**Guide**

**1. Installing Arduino**

If you already have the Arduino software installed, that’s great. You only need to install the dependencies (refer to the **Installing Dependencies** section). If not, you can follow this YouTube tutorial:  
<https://www.youtube.com/watch?v=3awCkLS7gHI>

**Installing dependencies**

Once installation is complete, always run the Arduino IDE as an administrator. This ensures that the application has access to all connected devices, such as the ESP8266.

Next, open the Library Manager (the book symbol on the left panel) and install the following libraries by searching for their names (in **bold** letters):

A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a device

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

After installation, click on the **Verify** (checkmark) button at the top of the IDE. This confirms that the correct libraries have been installed. The output should resemble the following:

A screenshot of a computer program

AI-generated content may be incorrect.

Don’t worry about red text in the output; this is normal and expected each time you upload code to the ESP8266. If the output differs from the expected result, please reach out for assistance.

**2. Test mode**

While developing the code, I included a **Test Mode**, which prevents the program from sending data to ThingSpeak. This helps prevent excessive data logging and assists in troubleshooting wiring connections between the ESP8266 and sensors.

A variable named **testMode** is present in the code and is set to **true** by default:

A screenshot of a computer program

AI-generated content may be incorrect.

This activates the sensors without transmitting data to ThingSpeak. If there are issues when activating the circuit or if a sensor stops working, set this variable to **true** to prevent unnecessary data logging while troubleshooting the circuit.

However, if you are ready to use the circuit, change the value to **false**:

**bool testMode = false;**

**3. Changing SSID and Password fields in code:**

To connect to ThingSpeak, the circuit requires the **SSID** and **password** of the Wi-Fi network. Input these credentials in the following fields within the code:

**const char\* ssid = "Your\_WiFi\_SSID";**

**const char\* password = "Your\_WiFi\_Password";**

To verify the SSID of your modem:

1. Open **Wi-Fi Settings**.

A screenshot of a computer

AI-generated content may be incorrect.

1. A screenshot of a computer

   AI-generated content may be incorrect.Click on **Properties**.:

3. **Viewing output from Arduino.**

You can view the output of what you’re Arduino sends by Going to **Tools > Serial Monitor:**

**A white rectangular object with a black border

AI-generated content may be incorrect.**

A new window should appear beside the output tab named **Serial monitor**. Click on that and it should show a white blank screen.

Make sure that the device is connected or else this error appears:  
A screenshot of a computer

AI-generated content may be incorrect.

If connected, then logs from the Arduino should appear in the Serial Monitor.

Also check that the baud rate of the serial monitor is set to **115200:**

A screenshot of a computer program

AI-generated content may be incorrect.

The baud rate can be seen in line 35:

A screen shot of a computer program

AI-generated content may be incorrect.

**3. Wiring Diagram**

If in any case that some of the pins in the Arduino gets disconnected, here are the connection wiring diagram pinouts for each component:

**Wiring Diagram:**

**ESP8266 Pins:**

**3.3V → Power for sensors and LCD (if compatible)**

**GND → Common ground for all components**

**SDA (D2 on NodeMCU) → SDA of ADS1115 and LCD**

**SCL (D1 on NodeMCU) → SCL of ADS1115 and LCD**

**Digital Pins → For DHT22 (e.g., D4 on NodeMCU)**

**ADS1115 Connections:**

**VDD → 5V (depending on the ADS1115 module)**

**GND → GND**

**SCL → SCL (D1 on NodeMCU)**

**SDA → SDA (D2 on NodeMCU)**

**A0 → MQ137 Analog Output**

**A1 → MQ136 Analog Output**

**DHT22 Connections:**

**VCC → 5V**

**GND → GND**

**Data → Digital Pin (D7 on NodeMCU)**

**16x2 LCD Connections (I2C):**

**VCC → 5V**

**GND → GND**

**SDA → SDA (D2 on NodeMCU)**

**SCL → SCL (D1 on NodeMCU)**

This is also available at the very end of the code, commented out for your reference.

**Uploading the code:**

If you have modified the code and would like to add changes to the code. First, click on the checkmark to verify if the code you edited doesn’t contain any errors:

A screenshot of a computer

AI-generated content may be incorrect.

If there are no errors, then the output window should output a similar output:  
A screenshot of a computer program

AI-generated content may be incorrect.

In any case that the output window provides a different output, then please send a screenshot of the error to me and let me know as well any changes you have made in the code. So that I can assist you.

In the case that there are no errors, then plug in the ESP8266 to your machine/computer/laptop using the white micro USB cable and ensure that the **Generic ESP8266 Module** text becomes **highlighted.**

Similar to this output:

A screen shot of a computer

AI-generated content may be incorrect.

This indicates that the machine/computer/laptop recognizes the ESP8266 module. If this doesn’t occur, then please let me know.

Once connected, Press the arrow button to upload the code to the esp8266 module. During the upload, **do not disconnect the ESP8266 module from the machine**.

Once done uploading, this mini tab should appear indicating the successful upload:  
A screenshot of a computer

AI-generated content may be incorrect.

You can now disconnect the ESP8266 module from the machine.