

# National Textile University, Faisalabad



## Department of Computer Science

<b>Name:</b>	SAJAL NAEEM
<b>Class:</b>	BSCS 5 <sup>TH</sup> B
<b>Registration No:</b>	23-NTU-CS-1089
<b>Assignment:</b>	1st
<b>Course Name:</b>	IOT AND EMBEDDED SYSTEMS
<b>Submitted To:</b>	SIR NASIR MEHMOOD
<b>Submission Date:</b>	26 <sup>TH</sup> OCTOBER 2025

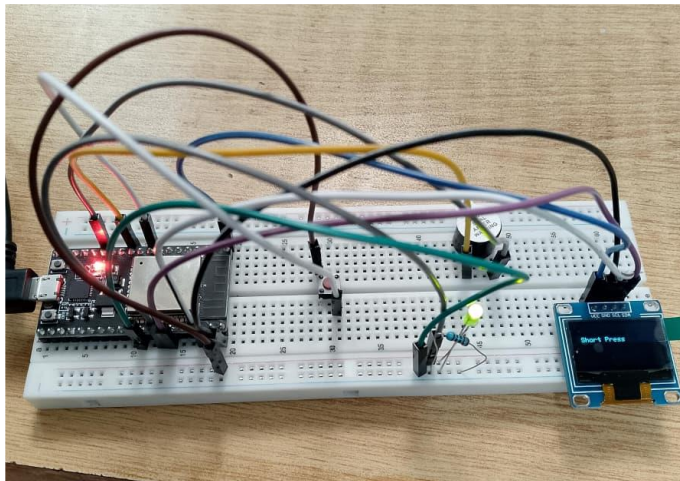
## Code Screenshot:

```
1 #include <Arduino.h>
2 #include <Wire.h>
3 #include <Adafruit_GFX.h>
4 #include <Adafruit_SSD1306.h>
5
6 // Pin Definitions
7 #define BUTTON_PIN 25
8 #define LED_PIN 18
9 #define BUZZER_PIN 27
10 #define SDA_PIN 21
11 #define SCL_PIN 22
12
13 // OLED Setup
14 #define SCREEN_WIDTH 128
15 #define SCREEN_HEIGHT 64
16 Adafruit_SSD1306 display(SCREEN_WIDTH, SCREEN_HEIGHT, &Wire, -1);
17
18 // Globals
19 bool ledState = false;
20 bool buttonPressed = false;
21 unsigned long pressStartTime = 0;
22 const unsigned long longPressDuration = 1500;
23
24 // Display message
25 void showMessage(const String &msg) {
26   display.clearDisplay();
27   display.setTextSize(1);
28   display.setTextColor(SSD1306_WHITE);
29   display.setCursor(10, 20);
30   display.print(msg);
31   display.display();
32 }
33
34 // Setup
35 void setup() {
36   Serial.begin(115200);
37   pinMode(BUTTON_PIN, INPUT_PULLUP);
38   pinMode(LED_PIN, OUTPUT);
39   pinMode(BUZZER_PIN, OUTPUT);
40
41   // Initialize OLED
42   Wire.begin(SDA_PIN, SCL_PIN);
43   if (!display.begin(SSD1306_SWITCHCAPVCC, 0x3C)) {
44     Serial.println("SSD1306 allocation failed");
45     while (true);
46   }
47   display.clearDisplay();
48   display.display();
49
50   showMessage("System Ready");
51   digitalWrite(LED_PIN, LOW);
52   digitalWrite(BUZZER_PIN, LOW);
53 }
54
55 // Loop
56 void loop() {
57   bool btnState = digitalRead(BUTTON_PIN);
58
59   if (!btnState && !buttonPressed) {
60     buttonPressed = true;
61     pressStartTime = millis();
62   }
63
64   // Button held down
65   if (buttonPressed && !btnState) {
66     unsigned long pressDuration = millis() - pressStartTime;
67     if (pressDuration > longPressDuration) {
68
69       showMessage("Long Press");
70
71       tone(BUZZER_PIN, 2000);
72
73     }
74   }
75
76   // Button released
77   if (buttonPressed && btnState) {
78     unsigned long pressDuration = millis() - pressStartTime;
79     buttonPressed = false;
80
81     // Stop buzzer
82     noTone(BUZZER_PIN);
83     digitalWrite(BUZZER_PIN, LOW);
84
85     if (pressDuration <= longPressDuration) {
86       ledState = !ledState;
87       digitalWrite(LED_PIN, ledState ? HIGH : LOW);
88       showMessage("Short Press");
89     }
90   }
91 }
92
93 }
```

## Code Explanation

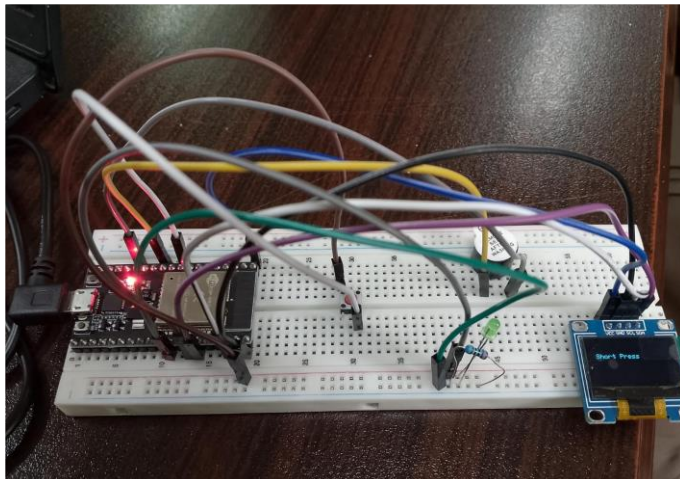
This Arduino code controls an LED, a buzzer, and an OLED display using a single button. When the system starts, it shows “System Ready” on the screen. If you press and quickly release the button, it’s detected as a short press — the LED toggles ON or OFF, and “Short Press” appears on the OLED. If you hold the button for more than 1.5 seconds, it’s a long press — the display shows “Long Press” and the buzzer sounds while the button is held.

**Output Screenshot:**



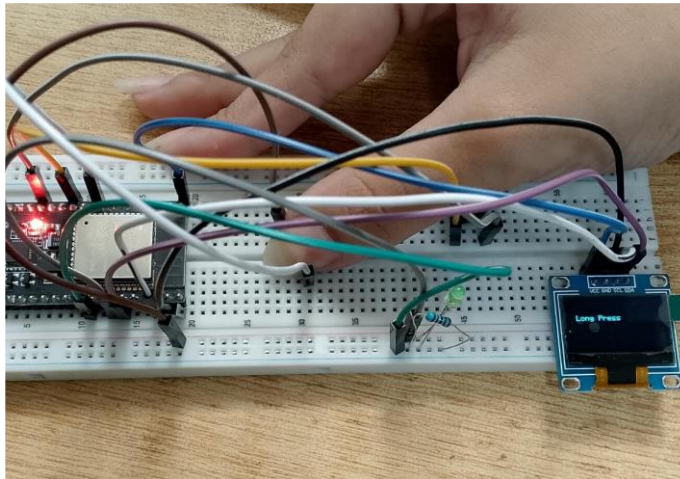
Short press

led is on



short press

led is off



long press

buzzer is played

### Code Build Screenshot:

```
src> C: main.cpp > ...
42 void setup() {
43   digitalWrite(LED_PIN, LOW);
44   display.display();
45 }
46
47 void loop() {
48   bool btnState = digitalRead(BUTTON_PIN);
49   if (!btnState && !buttonPressed) {
50     digitalWrite(LED_PIN, HIGH);
51     digitalWrite(BUZZER_PIN, LOW);
52   }
53   buttonPressed = true;
54 }
55
56 // Loop
57 void loop() {
58   bool btnState = digitalRead(BUTTON_PIN);
59   if (!btnState && !buttonPressed) {
60     digitalWrite(LED_PIN, HIGH);
61     digitalWrite(BUZZER_PIN, LOW);
62   }
63   buttonPressed = true;
64 }
```

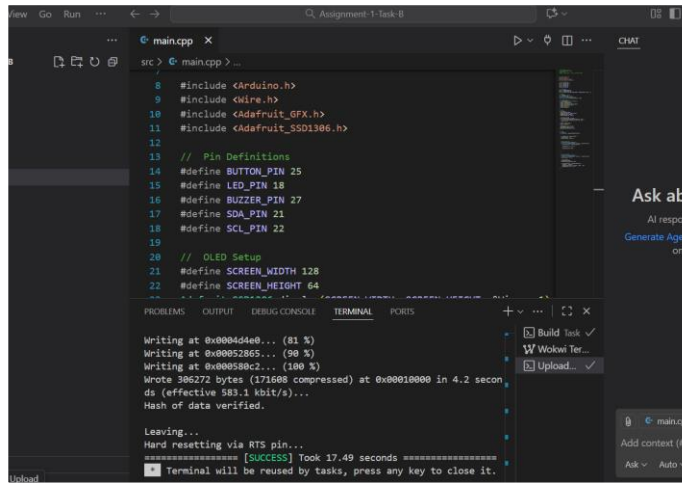
PROBLEMS OUTPUT TERMINAL

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 22888 bytes from 327680 bytes)  
Flash: [==] 23.3% (used 385985 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 58.10 seconds  
Terminal will be reused by tasks, press any key to close it.

code is build  
successfully

code build

### Code Upload Success:



The screenshot shows the Wokwi IDE interface. The main editor displays a C++ file named `main.cpp` with the following content:

```
1 #include <Arduino.h>
2 #include <Wire.h>
3 #include <Adafruit_GFX.h>
4 #include <Adafruit_SSD1306.h>
5
6 // Pin Definitions
7 #define BUTTON_PIN 25
8 #define LED_PIN 18
9 #define BUZZER_PIN 27
10 #define SDA_PIN 21
11 #define SCL_PIN 22
12
13 // OLED Setup
14 #define SCREEN_WIDTH 128
15 #define SCREEN_HEIGHT 64
```

The bottom panel shows the `TERMINAL` tab with the following output:

```
Writing at 0x0004d4e0... (81 %)
Writing at 0x00052865... (90 %)
Writing at 0x00058bc2... (100 %)
Wrote 386272 bytes (171608 compressed) at 0x00018000 in 4.2 seconds (effective 583.1 kbit/s)...
Hash of data verified.
Leaving...
Hard resetting via RTS pin...
===== [SUCCESS] Took 17.49 seconds =====
Terminal will be reused by tasks, press any key to close it.
```

On the right side, there is a `CHAT` panel with a search bar and a list of messages. Below the chat panel, there are buttons for `Build task`, `Wokwi Ter...`, and `Upload...`.

Code is uploaded successfully

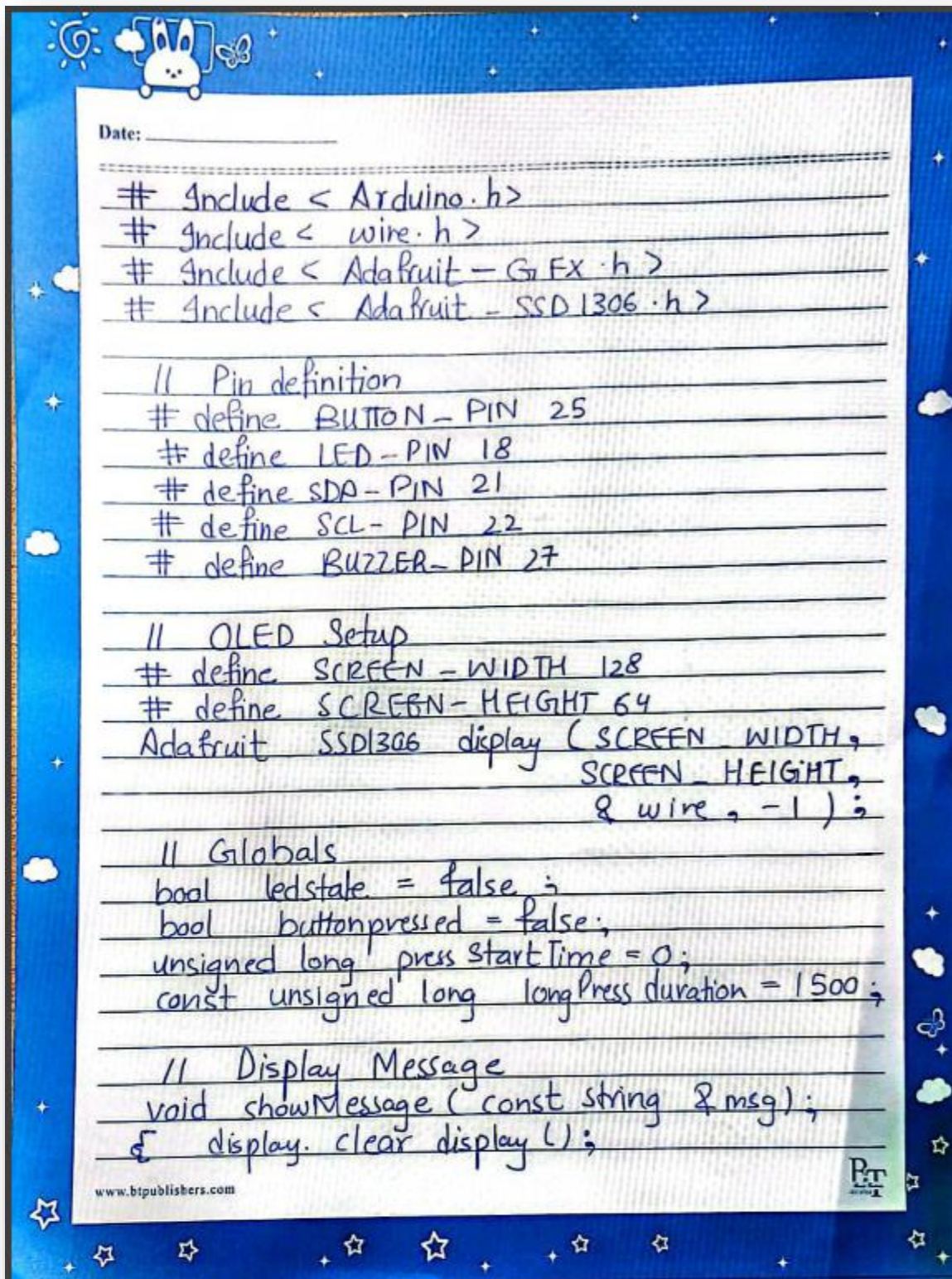
Code Upload

**WOKWI project link:**

<https://wokwi.com/projects/445578682914863105>



## Handwritten Code:





Date: \_\_\_\_\_

```
display.set Text size ( 1 ) ;  
display.set Text color ( SSD-1306 - white ) ;  
display.set cursor ( 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 - SWITCHCAP VCC ) )  
{ Serial.println ( "SSD allocation failed" ) ;  
  while ( true ) ;  
}
```

```
Show Message ( " System Ready " ) ;  
digitalWrite ( LED-PIN , LOW ) ;  
digitalWrite ( BUZZER-PIN , LOW ) ;
```

// Loop

void loop ( )

```
{ bool btnstate = digitalRead ( BUTTON-PIN ) ;}
```





Date: \_\_\_\_\_

// Button held down

```
if (button Pressed && ! btnstate)
{ unsigned long pressduration = millis();
  if (pressduration > long Press duration)
  { Show message ("long Press");
    tone (BUZZER-PIN, 2000);
  }
}
```

// Button Released

```
if (button Pressed && btnstate)
{ unsigned long Press Duration = millis();
  button pressed = false;
}
```

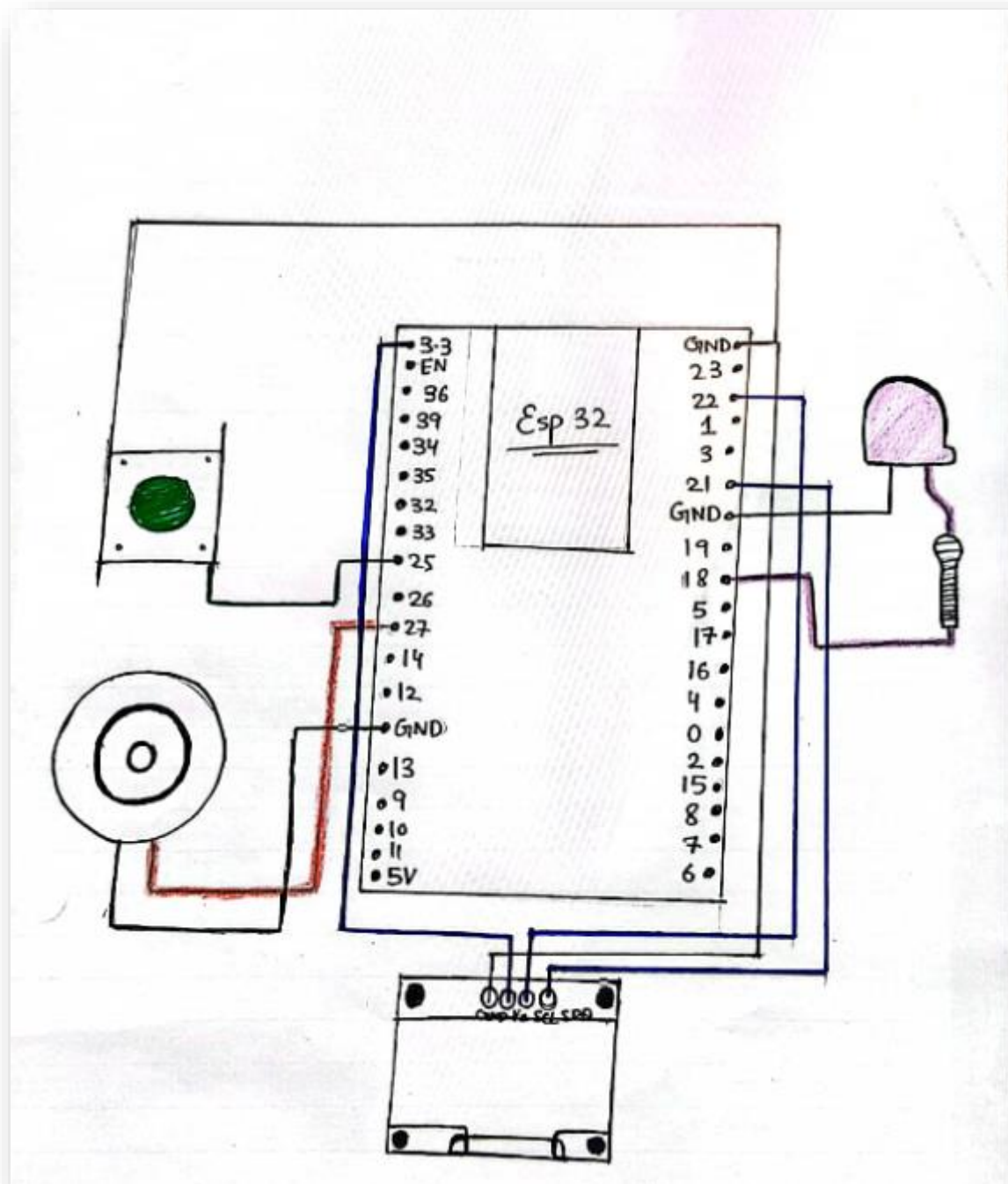
// Stop buzzer

```
noTone (BUZZER-PIN);
digitalWrite (BUZZER-PIN, LOW);

if (press Duration <= long Press Duration)
{ ledstate = ! ledstate;
  digitalWrite (LED-PIN, ledstate ? HIGH:
    LOW);
  show Message ("Short Press");
}
}
```



Hand Drawn Sketch:



**Pin map:**

Component	Pin Name	Pin number
LED	Resistor pin	GPIO 18
LED	Ground pin	GND 3
Button	Button pin	GPIO 25
Button	Ground pin	GND 2
OLED	Ground pin	GND 2
OLED	Voltage pin	GPIO 3.3
OLED	Serial clock pin	GPIO 22
OLED	Serial data pin	GPIO 21
Buzzer	Buzzer pin	GPIO 27
Buzzer	Ground Pin	GND 1