National Textile University, Faisalabad

Department of Computer Science

Name:	**Samar Khalid**
Class:	**BSCS-5B**
Registration No:	**23-NTU-CS-1091**
Assignment:	**1 (Task-b)**
Course Name:	**Embedded IoT systems**
Submitted To:	**Sir Nasir Mahmood**
Submission Date:	23-10-2025

Assignment 1

Task B

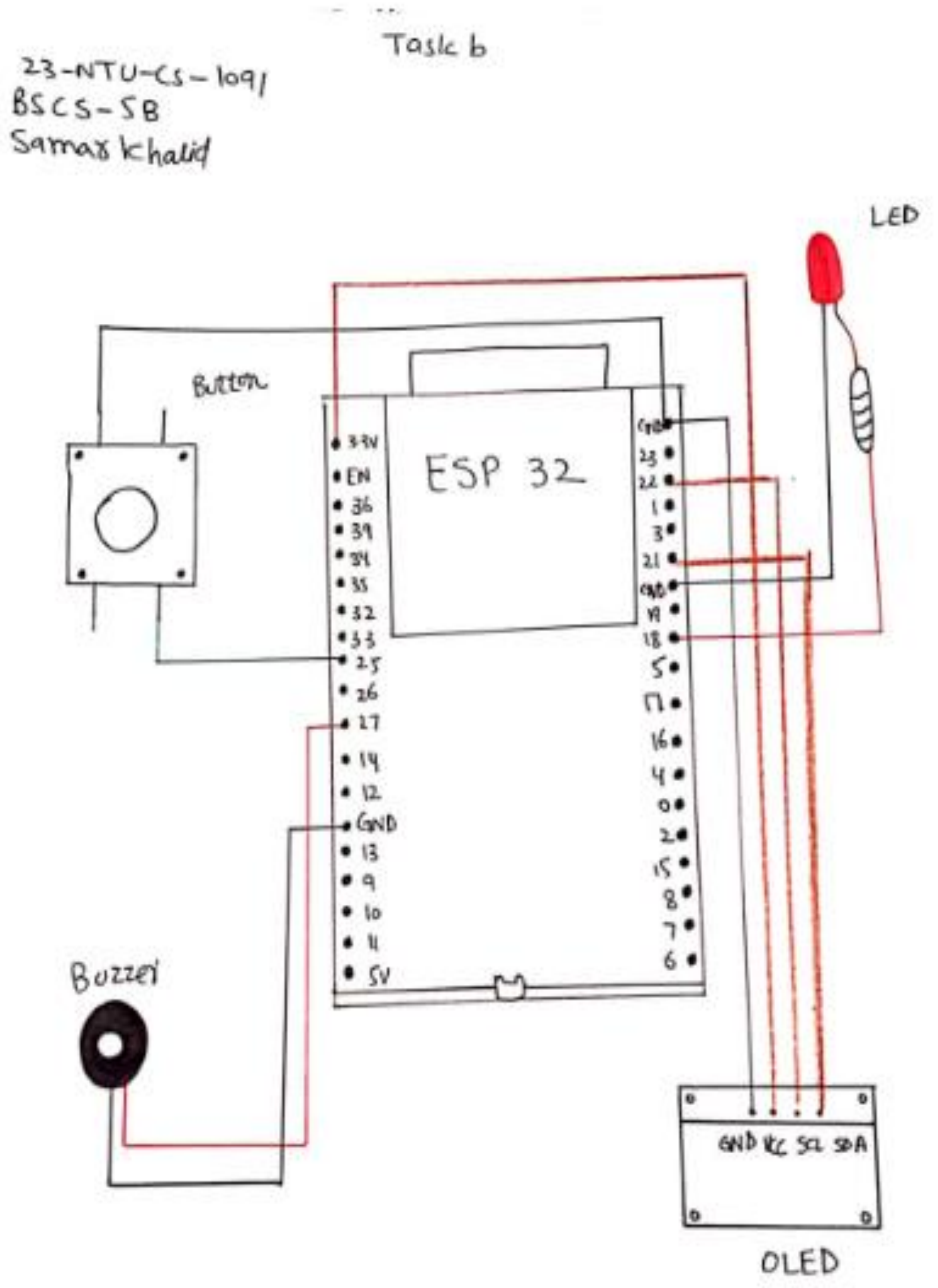
Task Explanation:

This code detects short and long button presses using an ESP32 equipped with a button, LED, buzzer, and OLED display. The LED turns ON or OFF and the screen displays "Short Press" when you press the button rapidly. The buzzer sounds and the message "Long Press" appears on the screen if you hold the button for longer than 1.5 seconds. Every action receives feedback from the OLED, and the buzzer or LED physically displays the outcome. When everything is functioning, it first sets up all the pins and displays "System Ready." In summary, the length of time you press the button effects how the program responds.

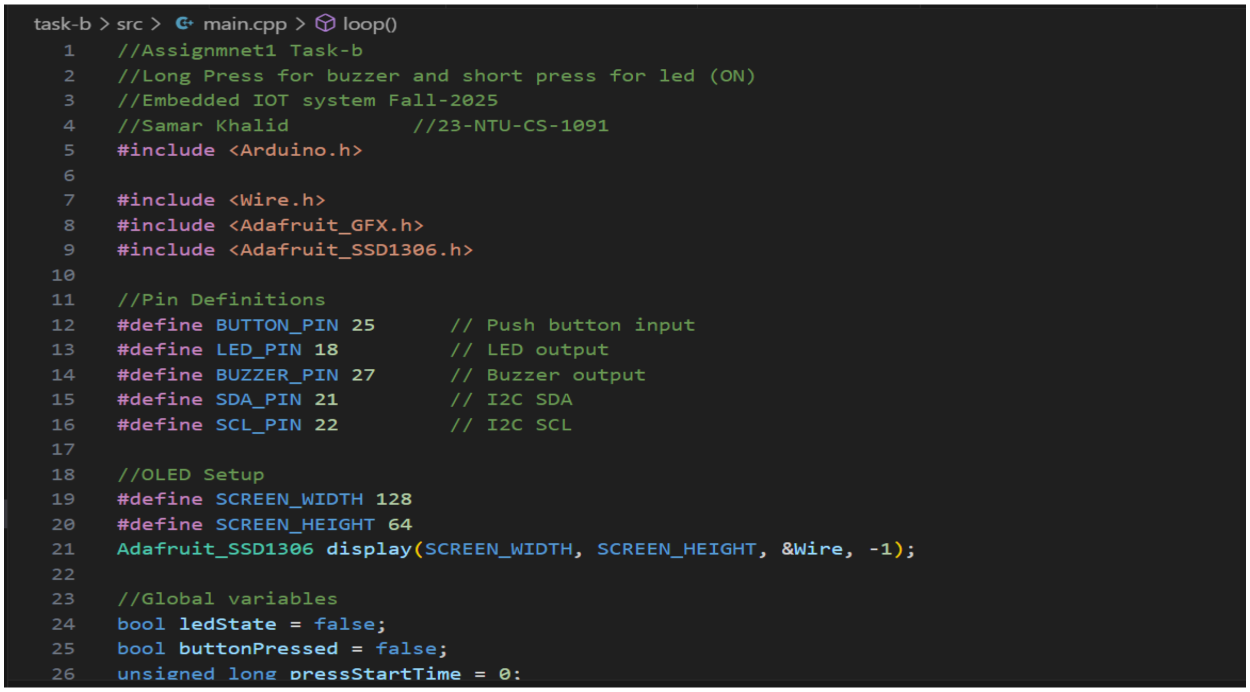
Pin Diagram:

Device Names	Device Pins	Esp-32 Pins
OLED	Vcc3	3.3 V
OLED	GND	GND
OLED	SCL	GPIO22
OLED	SDA	GPIO21
LED	Cathode (short leg)	GND
LED	Anode (Long leg)	GPIO18 (through resistor)
Buzzer	Cathode (Black)	GND
Buzzer	Anode (Red)	GPIO27
Button	Leg one	GND
Button	Other leg	GPIO25

Circuit Diagram:

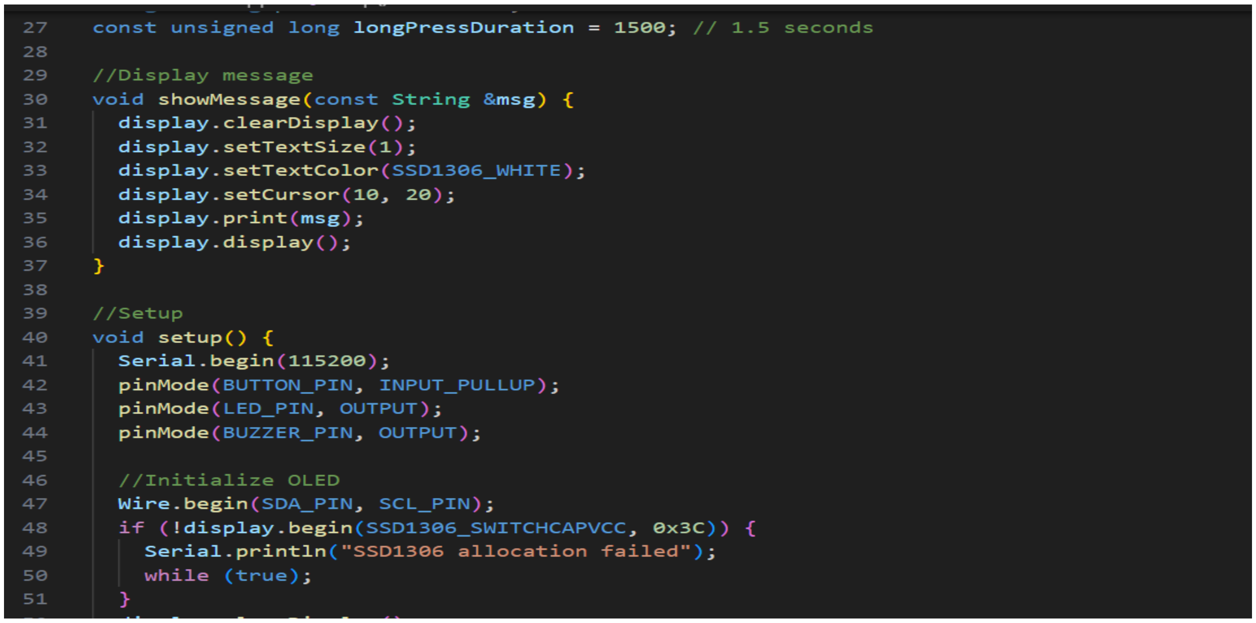


Code screenshot:

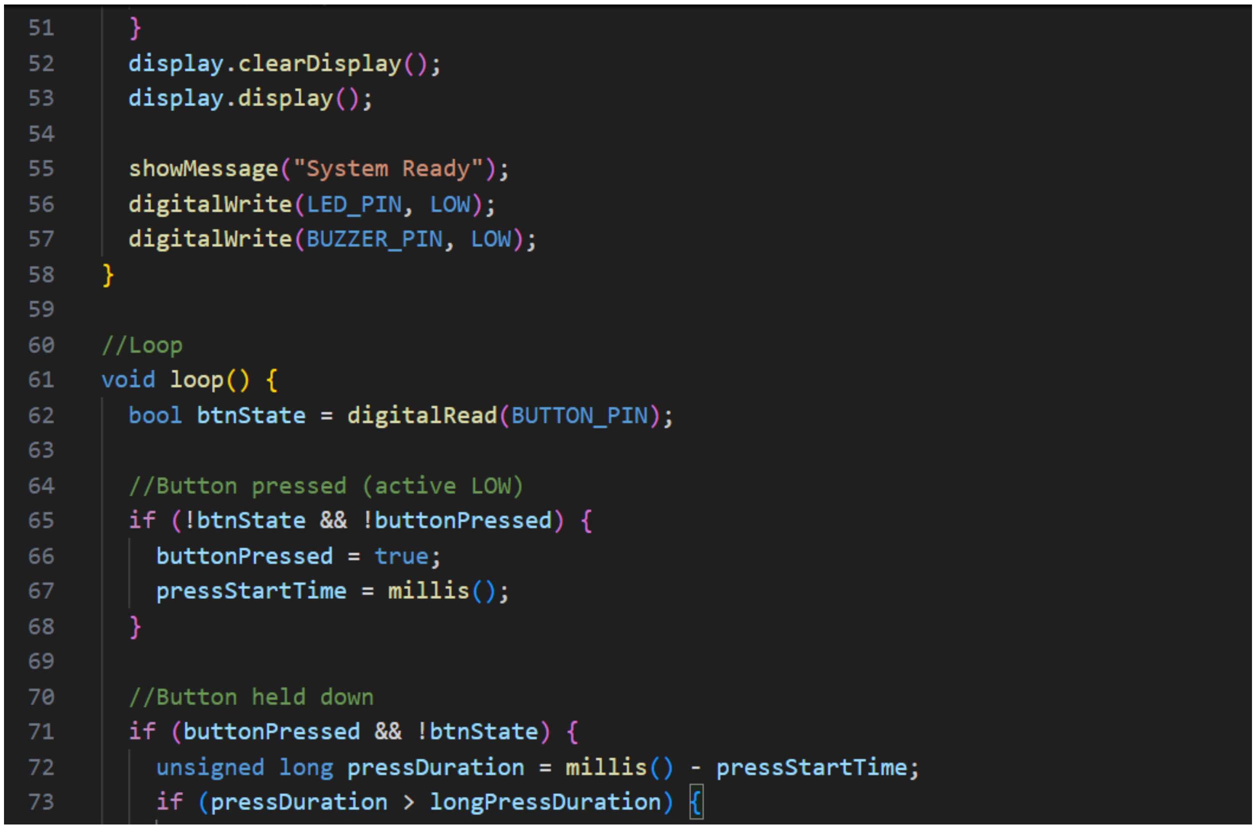


task-b > src > main.cpp > loop()

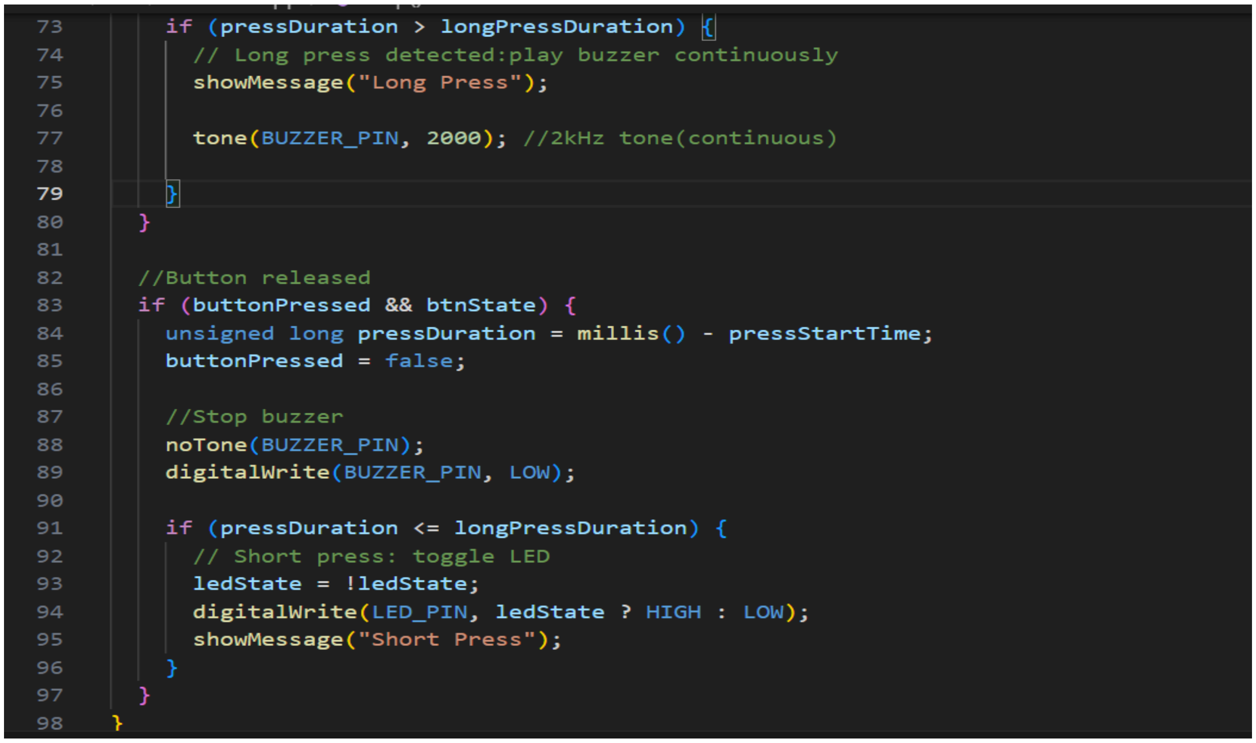
```cpp
//Assignmnet1 Task-b
//Long Press for buzzer and short press for led (ON)
//Embedded IOT system Fall-2025
//Samar Khalid          //23-NTU-CS-1091
#include <Arduino.h>

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

//Pin Definitions
#define BUTTON_PIN 25      // Push button input
#define LED_PIN 18         // LED output
#define BUZZER_PIN 27      // Buzzer output
#define SDA_PIN 21         // I2C SDA
#define SCL_PIN 22         // I2C SCL

//OLED Setup
#define SCREEN_WIDTH 128
#define SCREEN_HEIGHT 64
Adafruit_SSD1306 display(SCREEN_WIDTH, SCREEN_HEIGHT, &Wire, -1);

//Global variables
bool ledState = false;
bool buttonPressed = false;
unsigned long pressStartTime = 0;
```



```cpp
const unsigned long longPressDuration = 1500; // 1.5 seconds

//Display message
void showMessage(const String &msg) {
  display.clearDisplay();
  display.setTextSize(1);
  display.setTextColor(SSD1306_WHITE);
  display.setCursor(10, 20);
  display.print(msg);
  display.display();
}

//Setup
void setup() {
  Serial.begin(115200);
  pinMode(BUTTON_PIN, INPUT_PULLUP);
  pinMode(LED_PIN, OUTPUT);
  pinMode(BUZZER_PIN, OUTPUT);

  //Initialize OLED
  Wire.begin(SDA_PIN, SCL_PIN);
  if (!display.begin(SSD1306_SWITCHCAPVCC, 0x3C)) {
    Serial.println("SSD1306 allocation failed");
    while (true);
  }
```

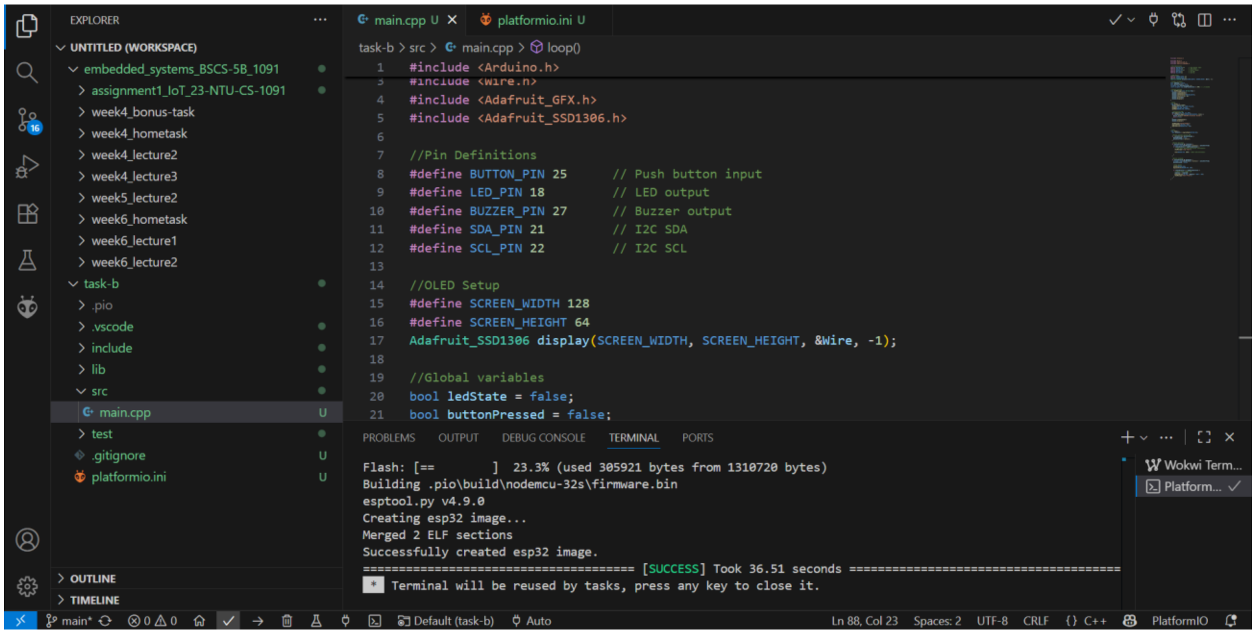


```
  }
  display.clearDisplay();
  display.display();

  showMessage("System Ready");
  digitalWrite(LED_PIN, LOW);
  digitalWrite(BUZZER_PIN, LOW);
}

//Loop
void loop() {
  bool btnState = digitalRead(BUTTON_PIN);

  //Button pressed (active LOW)
  if (!btnState && !buttonPressed) {
    buttonPressed = true;
    pressStartTime = millis();
  }

  //Button held down
  if (buttonPressed && !btnState) {
    unsigned long pressDuration = millis() - pressStartTime;
    if (pressDuration > longPressDuration) {
```



```
    if (pressDuration > longPressDuration) {
      // Long press detected:play buzzer continuously
      showMessage("Long Press");

      tone(BUZZER_PIN, 2000); //2kHz tone(continuous)

    }
  }

  //Button released
  if (buttonPressed && btnState) {
    unsigned long pressDuration = millis() - pressStartTime;
    buttonPressed = false;

    //Stop buzzer
    noTone(BUZZER_PIN);
    digitalWrite(BUZZER_PIN, LOW);

    if (pressDuration <= longPressDuration) {
      // Short press: toggle LED
      ledState = !ledState;
      digitalWrite(LED_PIN, ledState ? HIGH : LOW);
      showMessage("Short Press");
    }
  }
}
```

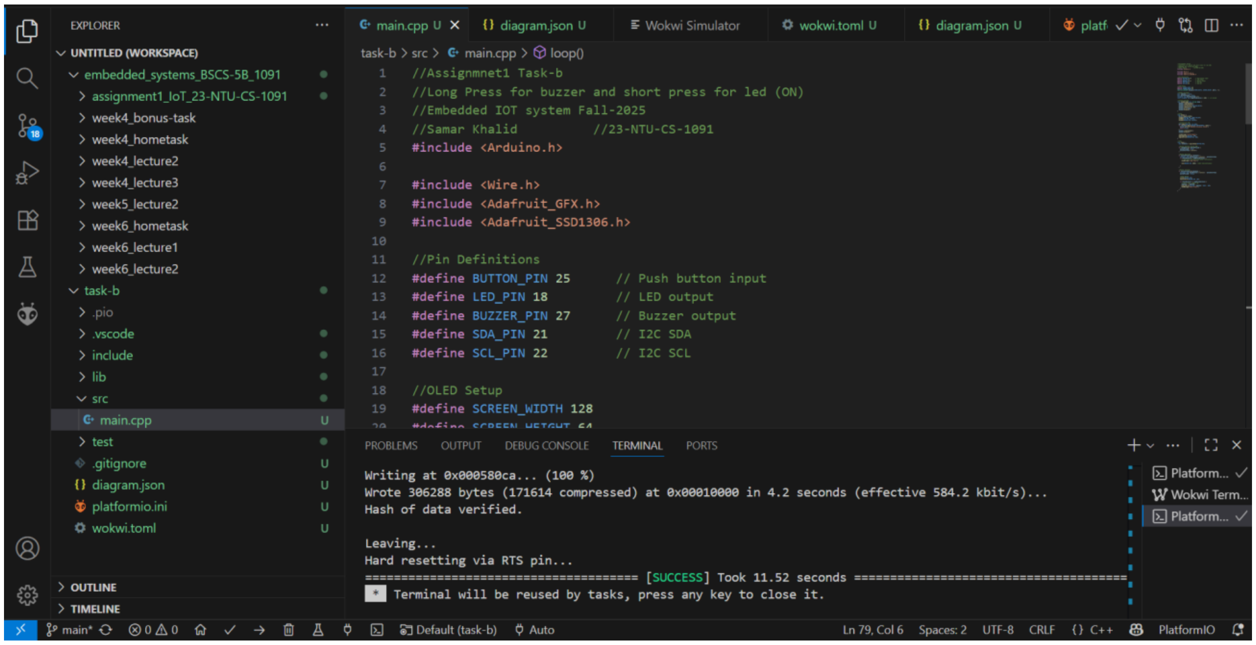
VS code Build success:



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

//Pin Definitions
#define BUTTON_PIN 25      // Push button input
#define LED_PIN 18         // LED output
#define BUZZER_PIN 27      // Buzzer output
#define SDA_PIN 21         // I2C SDA
#define SCL_PIN 22         // I2C SCL

//OLED Setup
#define SCREEN_WIDTH 128
#define SCREEN_HEIGHT 64
Adafruit_SSD1306 display(SCREEN_WIDTH, SCREEN_HEIGHT, &Wire, -1);

//Global variables
bool ledState = false;
bool buttonPressed = false;
```

```
Flash: [==        ]  23.3% (used 305921 bytes from 1310720 bytes)
Building .pio\build\nodemcu-32s\firmware.bin
esptool.py v4.9.0
Creating esp32 image...
Merged 2 ELF sections
Successfully created esp32 image.
==================================== [SUCCESS] Took 36.51 seconds ====================================
 *  Terminal will be reused by tasks, press any key to close it.
```

Upload on ESP-32 success:

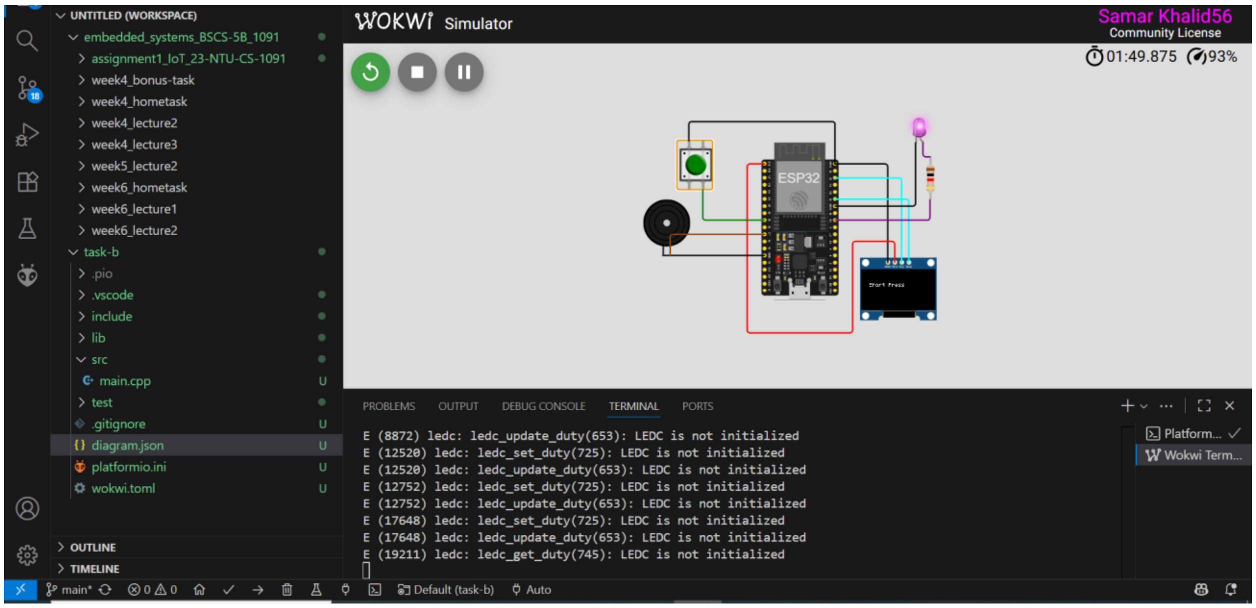


```
//Assignmnet1 Task-b
//Long Press for buzzer and short press for led (ON)
//Embedded IOT system Fall-2025
//Samar Khalid          //23-NTU-CS-1091
#include <Arduino.h>

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

//Pin Definitions
#define BUTTON_PIN 25      // Push button input
#define LED_PIN 18         // LED output
#define BUZZER_PIN 27      // Buzzer output
#define SDA_PIN 21         // I2C SDA
#define SCL_PIN 22         // I2C SCL

//OLED Setup
#define SCREEN_WIDTH 128
```

```
Writing at 0x000580ca... (100 %)
Wrote 306288 bytes (171614 compressed) at 0x00010000 in 4.2 seconds (effective 584.2 kbit/s)...
Hash of data verified.

Leaving...
Hard resetting via RTS pin...
====================================== [SUCCESS] Took 11.52 seconds ======================================
 *  Terminal will be reused by tasks, press any key to close it.
```

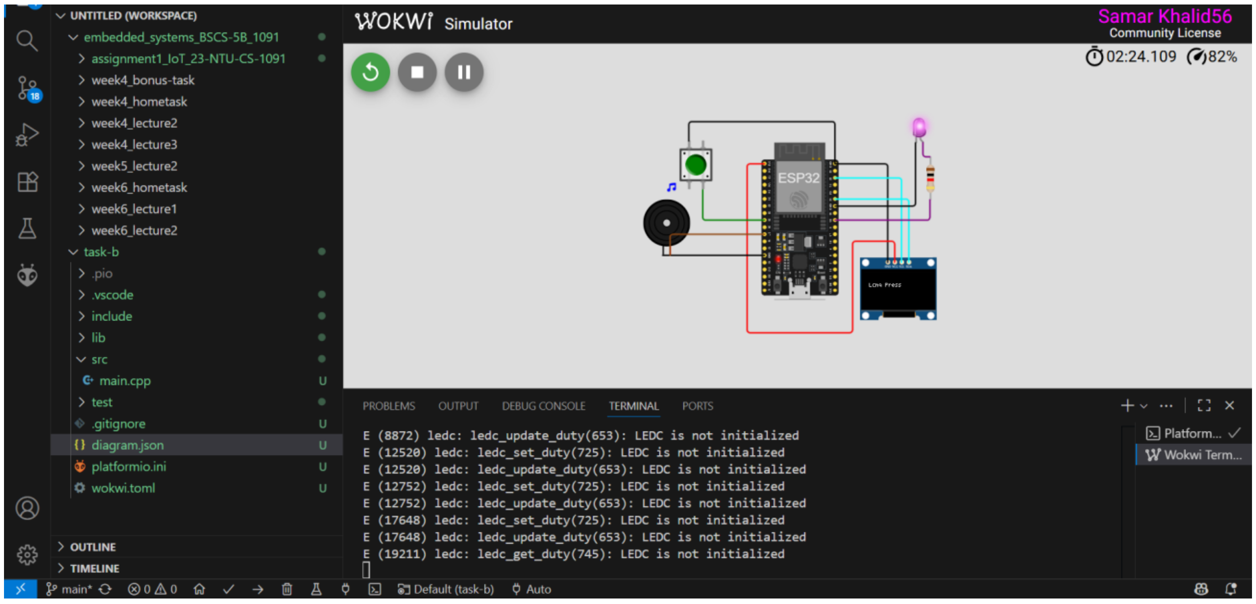
Output on wokwi:

Short Press:



```
E (8872) ledc: ledc_update_duty(653): LEDC is not initialized
E (12520) ledc: ledc_set_duty(725): LEDC is not initialized
E (12520) ledc: ledc_update_duty(653): LEDC is not initialized
E (12752) ledc: ledc_set_duty(725): LEDC is not initialized
E (12752) ledc: ledc_update_duty(653): LEDC is not initialized
E (17648) ledc: ledc_set_duty(725): LEDC is not initialized
E (17648) ledc: ledc_update_duty(653): LEDC is not initialized
E (19211) ledc: ledc_get_duty(745): LEDC is not initialized
```

Long Press:

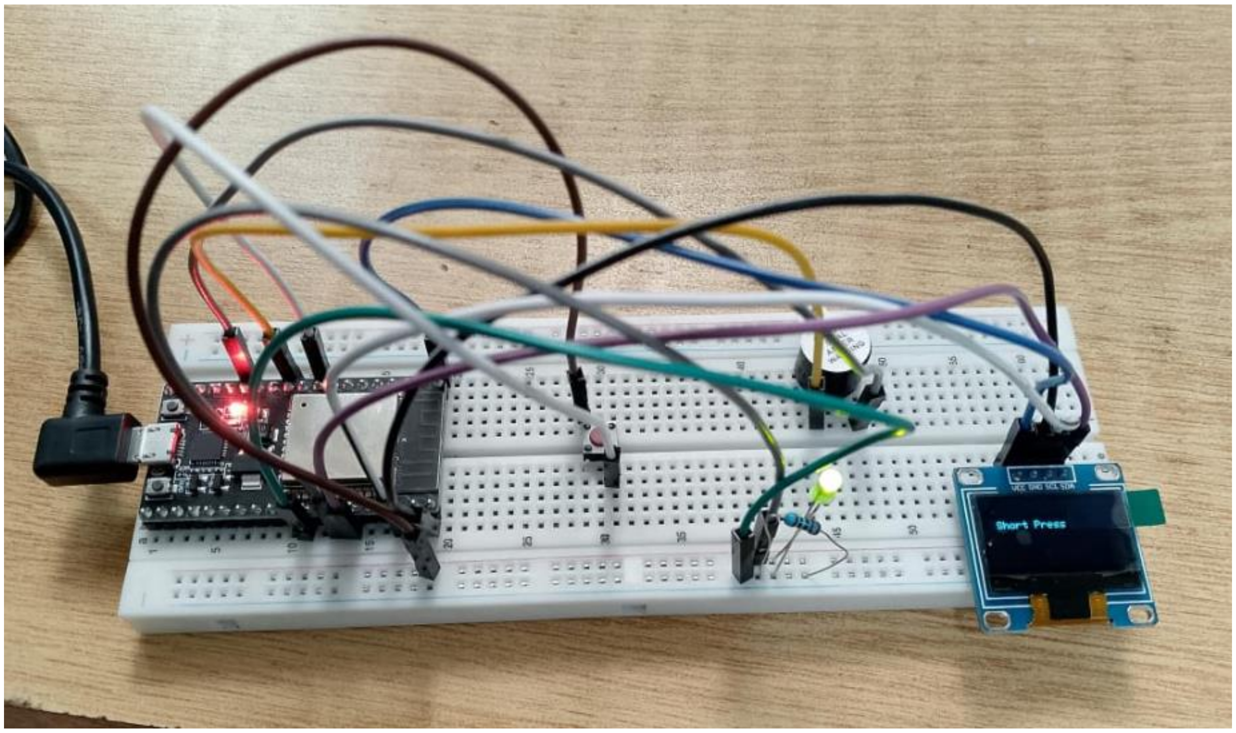


```
E (8872) ledc: ledc_update_duty(653): LEDC is not initialized
E (12520) ledc: ledc_set_duty(725): LEDC is not initialized
E (12520) ledc: ledc_update_duty(653): LEDC is not initialized
E (12752) ledc: ledc_set_duty(725): LEDC is not initialized
E (12752) ledc: ledc_update_duty(653): LEDC is not initialized
E (17648) ledc: ledc_set_duty(725): LEDC is not initialized
E (17648) ledc: ledc_update_duty(653): LEDC is not initialized
E (19211) ledc: ledc_get_duty(745): LEDC is not initialized
```

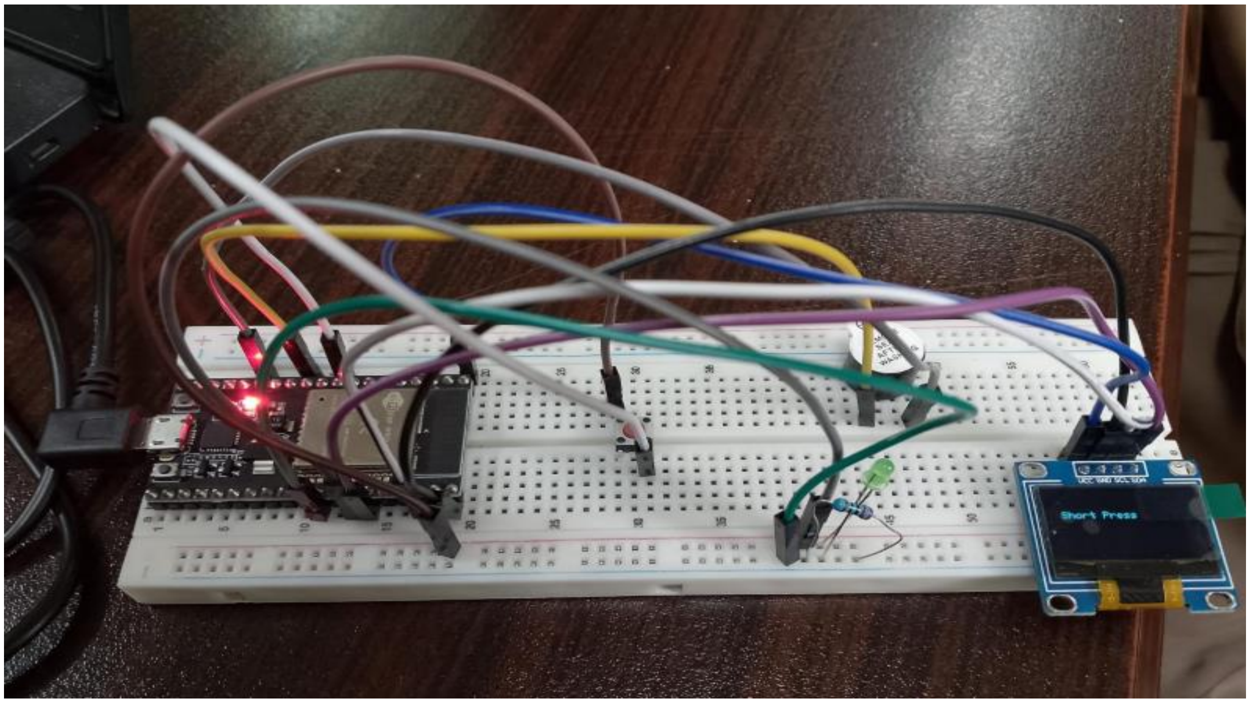
Output on Kit:

Short press:

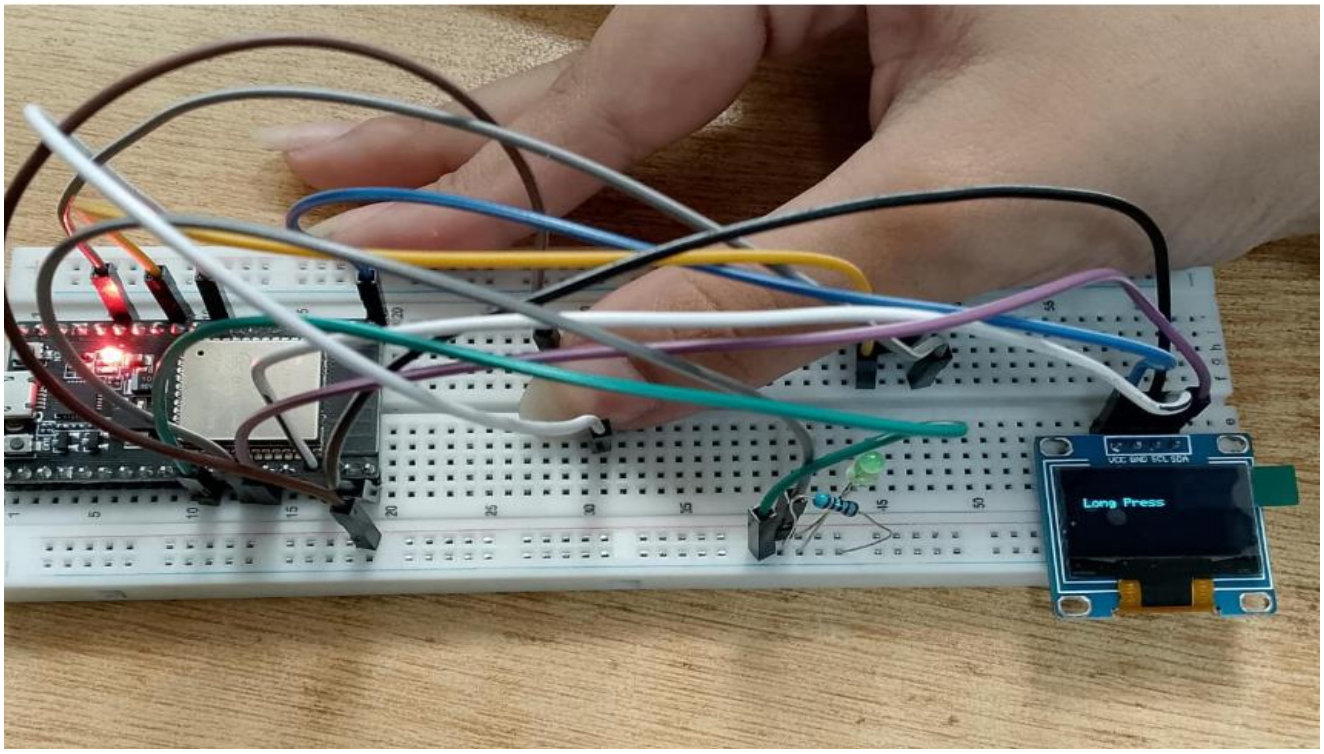
LED ON:



LED OFF:



Long Press:



Wokwi link:

https://wokwi.com/projects/445578718745765889

Handwritten Code:

Assignment 1

(Task b)

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

// Pin Definitions
#define BUTTON_PIN 25
# define LED_PIN 18
# define BUZZER_PIN 27
# define SDA_PIN 21
# define SCL_PIN 22

// OLED Setup
# define SCREEN_WIDTH 128
# define SCREEN_HEIGHT 64
Adafruit_SSD1306 display (SCREEN_WIDTH, SCREEN_
HEIGHT, &Wire, -1);

// Global variables
bool ledstate = false;
bool buttonPressed = false;
Unsigned long pressStartTime = 0;
const unsigned long longPressDuration = 1500; //1.5
                                               seconds
```

```
// Display message
void showMessage (const String &msg) {

    display.clearDisplay ();
    display.setTextSize( 1);
    display.setTextColor (SSD1306_WHITE);
    display.setCursor (10,20);
    display.print (msg);
    display.display ();
}

// setup
void setup() {

   Serial.begin (115200);
   pinMode ( BUTTON_PIN, INPUT_PULLUP);
   pinMode (LED_PIN, OUTPUT);
   pinMode (BUZZER_PIN, OUTPUT);

   // Initialize OLED
   Wire.begin (SDA_PIN, SCL_PIN);
   if (!display.begin (SSD1306_SWITCHCAPVCC,
                                          0X32)){
       Serial.println("SSD1306 allocation failed");
       While(true);
   }
    display.clearDisplay ();
    display.display ();
    ShowMessage ("System Ready");
    digital Write( LED_PIN , LOW);
    digitalWrite ( BUZZER_PIN, LOW);
}

//Loop
void loop() {
  bool btnState = digitalRead (BUTTON_PIN);

   // Button pressed ( active low)
   if ( !btnState && !button Pressed) {
       button Pressed = true;
       press startTime = millis();
   }

   // Button held down

   if ( buttonPressed && btnState) {
     unsigned long pressDuration = millis()-pressStartTim.
     button Pressed = false ;

     // stop buzzer
     noTone ( BUZZER_PIN);
     digital Write ( BUZZER PIN, LOW);

     if ( pressDuration <= longpress Duration) {
        // short Press → toggle LED
        ledstate =! ledstate;
        digitalWrite (LED-PIN, ledState ? HIGH : LOW
        showMessage ("short Press");
        }
    }
}
```