

PROGRAM 2

Board & Environment Setup (ESP8266 – NodeMCU)

IDE

- Arduino IDE

Board Package Configuration (One-Time Setup)

1. Open Arduino IDE
2. Go to **File** → **Preferences**
3. In **Additional Board Manager URLs**, add:

```
http://arduino.esp8266.com/stable/package_esp8266com_index.json
```

4. Click **OK**
5. Go to **Tools** → **Board** → **Boards Manager**
6. Search for **ESP8266**
7. Install **esp8266 by ESP8266 Community**

Board & Port Selection

- **Tools** → **Board** → **NodeMCU 1.0 (ESP-12E Module)**
- **Tools** → **Port** → **COMx** (select the detected port)

Required Libraries

Install using **Sketch** → **Include Library** → **Manage Libraries**: - Adafruit GFX Library - Adafruit SSD1306 Library - DHT sensor library (by Adafruit)

Program Title

Temperature and Humidity Monitoring using DHT11 with OLED Display using ESP8266

Components Used

- NodeMCU ESP8266
 - DHT11 Temperature and Humidity Sensor
 - OLED Display (128×64, SSD1306, I2C)
 - Jumper Wires
-

Pin Configuration

DHT11 Sensor

DHT11 Pin	NodeMCU Pin	GPIO	Description
VCC	3.3V	—	Power Supply
DATA	D7	GPIO13	Sensor Data Output
GND	GND	—	Ground

OLED Display (I2C)

OLED Pin	NodeMCU Pin	GPIO
SDA	D2	GPIO4
SCL	D1	GPIO5
VCC	3.3V / Vin	—
GND	GND	—

Working Explanation

1. ESP8266 initializes the DHT11 sensor and OLED display
2. DHT11 measures temperature and humidity
3. Sensor sends digital data to ESP8266 via DATA pin
4. ESP8266 reads sensor values using DHT library
5. If values are invalid, error message is shown
6. Valid temperature and humidity values are displayed on OLED
7. Readings are updated every 2 seconds

Code (With Self-Explanatory Comments)

```
#include <Arduino.h>           // Core Arduino / ESP8266 functions
#include <Wire.h>               // I2C communication
#include <Adafruit_GFX.h>       // Graphics library
#include <Adafruit_SSD1306.h>   // OLED driver
#include <DHT.h>               // DHT sensor library

#define SCREEN_WIDTH 128
#define SCREEN_HEIGHT 64
```

```

// Create OLED display object
Adafruit_SSD1306 display(SCREEN_WIDTH, SCREEN_HEIGHT, &Wire, -1);

// DHT sensor configuration
#define DHTPIN D7           // GPIO13
#define DHTTYPE DHT11      // Sensor type

DHT dht(DHTPIN, DHTTYPE);

void setup() {
  Serial.begin(115200);    // Start serial communication
  dht.begin();             // Initialize DHT sensor

  // Initialize OLED display
  if (!display.begin(SSD1306_SWITCHCAPVCC, 0x3C)) {
    while (true);         // Stop execution if OLED fails
  }

  display.clearDisplay();
  display.setTextColor(SSD1306_WHITE);
}

void loop() {
  // Read humidity and temperature values
  float humidity = dht.readHumidity();
  float temperature = dht.readTemperature();

  // Check for sensor read failure
  if (isnan(humidity) || isnan(temperature)) {
    display.clearDisplay();
    display.setTextSize(1);
    display.setCursor(0, 0);
    display.print("DHT Error");
    display.display();
    delay(2000);
    return;
  }

  // Display temperature and humidity
  display.clearDisplay();
  display.setTextSize(2);

  display.setCursor(0, 0);
  display.print("T: ");
  display.print(temperature);
  display.print(" C");

  display.setCursor(0, 32);

```

```
display.print("H: ");  
display.print(humidity);  
display.print(" %");  
  
display.display();           // Update OLED screen  
delay(2000);                 // Delay for DHT11 stability  
}
```

Possible Viva / Exam Questions

1. What are the temperature and humidity ranges of DHT11?
2. Why is a delay required between DHT readings?
3. What does `isnan()` check?
4. Difference between DHT11 and DHT22?
5. Why is DHT sensor communication handled using a library?
6. Why does OLED require `display.display()` to update?