



ISEA Phase III Bootcamp on SDN, IoT and Network Slicing in 5G and Beyond

Topic : IoT enabled Text Display on OLED
Screen using Arduino Cloud IoT platform
or choose any IoT Platform

Submitted by:

— Group 3 —

Soumyasish Sarkar

Ranveer Bharti

Harsh Kumar

Souhardyo Sikder

Debasmita Sarma

Tezpur University

February 2025

Contents

1	Connection Diagram	2
2	Provided Hardware and Software Requirement	2
2.1	Hardware Requirements	2
2.2	Software Requirements	3
3	Step-wise Procedure	3
3.1	Step 1: Hardware Connections	3
3.2	Step 2: Software Setup	3
3.3	Step 3: Configure Arduino IoT Cloud	3
3.4	Step 4: Upload Code	4
4	Program Sequence and Code	4
4.1	Arduino Sketch	4
4.2	thingProperties.h	6
5	Conclusion	6

1 Connection Diagram

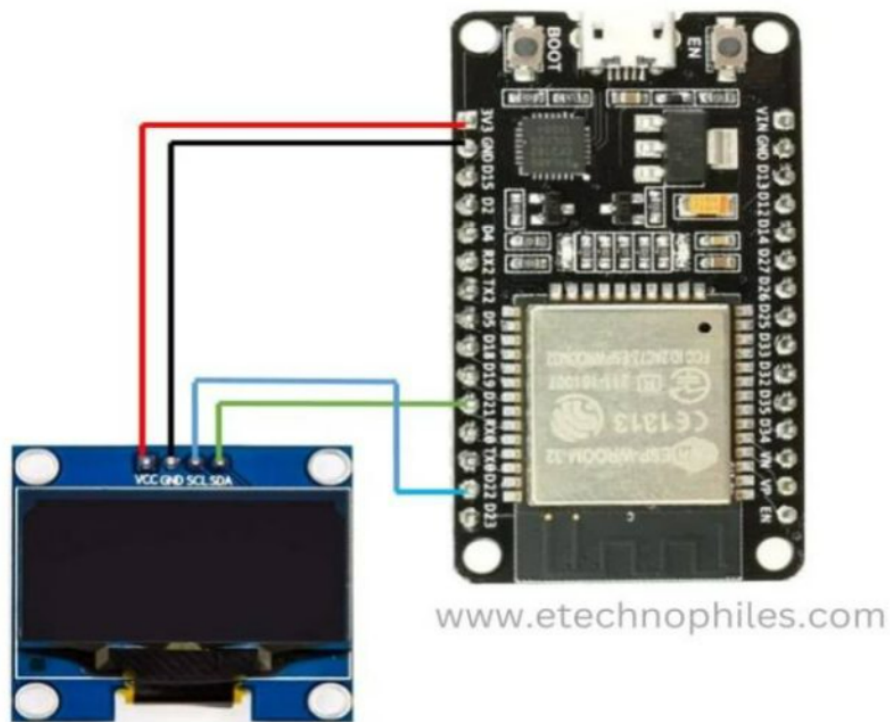


Figure 1: Connection Diagram for OLED Display with ESP32/NodeMCU

2 Provided Hardware and Software Requirement

2.1 Hardware Requirements

S. No	Component	Specification	Quantity
1	Microcontroller Board	NodeMCU / ESP32	1
2	OLED Display	128×64 px (SSD1306)	1
3	Breadboard	-	1
4	Jumper Wires	Male-Male, Female-Female	As required
5	USB Cable	Micro USB/Type-C (as per board)	1

2.2 Software Requirements

S. No	Software	Description	Availability
1	Arduino IDE	Programming the microcontroller	Download
2	Arduino IoT Cloud	IoT cloud platform for text input	Website
3	Libraries	Adafruit GFX, Adafruit SSD1306, Wire, IoT Cloud	Installed via Arduino IDE

3 Step-wise Procedure

3.1 Step 1: Hardware Connections

- Connect VCC, GND, SDA, and SCL of the OLED display to NodeMCU/ESP32.
- Use a breadboard and jumper wires for stable connections.
- Power the board using a USB cable.

3.2 Step 2: Software Setup

- Install Arduino IDE and required libraries:
 - Adafruit GFX
 - Adafruit SSD1306
 - Wire
 - ArduinoIoTCloud
- Set up an Arduino IoT Cloud account.

3.3 Step 3: Configure Arduino IoT Cloud

- Create a variable ‘displayText’ in Arduino IoT Cloud.
- Associate your ESP32/NodeMCU with the cloud.
- Enter Wi-Fi credentials in the setup.

3.4 Step 4: Upload Code

- Copy the program from Section 4.
- Paste into Arduino IDE.
- Select the correct Board & Port and upload.

4 Program Sequence and Code

4.1 Arduino Sketch

```
#include <Wire.h>
#include <Adafruit_GFX.h>
#include <Adafruit_SSD1306.h>
#include "thingProperties.h"
#include <ArduinoIoTCloud.h>
#include <Arduino_ConnectionHandler.h>

// OLED display dimensions
#define SCREEN_WIDTH 128
#define SCREEN_HEIGHT 64

// Define I2C pins for ESP32
#define OLED_SDA 21
#define OLED_SCL 22

// Create an SSD1306 display object connected to I2C
Adafruit_SSD1306 display(SCREEN_WIDTH, SCREEN_HEIGHT, &Wire, -1);

// Variable to track the last displayed text
String lastDisplayText = "";

void setup() {
  // Initialize Serial Monitor
  Serial.begin(115200);

  // Initialize I2C communication
  Wire.begin(OLED_SDA, OLED_SCL);

  // Initialize the OLED display
```

```

if (!display.begin(SSD1306_SWITCHCAPVCC, 0x3C)) {
    Serial.println(F("SSD1306 allocation failed"));
    for (;;)
}
display.clearDisplay();
display.setTextSize(2);
display.setTextColor(SSD1306_WHITE);

// Initialize Arduino IoT Cloud
initProperties();
ArduinoCloud.begin(ArduinoIoTPreferredConnection);

// Synchronize with Arduino Cloud
setDebugMessageLevel(2);
ArduinoCloud.printDebugInfo();
}

void loop() {
    ArduinoCloud.update(); // Maintain connection to Arduino Cloud

    // Update display only if the text has changed
    if (displayText != lastDisplayText) {
        display.clearDisplay();
        display.setCursor(0, 0);
        display.print(displayText);
        display.display();

        lastDisplayText = displayText; // Update the last displayed text
    }
}

// Callback function triggered when displayText changes
void onDisplayTextChange() {
    Serial.print("Received from Cloud: ");
    Serial.println(displayText);
}

```

4.2 thingProperties.h

```
// Code generated by Arduino IoT Cloud, DO NOT EDIT.
```

```
#include <ArduinoIoTCloud.h>
```

```
#include <Arduino_ConnectionHandler.h>
```

```
const char DEVICE_LOGIN_NAME[] = "6a5e6a0d-2eb7-4597-ba96-9d0c9eadc9fc";
```

```
const char SSID[] = "SSID";    // Network SSID (name)
```

```
const char PASS[] = "password"; // Network password
```

```
const char DEVICE_KEY[] = "DEVICE_KEY";    // Secret device password
```

```
void onDisplayTextChange();
```

```
String displayText;
```

```
void initProperties(){
```

```
    ArduinoCloud.setBoardId(DEVICE_LOGIN_NAME);
```

```
    ArduinoCloud.setSecretDeviceKey(DEVICE_KEY);
```

```
    ArduinoCloud.addProperty(displayText, READWRITE, ON_CHANGE, onDisplayTextChange);
```

```
}
```

```
WiFiConnectionHandler ArduinoIoTPreferredConnection(SSID, PASS);
```

5 Conclusion

This project successfully implements IoT-enabled text display on an OLED screen using ESP32/NodeMCU and Arduino IoT Cloud. The system enables real-time text updates via a mobile app and web browser both.