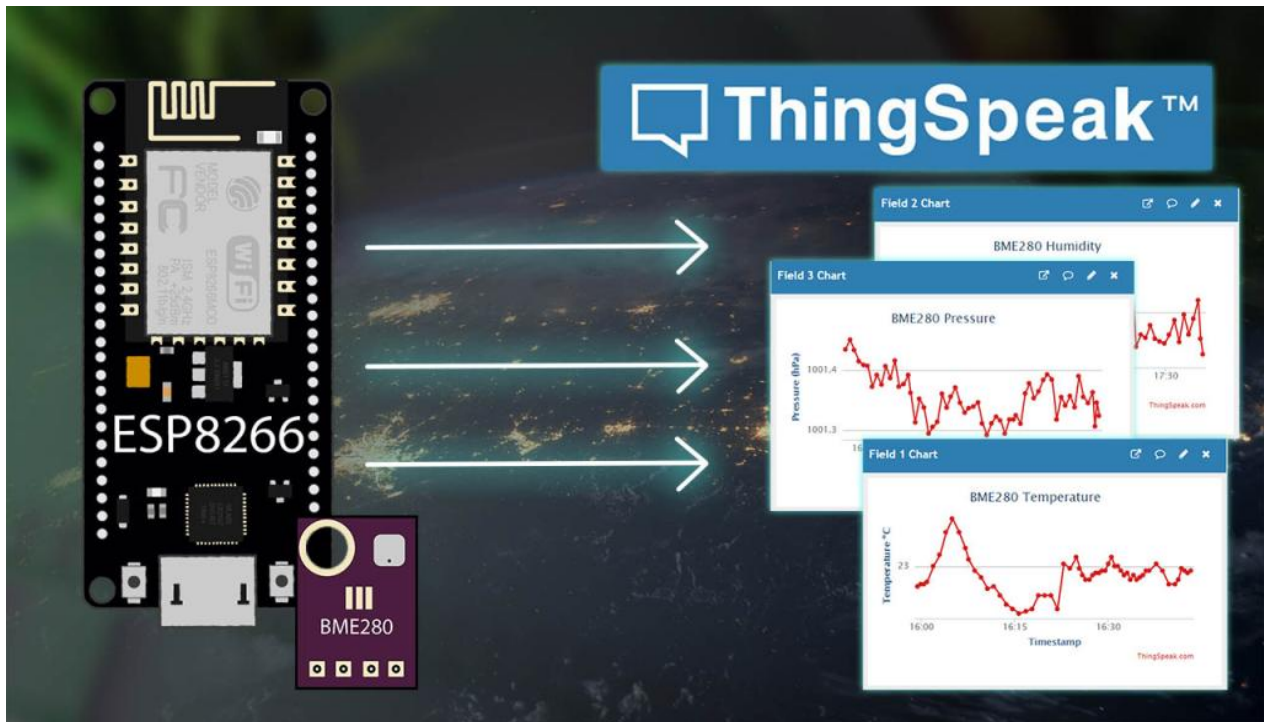


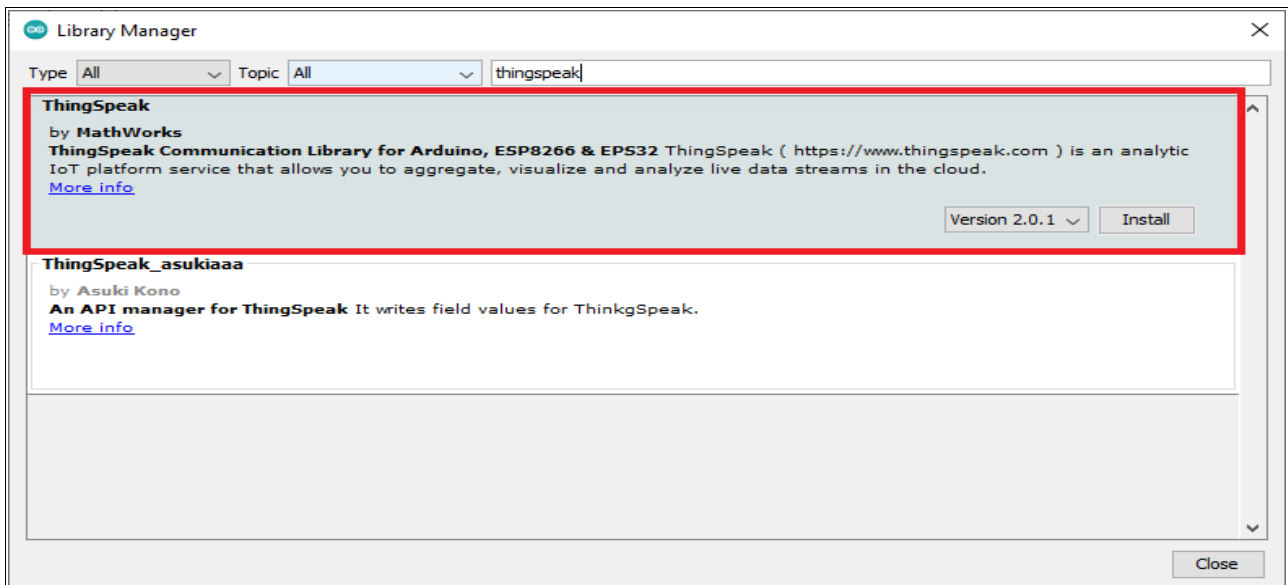
Practical -8

Objective: Create program to connect Node MCU with the WiFi networks and connect with Thingspeak cloud environment & uploading sensor data to thingspeak.

Introduction: ThingSpeak allows you to publish your sensor readings to their website and plot them in charts with timestamps. Then, you can access your readings from anywhere in the world.



Installing the ThingSpeak Library: To send sensor readings to ThingSpeak, we'll use the thingspeak-arduino library. You can install this library through the Arduino Library Manager. Go to **Sketch > Include Library > Manage Libraries...** and search for **"ThingSpeak"** in the Library Manager. Install the ThingSpeak library by MathWorks.



Installing BME280 Libraries:

- Adafruit_BME280 library
- Adafruit_Sensor library

You can install the libraries using the Arduino Library Manager. Go to **Sketch** > **Include Library** > **Manage Libraries** and search for the library name.

Installing Libraries (VS Code + PlatformIO):

If you're using VS Code with the PlatformIO extension, copy the following to the **platformio.ini** file to include the libraries.

Building the Circuit

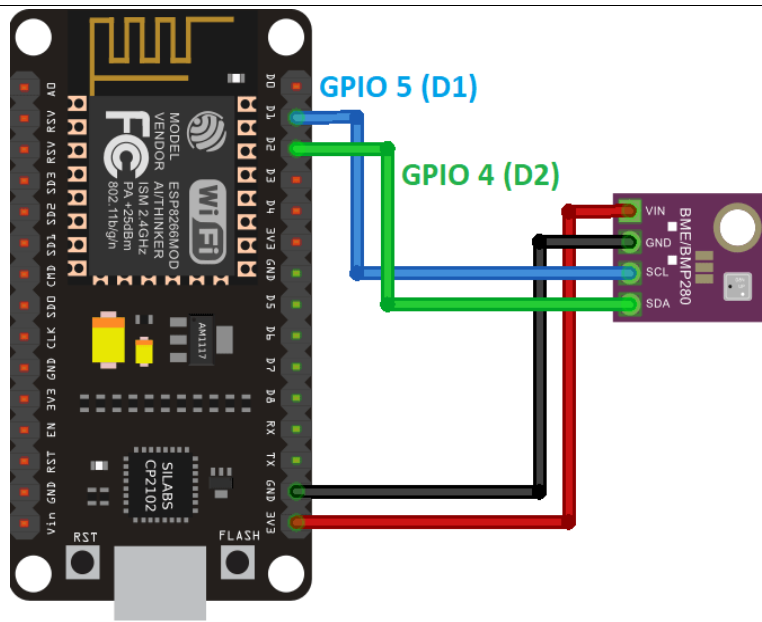
To exemplify how to send data to ThingSpeak, we'll send sensor readings from a BME280 sensor. So, you need to wire a BME280 sensor to your ESP8266.

Parts Required:

To complete this tutorial you need the following parts:

- BME280 sensor module
- ESP8266 (read Best ESP8266 development boards)
- Breadboard
- Jumper wires

Schematic Diagram: We're going to use I2C communication with the BME280 sensor module. For that, wire the sensor to the default ESP8266 SCL (**GPIO 5**) and SDA (**GPIO 4**) pins, as shown in the following schematic diagram.

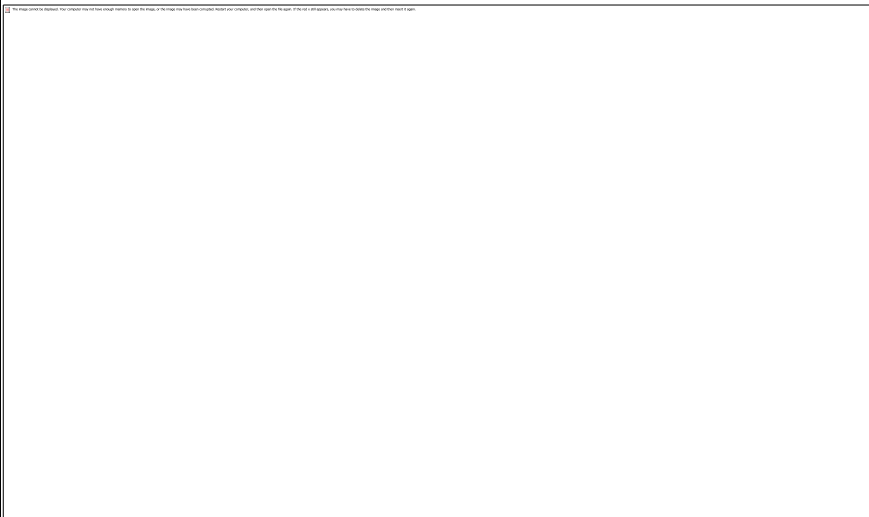


ThingSpeak – Getting Started:

Go to ThingSpeak and click the “**Get Started For Free**” button to create a new account. This account is linked to a Mathworks account. So, if you already have a Mathworks account, you should log in with that account.

Creating New Channel:

After your account is ready, sign in, open the “**Channels**” tab and select “**My Channels**”.



Press the “New Channel” button to create a new channel.

Type a name for your channel and add a description. In this example, we'll just publish temperature. If you want to publish multiple readings (like humidity and pressure), you can enable more fields.

ThingSpeak™ Channels ▾ Apps ▾ Support ▾

New Channel

Name	BME280 Readings
Description	Readings from BME280 (ESP8266)
Field 1	Temperature <input checked="" type="checkbox"/>
Field 2	<input type="checkbox"/>
Field 3	<input type="checkbox"/>

Click the **Save Channel** button to create and save your channel.

Save Channel

Customizing Chart:

The chart can be customized, go to your **Private View** tab and click on the edit icon.

Private View

Public View

Channel Settings

Sharing

API Keys

+ Add Visualizations

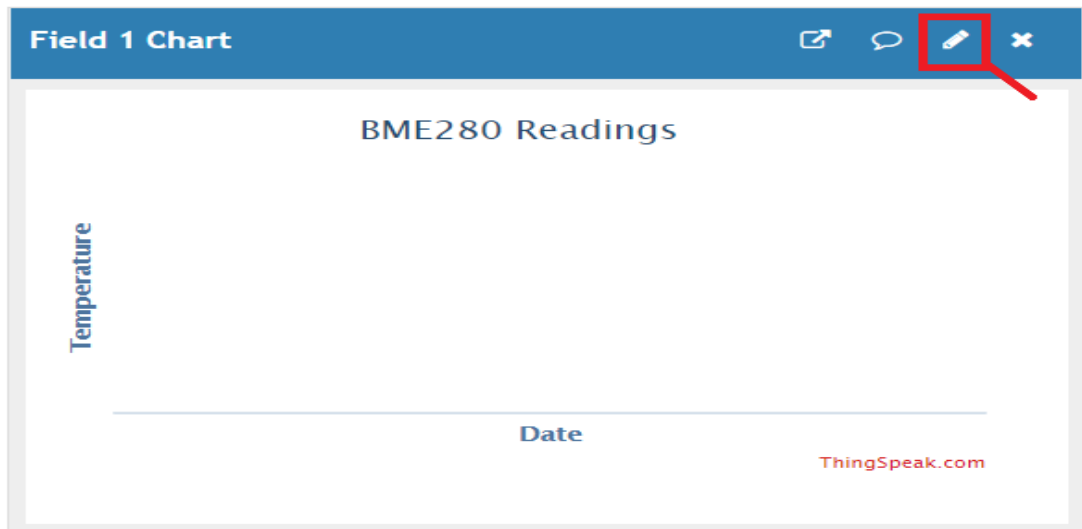
+ Add Widgets

Export recent data

Channel Stats

Created: [about a minute ago](#)

Entries: 0



You can give a title to your chart, customize the background colour, x and y axis, and much more.

Field 1 Chart Options



Title:	<input type="text" value="BME280 Temperature"/>
X-Axis:	<input type="text" value="Timestamp"/>
Y-Axis:	<input type="text" value="Temperature °C"/>
Color:	<input type="text" value="#d62020"/>
Background:	<input type="text" value="#ffffff"/>
Type:	<input type="text" value="line"/>
Dynamic?:	<input type="text" value="true"/>
Days:	<input type="text"/>
Results:	<input type="text" value="60"/>

Timescale:	<input type="text"/>
Average:	<input type="text"/>
Median:	<input type="text"/>
Sum:	<input type="text"/>
Rounding:	<input type="text"/>
Data Min:	<input type="text"/>
Data Max:	<input type="text"/>
Y-Axis Min:	<input type="text"/>
Y-Axis Max:	<input type="text"/>

Save

Cancel

When you're done, press the "Save" button.

API Key:

To send values from the ESP8266 to ThingSpeak, you need the Write API Key. Open the "API Keys" tab and copy the Write API Key to a safe place because you'll need it in a moment.

Private View

Public View

Channel Settings

Sharing

API Keys

Write API Key

Key

HE00X0H.7H4477

Generate New Write API Key

ESP8266 Publish Sensor Readings to ThingSpeak – Code:

```
#include <ESP8266WiFi.h>
#include "ThingSpeak.h"
#include <Adafruit_BME280.h>
#include <Adafruit_Sensor.h>

const char* ssid = "REPLACE_WITH_YOUR_SSID"; // your network SSID (name)
const char* password = "REPLACE_WITH_YOUR_PASSWORD"; // your network password
WiFiClient client;
unsigned long myChannelNumber = X;
const char * myWriteAPIKey = "XXXXXXXXXXXXXXXXXXXX";
// Timer variables
unsigned long lastTime = 0;
unsigned long timerDelay = 30000;
// Variable to hold temperature readings
float temperatureC;
//uncomment if you want to get temperature in Fahrenheit
//float temperatureF;
// Create a sensor object
Adafruit_BME280 bme; //BME280 connect to ESP8266 I2C (GPIO 4 = SDA, GPIO 5 = SCL)
void initBME(){
  if (!bme.begin(0x76)) {
    Serial.println("Could not find a valid BME280 sensor, check wiring!");
    while (1);
  }
}

void setup() {
  Serial.begin(115200); //Initialize serial
  initBME();
  WiFi.mode(WIFI_STA);
  ThingSpeak.begin(client); // Initialize ThingSpeak
}

void loop() {
  if ((millis() - lastTime) > timerDelay) {
    // Connect or reconnect to WiFi
    if(WiFi.status() != WL_CONNECTED){
      Serial.print("Attempting to connect");
      while(WiFi.status() != WL_CONNECTED){
        WiFi.begin(ssid, password);
        delay(5000);
      }
      Serial.println("\nConnected.");
    }
    // Get a new temperature reading
    temperatureC = bme.readTemperature();
    Serial.print("Temperature (°C): ");
    Serial.println(temperatureC);
  }
}
```

```

//uncomment if you want to get temperature in Fahrenheit
/*temperatureF = 1.8 * bme.readTemperature() + 32;
Serial.print("Temperature (°C): ");
Serial.println(temperatureF);*/
// Write to ThingSpeak. There are up to 8 fields in a channel, allowing you to store up to 8 different
// pieces of information in a channel. Here, we write to field 1.
int x = ThingSpeak.writeField(myChannelNumber, 1, temperatureC, myWriteAPIKey);
//uncomment if you want to get temperature in Fahrenheit
//int x = ThingSpeak.writeField(myChannelNumber, 1, temperatureF, myWriteAPIKey);
if(x == 200){
  Serial.println("Channel update successful.");
}
else{
  Serial.println("Problem updating channel. HTTP error code " + String(x));
}
lastTime = millis();
}
}

```

Demonstration: After inserting your network credentials, channel number and API key, upload the code to your board. Open the Serial Monitor at a baud rate of 115200, and press the on-board RST button. After 30 seconds, it should connect to Wi-Fi and start publishing the readings to ThingSpeak.

```

COM3
Channel update successful.
Temperature (°C): 22.41
Channel update successful.
Temperature (°C): 22.43
Channel update successful.
Temperature (°C): 22.43
Channel update successful.
Temperature (°C): 22.42
Channel update successful.
Temperature (°C): 22.42
Channel update successful.
Temperature (°C): 22.41
Channel update successful.
Temperature (°C): 22.44
Channel update successful.

```

Autoscroll
 Show timestamp
Newline
115200 baud
Clear output