

IoT SensorS & NodeMCU

Rain/Moisture/MQ-135 (air quality)/MQ-3 (Alcohol) /Pressure/ADXL135/FIRE/TOUCH/VIBRATION/SOUND/MOISTURE/DHT-11/LDR/HALL/LM35/OPTOCOUPLER/4-LED/4-DIP SW



**Emtron Technologies   
Flat No-101, B3 Wing, 1st Floor, Divyam Hights, Gilbert Hill, Shreenath Nagar, Andheri –West, Mumbai-58   
+91-9820962870 E-mail:** [**emtron.tech@gmail.com**](mailto:emtron.tech@gmail.com)**, www.emtrontech.in**

Lab Manual

**List of Experiments**

1. ARDUINO Tool Installation and Library configuration
2. A Blue LED is connected to GPIO pin of NodeMCU GPIO 0 ESP-12E. write a program to ON & OFF the LED
3. To print the GPIO values from “D4” pin of NodeMCU in Serial Terminal (Digital Output)
4. To print the analog values “A0” from 10-bit, 10MHz ADC pin of NodeMCU in Serial Terminal (Analog Output)
5. Control an LED GPIO0 from Web Browser
6. To print the values of ADC on a Website using the NodeMCU
7. To Plot the values on ThingSpeak using NodeMCU
8. To show the values of Temperature and Humidity on serial terminal using DHT11 sensor
9. To show the values of Temperature and Humidity on ThingSpeak using DHT11 Sensor
10. a & b: To Communicate between two Desktops using Only NodeMCU's Comm. Station Client Mode (ST) & Comm. Accesspoint Router (AP) Mode

**Image Processing using CMOS Camera**

1. Here we Connect NodeMCU as a Station client Mode (ST) and the Mobile (or Wi-Fi Laptop ) Hotspot as a Access Point router mode (AP). And to display the image grabbed from camera connected to the NodeMCU on Mobile/Laptop browser.
2. Here we create an Access Point i.e. AP router mode on NodeMCU (HotSpot). Other devices (Mobile etc) to can connect in Station client mode (ST).

**EX1: Installation of Arduino compiler and ESP8266 library &**

**EX2: A Blue LED is connected to GPIO-0 pin of NodeMCU ESP-12E. write a program to ON & OFF the LED**

1. Install : arduino-1.8.5-windows.exe
2. File -> Preferences -> Settings-> Additional Boards URL-> type the following -> OK

<http://arduino.esp8266.com/versions/2.3.0/package_esp8266com_index.json>

1. Tools -> Board -> BoardManager -> Select ESP8266 -> Install {Internet is required}

4. Next open File -> Examples -> ESP8266 -> Blinky

5. Tools -> Select Port

6. Tools-> Select the Board-> NodeMCU1.0 (ESP-12E Module)

7. Verify and Load the Program

8. You will notice Blue LED blinking

**EX3: To Read GPIO values from D4 pin of NodeMCU and Print in Serial Terminal (Digital Output)**

**EX4: To convert Analog to Digital form and display on Serial Monitor**

This program development is to use 10-bit ADC of NodeMCU which is connected to “A0” pin of the microcontroller and display the data using NodeMCU Serial Monitor

File ->Open-> Serial\_ADC.ino

1. Tools -> Select Port
2. Tools-> Select the Board-> NodeMCU1.0 (ESP-12E Module)
3. Compile and Load the Program
4. Open Tools-> Serial Monitor (baudrate 15200)-> Reset NodeMCU
5. You will notice ADC data is displayed on monitor

## EX5: To Control an LED from Web Browser

We will see how to "Turn On and Turn Off" an LED that has connected to the Esp8266, the esp8266 has programmed from Arduino IDE to control the LED through web browser.  
  
We will connect the Esp8266 to the Wi-Fi router using SSID and password of our Home network Wi-Fi , where the esp8266 connect to our Wi-Fi and create a web server, which can be accessed by looking through the serial Monitor of the Arduino window or you can also log into your Wifi router and check for the list of clients connected to your Wi-Fi router.

1. File ->Oen-> Webserver\_Blink.ino
2. Tools -> Select Port
3. Tools-> Select the Board-> NodeMCU1.0 (ESP-12E Module)
4. We need SSID and Password of Home (not institute) Network router (OR) Create a Hot Spot on your Mobile Phone and in the code change

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

//Station mode you should put your ssid and password

const char \*ssid = "ysrao"; // Put your SSID here

const char \*password = "12345678"; // Put your PASSWORD here

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

5. Compile and Load the Program

6. Open Tools-> Serial Monitor (baudrate 15200)-> Reset Node Mcu : Server IP Address mentioned

7. You will notice in Mobile Hotspot “ESP\_5E3489” is connected similarly Laptop Wi-Fi “ysrao” is paired

8. On any phone browser (Google Chrome) (or Wi-Fi enabled Laptop) type the IP address

http://----------------------------------------------

to see the Web Page Control LED/ON/OFF buttons

**EX6: To print the values of ADC on a Website using the NodeMCU**

By Default Connect the NodeMCU to Your Mobile/Laptop HotSpot by Changing

const char \*ssid = "SSID\_Username"; // Put your HotSpot Name here

const char \*password = "SSID\_Password"; // Put your HotSpot Password here

After Uploading the Code Immediately open your Serial Terminal and Open

http://IP\_Address\_Shown\_There

Refresh Everytime to See a New Value (Control + R)

**EX7: To plot the graph on ThingSpeak (Firstly Create Account on Thingspeak) by some numerical data**

ThingSpeak is an IoT platform that uses channels to store data sent from apps or devices. With the settings described in [Channel Configurations](https://in.mathworks.com/help/thingspeak/channel-settings.html), you create a channel, and then send and retrieve data to and from the channel. You can make your channels public to share data. Using the REST API calls such as GET, POST, PUT, and DELETE, you can create a channel and update its feed, update an existing channel, clear a channel feed, and delete a channel.

Note: Internet connection is required for this program

**#1🡪 Make an account on ThingSpeak ( go to** [**https://thingspeak.com/**](https://thingspeak.com/) **)by signing with Gmail ID or make a new**

**#2🡪 Go to Channels -> create new, Make sure it's private channel**

**#3🡪 GO to Channel Settings-> Select the Field1 (give some name) and Save channel**

**#4🡪 Select API Keys -> copy Write API key and paste it in your “C” code**

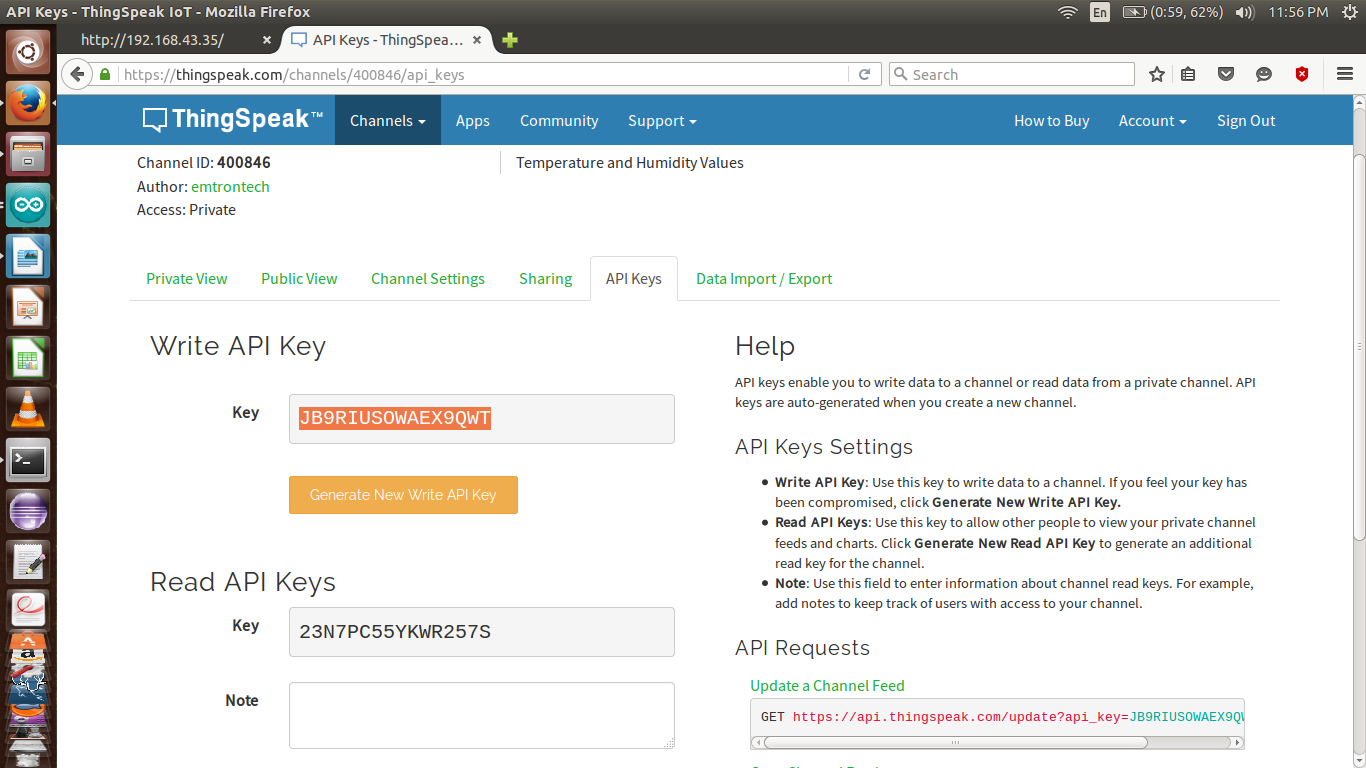
**#5🡪 Private View -> to see the updated data**

**#6🡪 GO to Channel Settings-> for clear & Delete channel**

**EX8: To print the values of Temperature and Humidity on Serial Terminal using DHT11 Sensor**

Do the Following before Compiling

1. Sketch ----> Include Library -------> Manage Library -------> Search DHT and Install the one by Adafruit Version : 1.3.0
2. Copy the folder: Adafruit\_Sensor in to -> C:\Program Files (x86)\Arduino\libraries
3. Connect DHT sensor to D4 of NodeMCU

**EX9: Aim : To Show the values of Temperature and Humidity on ThingSpeak using DHT11 Sensor (Firstly Create Account on Thingspeak)**

String apiKey = "\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*"; // Enter your Write API key from ThingSpeak as shown above

By Default Connect the NodeMCU to Your Mobile/Laptop HotSpot by Changing

const char \*ssid = "SSID\_Name";

const char \*pass = "SSID\_Password";

**EX10: Communication Between two NodeMCU's ( two Desktops using NodeMCU's)**

Comm Server

Create Access point (Hotspot)

AP\_ssid

AP\_password

With a Unique Hotspot Name and Password and IP address

Comm Client

ssid

pass

Change this with the AP\_ssid and AP\_password and IP address you have set in Comm Server