National Textile University, Faisalabad

Department of Computer Science

Assignment # 1

Name	Rameen Fatima
Section	BSCS-B
Semester	5th
Registration no.	23-NTU-CS-1086
Course title	Embedded IOT systems
Submitted to	Sir Nasir
Submission date	23-10-2025

Documentation for Task B

Question 3

ESP32 Multi-LED Control with Press-Type Detection

Circuit Diagram Description:

This project uses an ESP32 microcontroller to detect short and long button presses and perform different actions: toggle LEDs or play a buzzer tone. The system displays the event in real time on an OLED (SSD1306) display.

Component List:

Component Name	Quantity	Description
ESP32 Dev Board	1	Main microcontroller
Push Button	1	Used for short/long press detection
LED (Red, Green, Blue)	3	Visual indication
Buzzer	1	Audio feedback
OLED Display (SSD1306)	1	0.96" I2C display for messages
Jumper Wires	As required	For circuit connections
Breadboard	1	Used for circuit prototyping

Task Explanation:

This project demonstrates how to control multiple LEDs and a buzzer using a single button with an OLED display showing live feedback.

Features:

- Detects short and long button presses.
- Displays messages on OLED (SSD1306).
- Provides both visual (LED) and audio (buzzer) feedback.
- Uses non-blocking timing with millis () for accurate press detection.
- Includes OLED header display with real-time updates.

Working Principle:

1. Button Input Detection
    - The button is connected using the internal pull-up resistor.
    - When pressed, it pulls the input LOW.
    - The ESP32 measures press duration using millis().
2. Press-Type Decision
    - Short Press (<1.5 sec): Toggles all LEDs (ON/OFF).
    - Long Press (>1.5 sec): Activates the buzzer tone for 0.5 seconds.
3. LED and Buzzer Actions
    - LEDs are connected to GPIO 25, 27, and 33.
    - Buzzer is connected to GPIO 26 for tone generation.
4. OLED Display Feedback
    - Displays header: "PRESS-TYPE DETECT".
    - Shows messages such as:
        - “Short-Press Detected — LEDs: ON/OFF”
        - “Long-Press Detected — Playing Buzzer...”

Pin Diagram		
Component	**GPIO Pin**	**Description**
Push Button	14	Input with Pull-up
LED 1	25	Output
LED 2	27	Output
LED 3	33	Output
Buzzer	26	PWM Output
OLED SDA	21	I2C Data
OLED SCL	22	I2C Clock

CODE:

```
/*
  ------------------------------------------------------------
  Title       : ESP32 Multi-LED Control with Press-Type Detection
  Author      : Rameen Fatima
  Reg. No.    : 23-NTU-CS-1086 (BSCS-5TH B)
  Description :
      This program detects short and long presses of a single button.
      - Short Press (<1.5 sec): Toggles 3 LEDs ON/OFF
      - Long Press (>1.5 sec): Plays a buzzer tone
      The event is displayed on the OLED (SSD1306) with a header.

      Components:
        • 1 Push Button (GPIO 14)
        • 3 LEDs (GPIO 25, 26, 27)
        • 1 Buzzer (GPIO 33)
        • 1 OLED Display (SDA=21, SCL=22)
  ------------------------------------------------------------
*/

#include <Wire.h>
#include <Adafruit_GFX.h>
#include <Adafruit_SSD1306.h>

// --- OLED Configuration ---
#define SCREEN_WIDTH 128
#define SCREEN_HEIGHT 64
#define OLED_ADDR 0x3C
Adafruit_SSD1306 display(SCREEN_WIDTH, SCREEN_HEIGHT, &Wire, -1);
```

```
// --- Pin Configuration ---
#define LED1 25
#define LED2 27
#define LED3 33
#define BUZZER_PIN 26
#define BUTTON_PIN 14

// --- Variables ---
bool ledState = false;              // Stores ON/OFF state for all LEDs
unsigned long pressStart = 0;       // Time when button was pressed
bool isPressed = false;             // Flag to track button press
const unsigned long longPressTime = 1500;  // 1.5s threshold for long press

// --- Function to display header and message on OLED ---
void showOnOLED(const String &line1, const String &line2 = "") {
  display.clearDisplay();

  // Header area
  display.setTextSize(1);
  display.setTextColor(SSD1306_WHITE);
  display.setCursor(10, 0);
  display.println("PRESS-TYPE DETECT");
  display.drawLine(0, 10, 127, 10, SSD1306_WHITE); // underline
```

```
  // Main message
  display.setCursor(10, 25);
  display.println(line1);
  if (line2 != "") {
    display.setCursor(10, 40);
    display.println(line2);
  }

  display.display();
}

void setup() {
  Serial.begin(115200);

  // --- Pin Modes ---
  pinMode(LED1, OUTPUT);
  pinMode(LED2, OUTPUT);
  pinMode(LED3, OUTPUT);
  pinMode(BUZZER_PIN, OUTPUT);
  pinMode(BUTTON_PIN, INPUT_PULLUP);  // Button active LOW

  // --- Initialize OLED ---
  Wire.begin(21, 22);
  if (!display.begin(SSD1306_SWITCHCAPVCC, OLED_ADDR)) {
    Serial.println("OLED not found!");
    while (true);
  }
```

```
  // --- Startup Screen ---
  showOnOLED("System Ready...", "Press Button");
  delay(1500);

  // Turn off LEDs and Buzzer initially
  digitalWrite(LED1, LOW);
  digitalWrite(LED2, LOW);
  digitalWrite(LED3, LOW);
  digitalWrite(BUZZER_PIN, LOW);
}

void loop() {
  int buttonState = digitalRead(BUTTON_PIN);

  // --- Detect button press start ---
  if (buttonState == LOW && !isPressed) {
    isPressed = true;
    pressStart = millis();
  }

  // --- Detect button release ---
  if (buttonState == HIGH && isPressed) {
    unsigned long pressDuration = millis() - pressStart;
    isPressed = false;
```

```
    if (pressDuration < longPressTime) {
      // --- SHORT PRESS ACTION: Toggle LEDs ---
      ledState = !ledState;
      digitalWrite(LED1, ledState);
      digitalWrite(LED2, ledState);
      digitalWrite(LED3, ledState);

      showOnOLED("Short-Press Detected", ledState ? "LEDs: ON" : "LEDs: OFF");
      Serial.println(ledState ? "LEDs ON" : "LEDs OFF");

    } else {
      // --- LONG PRESS ACTION: Buzzer Tone ---
      showOnOLED("Long-Press Detected", "Playing Buzzer...");
      tone(BUZZER_PIN, 1000, 500);  // 1kHz for 0.5s
      Serial.println("Buzzer Tone Played");
    }

    delay(300); // Small delay to avoid flicker
  }
}
```

Build success:

diagram.json U | diagram.json U | main.cpp U | PIO Home | platformio.ini U | wokwi.toml U

src > main.cpp > loop()

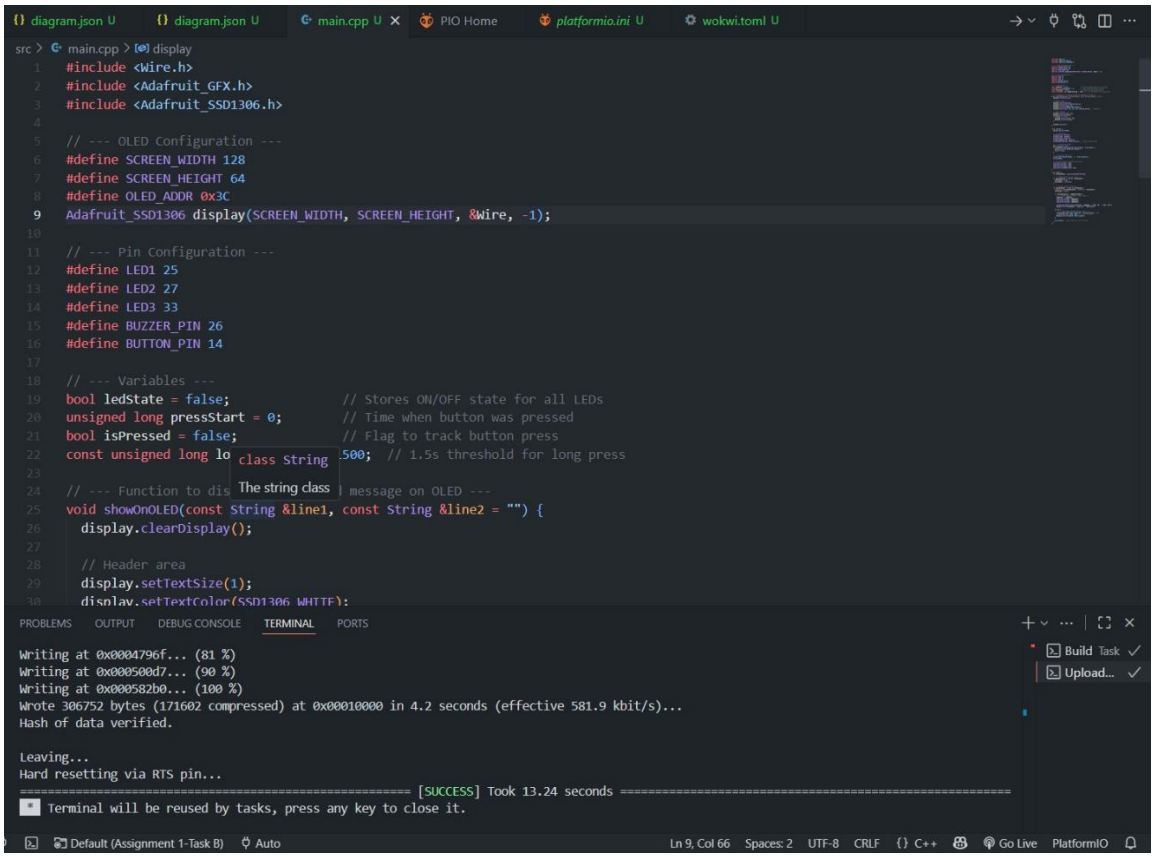
```
#include <Wire.h>
#include <Adafruit_GFX.h>
#include <Adafruit_SSD1306.h>

// --- OLED Configuration ---
#define SCREEN_WIDTH 128
#define SCREEN_HEIGHT 64
#define OLED_ADDR 0x3C
Adafruit_SSD1306 display(SCREEN_WIDTH, SCREEN_HEIGHT, &Wire, -1);

// --- Pin Configuration ---
#define LED1 25
#define LED2 27
#define LED3 33
#define BUZZER_PIN 26
#define BUTTON_PIN 14

// --- Variables ---
bool ledState = false;              // Stores ON/OFF state for all LEDs
unsigned long pressStart = 0;       // Time when button was pressed
bool isPressed = false;             // Flag to track button press
const unsigned long longPressTime = 1500;  // 1.5s threshold for long press

// --- Function to display header and message on OLED ---
void showOnOLED(const String &line1, const String &line2 = "") {
  display.clearDisplay();

  // Header area
  display.setTextSize(1);
  display.setTextColor(SSD1306_WHITE);
```

PROBLEMS | OUTPUT | DEBUG CONSOLE | TERMINAL | PORTS | Build - Task

```
|-- Adafruit SSD1306 @ 2.5.15
|-- Wire @ 2.0.0
Building in release mode
Retrieving maximum program size .pio\build\nodemcu-32s\firmware.elf
Checking size .pio\build\nodemcu-32s\firmware.elf
Advanced Memory Usage is available via "PlatformIO Home > Project Inspect"
RAM:   [=         ]   6.7% (used 22088 bytes from 327680 bytes)
Flash: [==        ]  23.4% (used 306381 bytes from 1310720 bytes)
========================================================================= [SUCCESS] Took 5.76 seconds =========================================================================
* Terminal will be reused by tasks, press any key to close it.
```

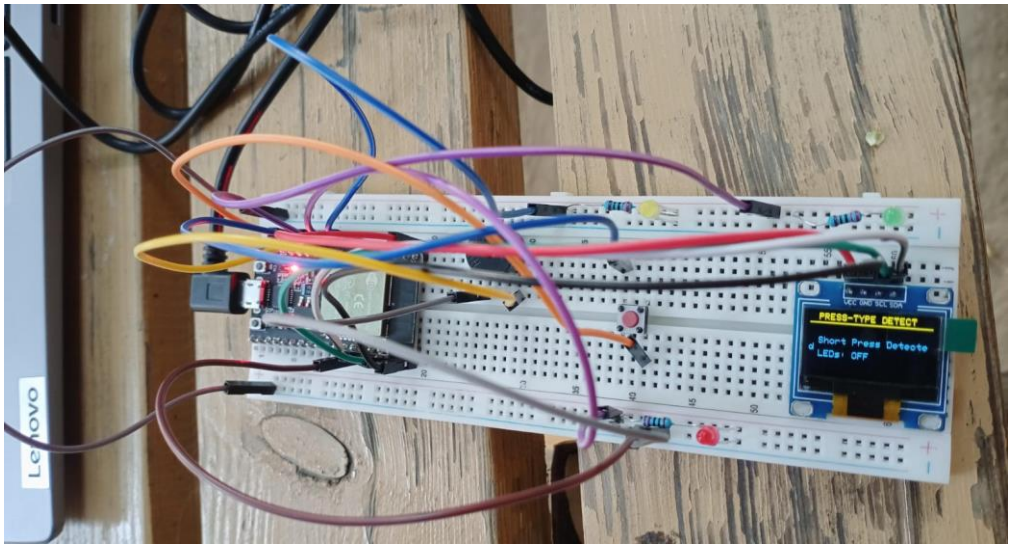
Default (Assignment 1-Task B) | Auto | Ln 107, Col 2 | Spaces: 2 | UTF-8 | CRLF | {} C++ | Go Live | PlatformIO

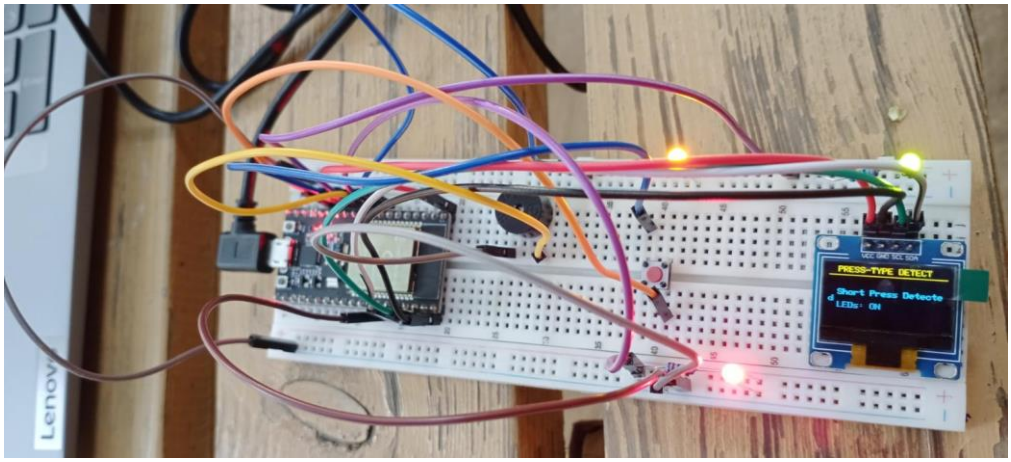
Upload success:



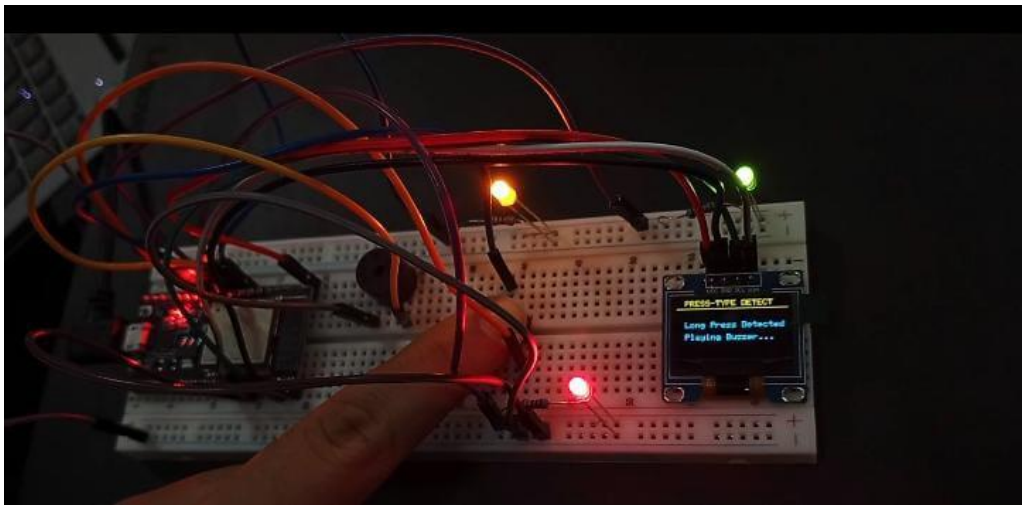
Hardware output:

Short press (Toggle LED):





Long Press (Play Buzzer tone):



Handwritten code:

Task B

Multi-LED Control with Press-Type Detection

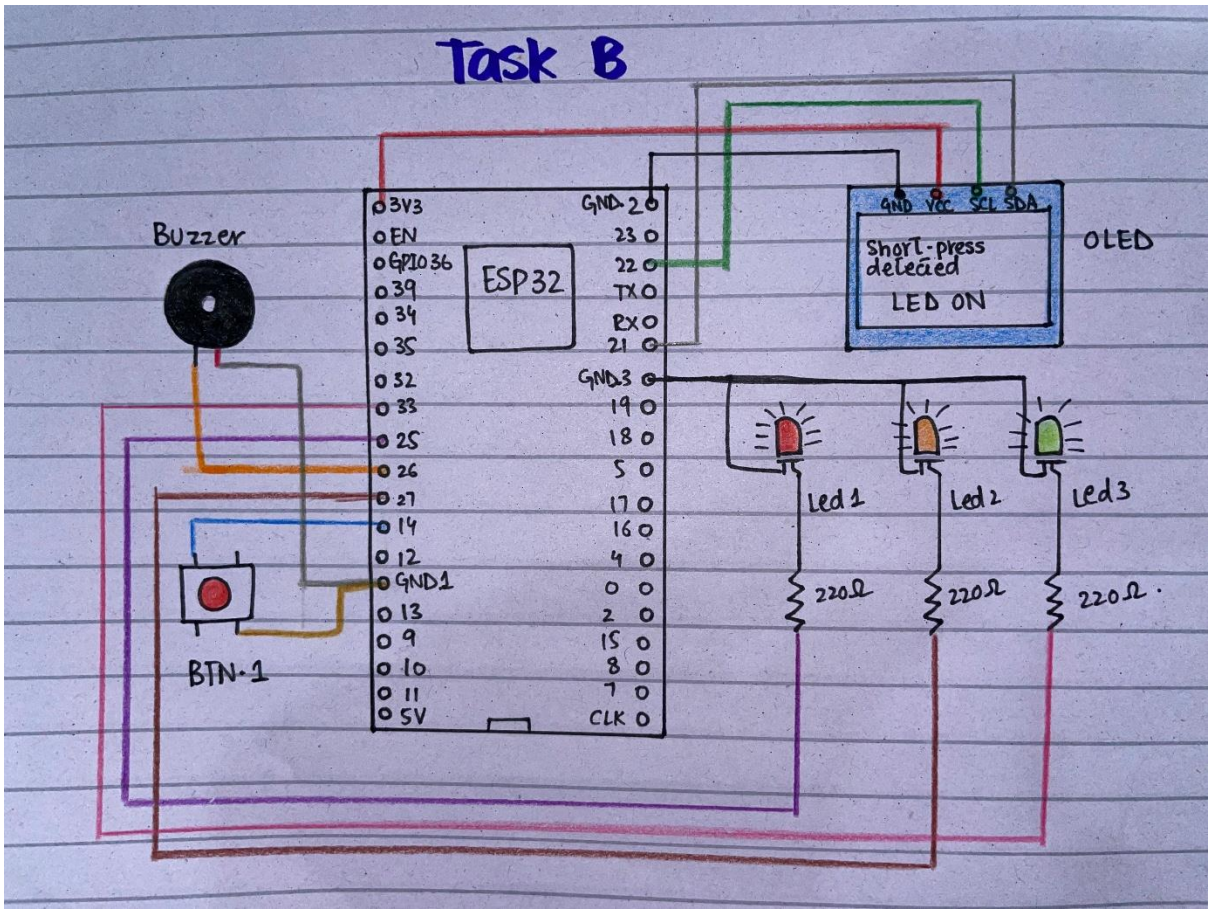
```
# include <Wire.h>
# include <Adafruit_GFX.h>
# include <Adafruit_SSD1306.h>

# define SCREEN_WIDTH 128
# define SCREEN_HEIGHT 64
# define OLED_ADDR 0x3C

Adafruit_SSD1306 display(SCREEN_WIDTH ,
 SCREEN_HEIGHT , &Wire , -1);

#define LED1 25
# define LED2 27
# define LED3 33
# define BUZZER_PIN 26
# define BUTTON_PIN 14

bool ledState = false;
unsigned long pressStart = 0;
bool isPressed = false;
const unsigned long longPressTime = 1500;

void showOnOLED(const String &Line1,
 const String &Line2 = "") {
 display.ClearDisplay();

 display.setTextSize(1);
 display.setTextColor(SSD1306_WHITE);
```

```
display. setCursor (10, 0);
display. println ("Press- Type Detect");
display. drawline (0, 10, 127, 10, SSD1306_WRITE);

display. setCursor (10, 25);
display. println (line1);
if (line 2 != " ") {
    display. setCursor (10, 40);
    display. println (line2);
}
    display. display();
}

void setup() {
  Serial. begin (115200);

pinMode (LED1, OUTPUT);
pinMode (LED2, OUTPUT);
pinMode (LED3, OUTPUT);
pinMode (BUZZER_PIN, OUTPUT);
pinMode (BUTTON_PIN, INPUT_PULLUP);

Wire. begin (21, 22);
if (!display. begin (SSD1306_SWITCHAPVCC,
                     OLED_ADDR)) {
  Serial. println ("OLED not found!");
  while (true);
}

ShowOnOLED ("System Ready . . .", "Press Button")
 delay (1500);
```

```
  digitalWrite (LED1 , LOW);
  digitalWrite (LED2 ,LOW);
  digitalWrite (LED3 , LOW);
  digitalWrite (BUZZER_PIN , LOW);
}
void loop(){
int buttonState = digitalRead (BUTTON_PIN);

if ( buttonstate == LOW && !isPressed){
  isPressed = true;
  pressStart = millis();
}

if (buttonState == HIGH && isPressed){
 unsigned long pressDuration = millis() -
                    pressStart;
 isPressed = false;

if (pressDuration < longPressTime){

  LedState = ! ledState;
  digitalWrite(LED1, ledstate);
  digitalWrite(LED2, ledstate);
  digitalWrite (LED3, LedState);

 showOnOLED ("Short-Press Detected",
  LedState ? "LEDs: ON" : "LEDs : OFF");
 Serial.println (ledstate ? "LEDs ON" :
                  "LEDs OFF");
} else {
  showOnOLED ("Long-Press Detected", "Playing
        Buzzer...");
  tone (BUZZER_PIN, 1000, 500);
```

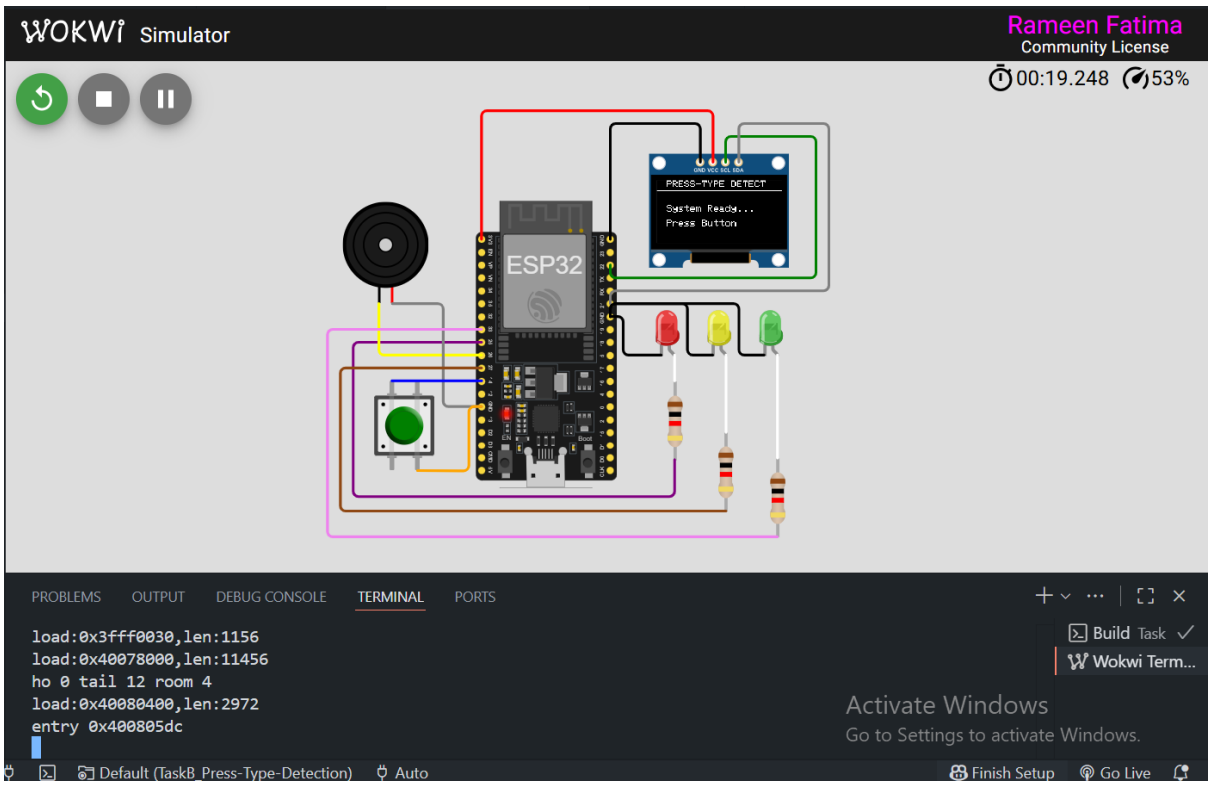
```
    Serial.println("Buzzer Tone Played");
  }
  delay(300);
}
}
```

Hand sketch diagram:

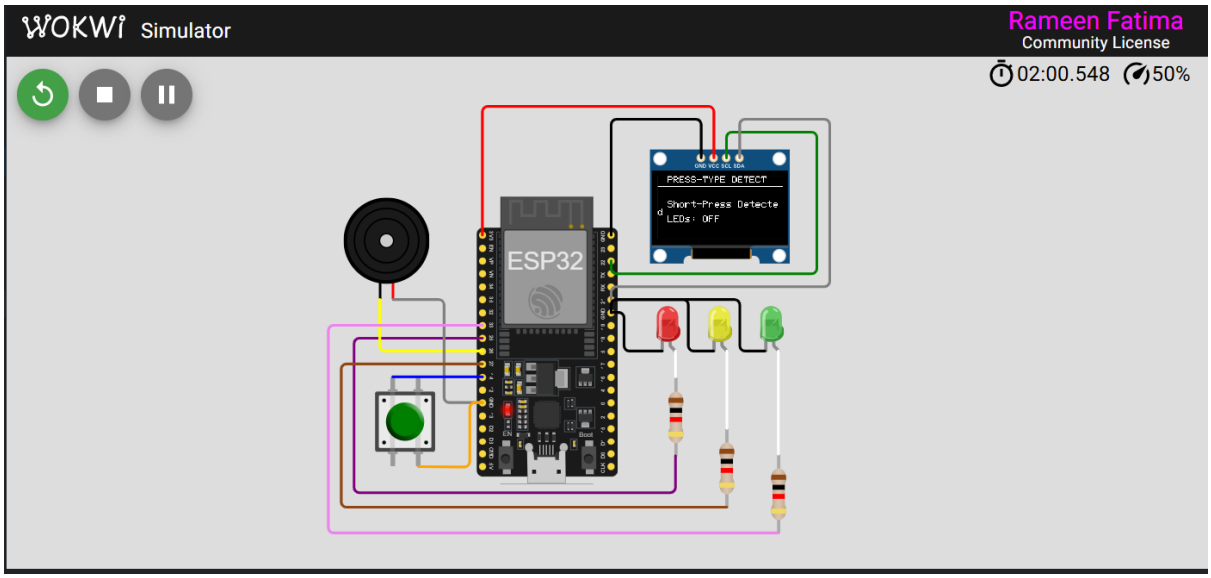


Wokwi circuit diagram:

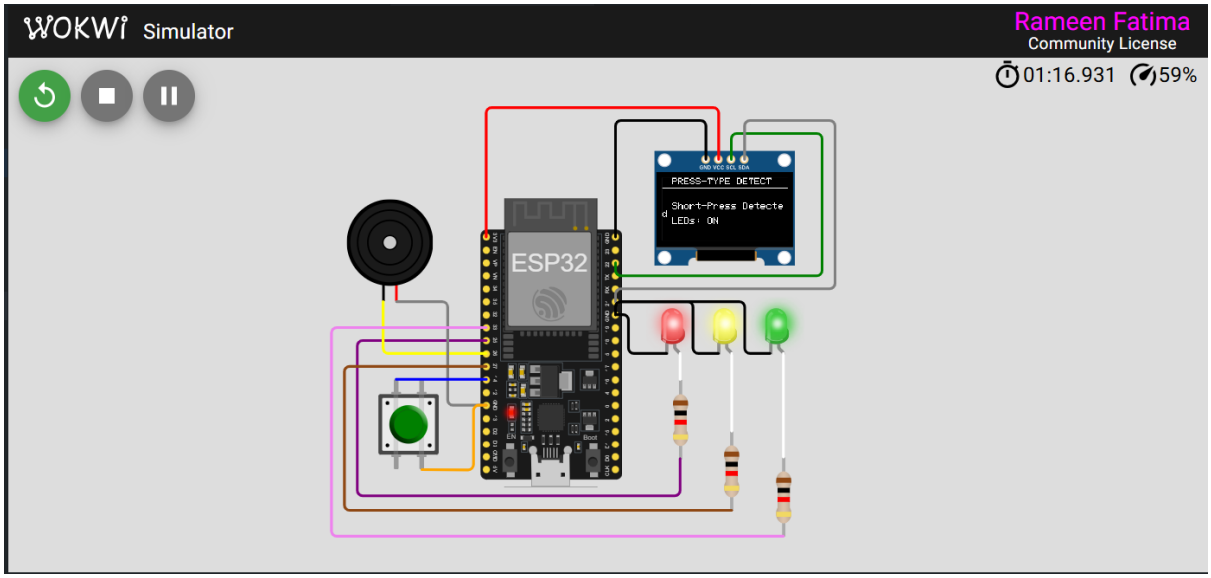
Initial state:



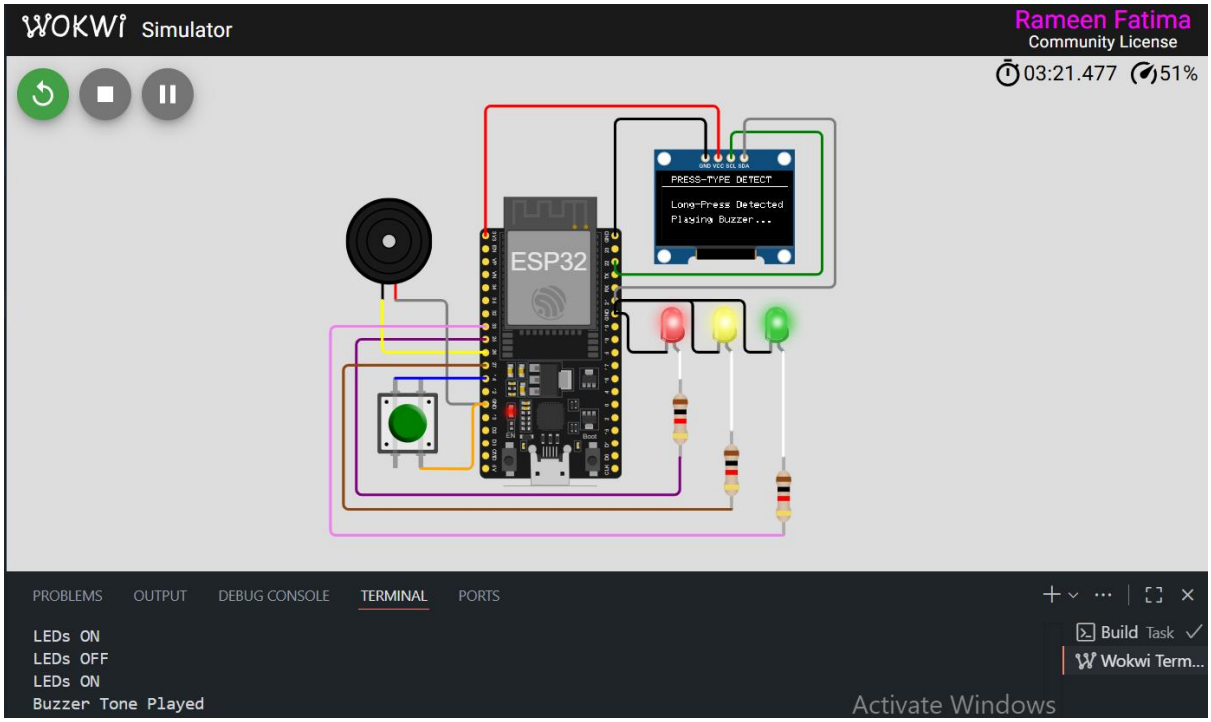
Short press LED OFF:



Short press LED ON:



Long press Buzzer played:



Wokwi Project Link:

https://wokwi.com/projects/445452596000373761