bool pressed = false;

void showText(string msg){
    display.clearDisplay();
    display.setTextSize(1);
    display.setColor(white);
    display.setCursor(0,20);
    display.println(msg);
    display.display();   }
```

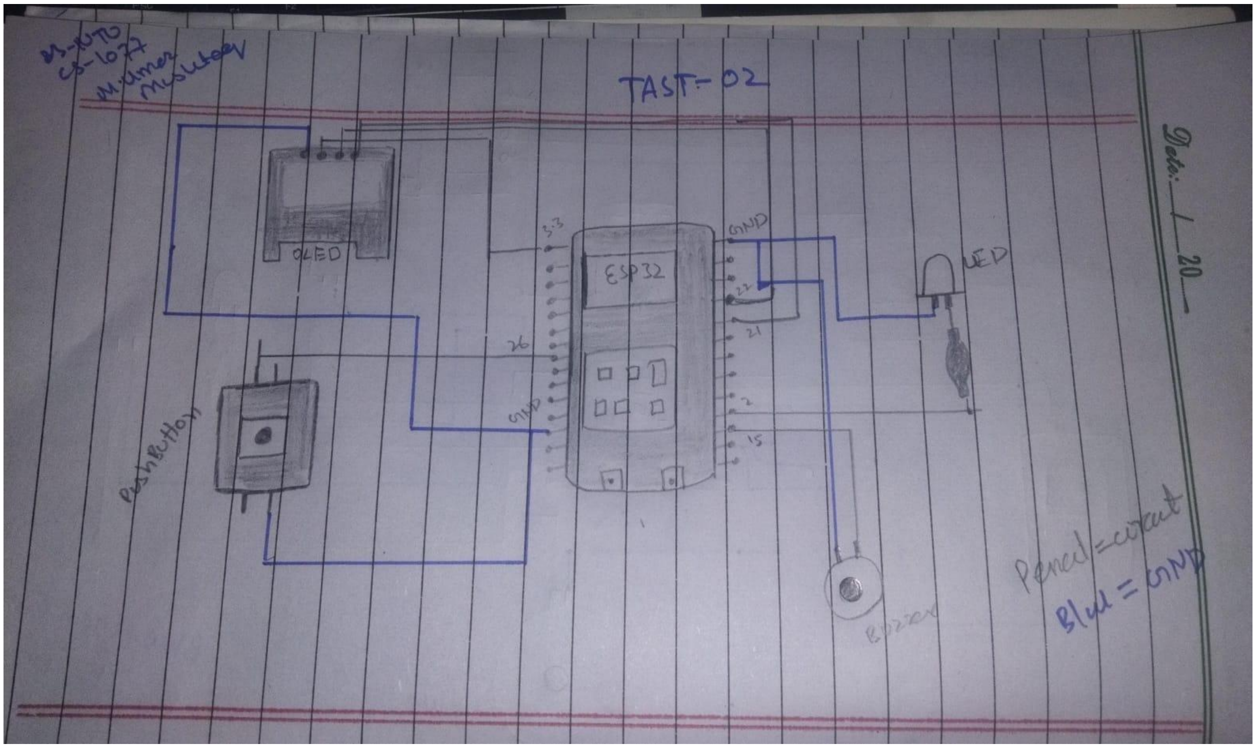
```
showMsg("Reset to OFF");
beepBuzzer(400,200); }
 if (mode==2){
   if (millis)() - preMillis >=500){
   preMillis = millis();
   ledstate = !ledstate;
   digitalwrite (LED1, ledstate);
   digitalwrite(LED2, !ledstate); }
 }
 if (mode= =4){
   for (int i=0 ; i<=255; i++){
    analogwrite(LED3, i);
    delay (5);
 }
 for (int i =0; i=255; i--; ){
     analogWrite (LED3, i);
     delay (5);
   }
  }
}
```

# Handmade Diagram

Task_01



Task_02



## Wokwi Link

Task_01 https://wokwi.com/projects/445898721187270657

Task_02 https://wokwi.com/projects/445899618251465729

# Github Repo Link

https://github.com/noniiiiiiiiiiiiiiii/1077embeddediot-system-cs-b.git