INTERNET OF THINGS

ACTIVITY - 6

PREPARED BY

Garv Baheti

(2020BTechCSE031)

SUBMITTED TO

Mr. Divanshu Jain

JK LAKSHMIPAT UNIVERSITY

ज्ञानम् अमृतम्

JKLU

NAAC 'A' Grade Accredited

Department of Computer Science Engineering
Institute of Engineering and Technology
JK Lakshmipat University Jaipur

November 07, 2022

OBJECTIVE

Interface BME sensor with NodeMCU and upload the pressure, temperature and humidity data using suitable dashboard.

Hardware

1. ESP8266 NodeMCU board
2. BME280 Sensor
3. Connecting Wires
4. Breadboard

Wiring

The connection of BME280 with the ESP8266 NodeMCU is very easy. We have to connect the VCC terminal with 3.3V, ground with the ground (common ground), SCL of the sensor with SCL of the module, and SDA of the sensor with the SDA pin of the ESP modules.

The I2C pin in ESP8266 NodeMCU for SDA is GPIO5 (D1) and for SCL is GPIO4 (D2).

SETTING UP THINGSPEAK ACCOUNT FOR LM35 SENSOR:

Before we proceed towards construction details we need to setup our thingspeak account correctly to receive the sensor data, this procedure need to be done for all three methods mentioned here.

You can sign up for *thingspeak account here*, if you haven't signed up yet.

- **You need to create a new channel** by clicking "New Channel" button and in the channel.

Ultra-Sonic Sensor Data to Thingspeak

1) Go to Channel settings tab and edit the name.

2) Enable Field 1 by checking the box and write the label as "Distance".

3) Scroll down and click save.

- Now click on API keys tab:

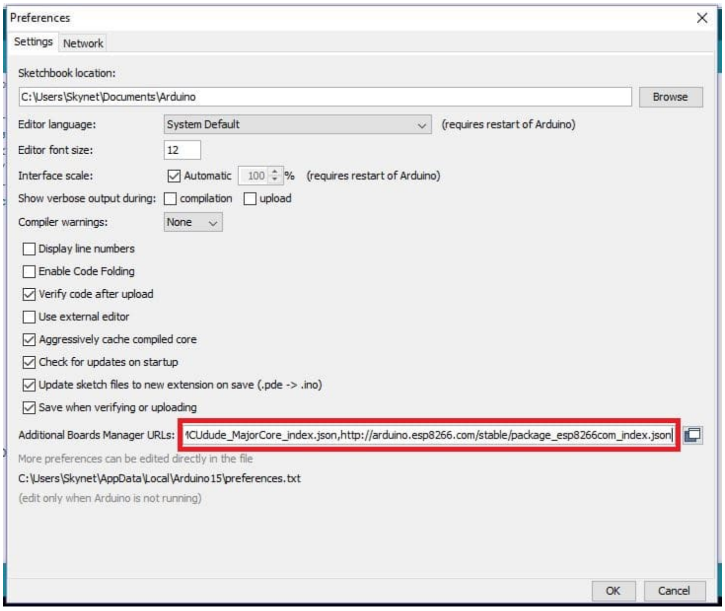
API keys are the access keys to your channel using which you can write and read values. In this project we are using only write API key which is already generated for your channel. You need to take note of it and this write API key need to be inserted to the given program codes.

- Now by clicking “Private View” tab you will see a blank channel. You will see some data once we send to it.

Installing ESP8266 package to Arduino IDE:

This process needs to be done if you are using NodeMCU or generic ESP8266 to connect to thingspeak. If you are going to use GSM modem to send LM35 data, this step is irrelevant for you, but **for NodeMCU and ESP8266 boards this step is mandatory.**

- In this step you need internet; we are going to install core files to Arduino IDE for IoT based boards.
- Copy this link: http://arduino.esp8266.com/stable/package_esp8266com_index.json
- Now open Arduino IDE and click on “**File**” >> “**Preferences**”.
- A window will open like this:



Sketchbook location: C:\Users\Skynet\Documents\Arduino

Editor language: System Default (requires restart of Arduino)

Editor font size: 12

Interface scale: ☑ Automatic 100 % (requires restart of Arduino)

Show verbose output during: □ compilation □ upload

Compiler warnings: None

□ Display line numbers

□ Enable Code Folding

☑ Verify code after upload

□ Use external editor

☑ Aggressively cache compiled core

☑ Check for updates on startup

☑ Update sketch files to new extension on save (.pde -> .ino)

☑ Save when verifying or uploading

Additional Boards Manager URLs: 4CUdude_MajorCore_index.json,http://arduino.esp8266.com/stable/package_esp8266com_index.json

More preferences can be edited directly in the file

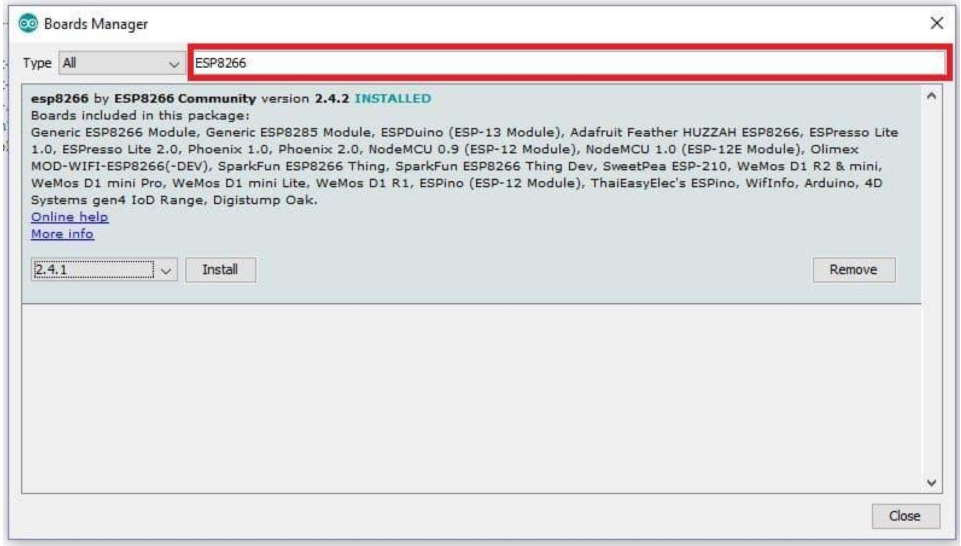
C:\Users\Skynet\AppData\Local\Arduino15\preferences.txt

(edit only when Arduino is not running)

Preferences

- Paste the URL on the box and click “OK”.
- Now go to **tools > Board > Boards Manager**.

- Now a window will popup:



esp8266 by **ESP8266 Community** version **2.4.2** INSTALLED
Boards included in this package:
Generic ESP8266 Module, Generic ESP8285 Module, ESPDuino (ESP-13 Module), Adafruit Feather HUZZAH ESP8266, ESPresso Lite 1.0, ESPresso Lite 2.0, Phoenix 1.0, Phoenix 2.0, NodeMCU 0.9 (ESP-12 Module), NodeMCU 1.0 (ESP-12E Module), Olimex MOD-WIFI-ESP8266(-DEV), SparkFun ESP8266 Thing, SparkFun ESP8266 Thing Dev, SweetPea ESP-210, WeMos D1 R2 & mini, WeMos D1 mini Pro, WeMos D1 mini Lite, WeMos D1 R1, ESPino (ESP-12 Module), ThaiEasyElec's ESPino, WifInfo, Arduino, 4D Systems gen4 IoD Range, Digistump Oak.

boards manager

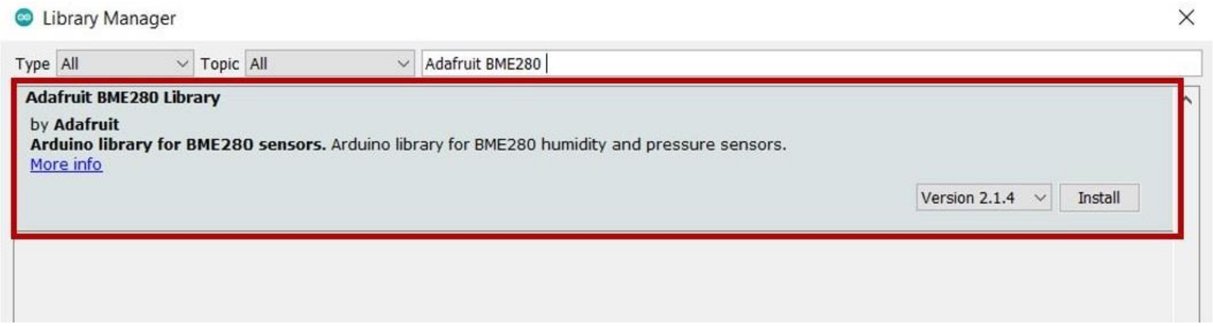
- Type ESP8266 on the box as shown and you will get an installation option, select the latest version and click install.

Installing BME280 Libraries

As we are connecting the BME280 sensor with ESP8266 NodeMCU so we will have to install BME280 libraries to our module. We will require two libraries for this project:

1. Adafruit_BME280 library
2. Adafruit_Sensor library

We will use the Library Manager in our Arduino IDE to install the latest versions of the libraries. Open your Arduino IDE and go to Sketch > Include Libraries > Manage Libraries. Type each library name in the search bar and install them both.



Library Manager

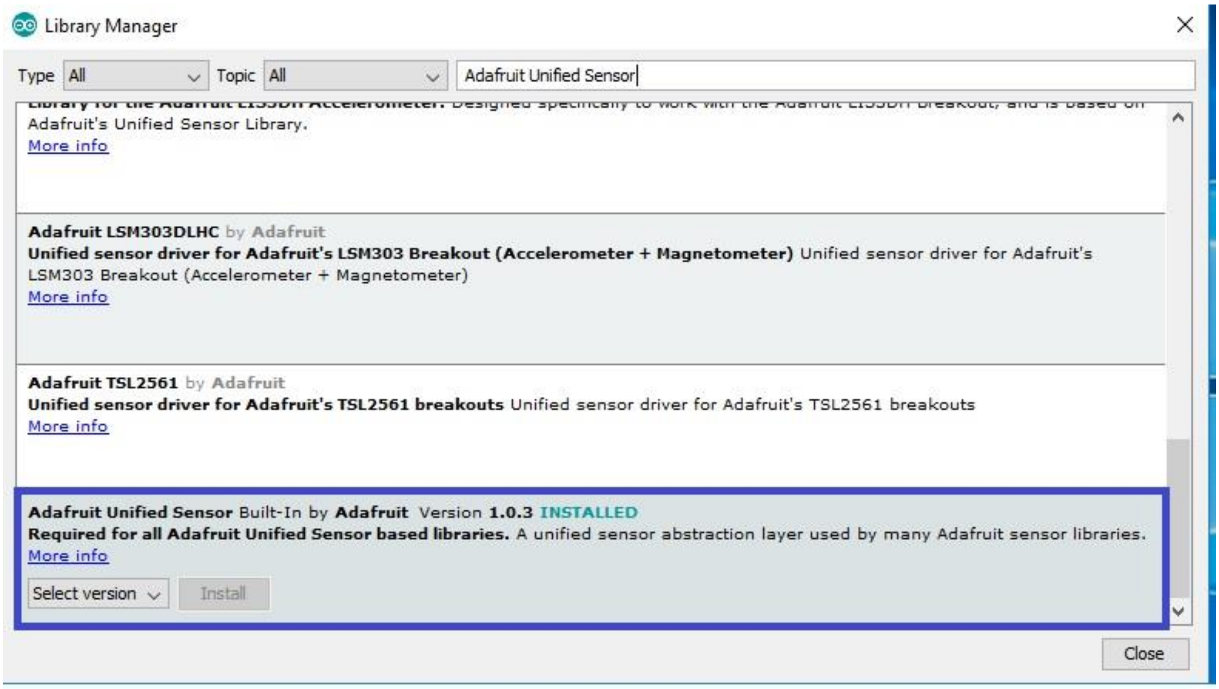
Type All Topic All Adafruit BME280

Adafruit BME280 Library

by **Adafruit**

Arduino library for BME280 sensors. Arduino library for BME280 humidity and pressure sensors.

More info



Library Manager

Type All Topic All Adafruit Unified Sensor

Library for the Adafruit LIS3DH Accelerometer: Designed specifically to work with the Adafruit LIS3DH Breakout, and is based on Adafruit's Unified Sensor Library.

More info

Adafruit LSM303DLHC by Adafruit

Unified sensor driver for Adafruit's LSM303 Breakout (Accelerometer + Magnetometer) Unified sensor driver for Adafruit's LSM303 Breakout (Accelerometer + Magnetometer)

More info

Adafruit TSL2561 by Adafruit

Unified sensor driver for Adafruit's TSL2561 breakouts Unified sensor driver for Adafruit's TSL2561 breakouts

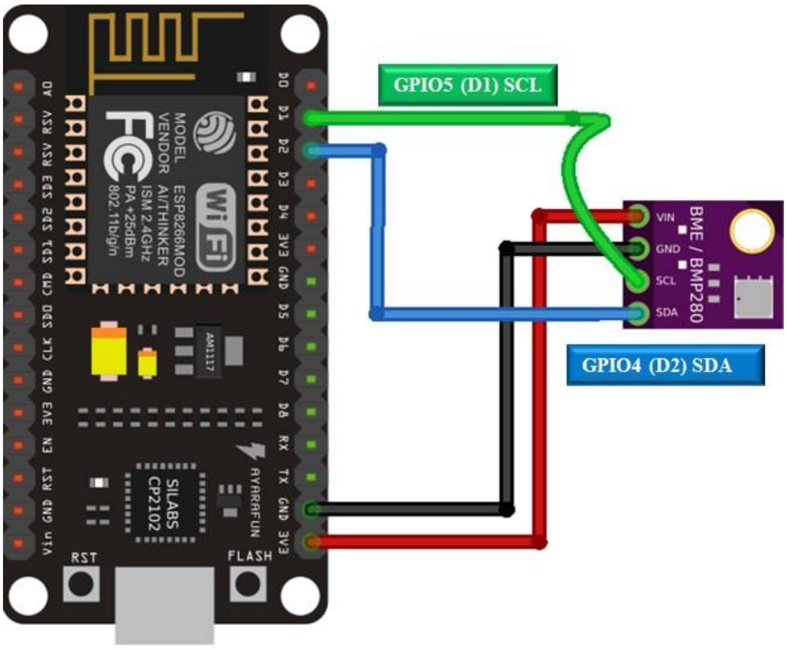
More info

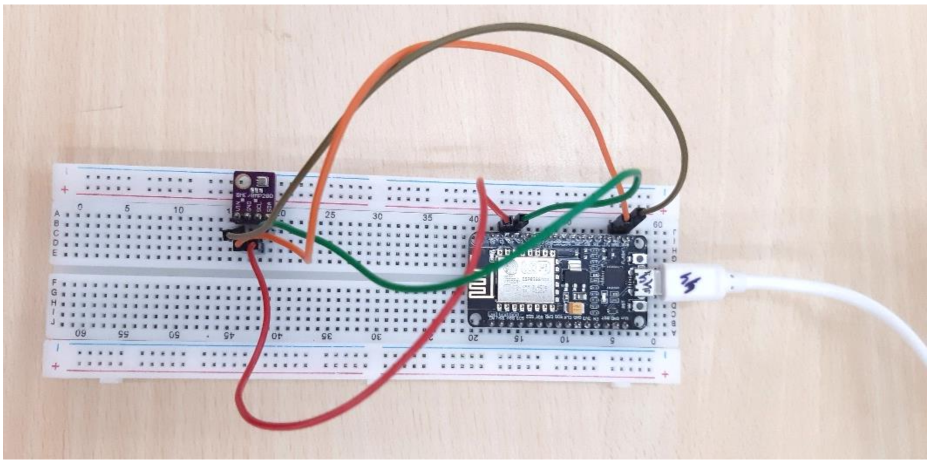
Adafruit Unified Sensor Built-In by **Adafruit** Version **1.0.3** INSTALLED

Required for all Adafruit Unified Sensor based libraries. A unified sensor abstraction layer used by many Adafruit sensor libraries.

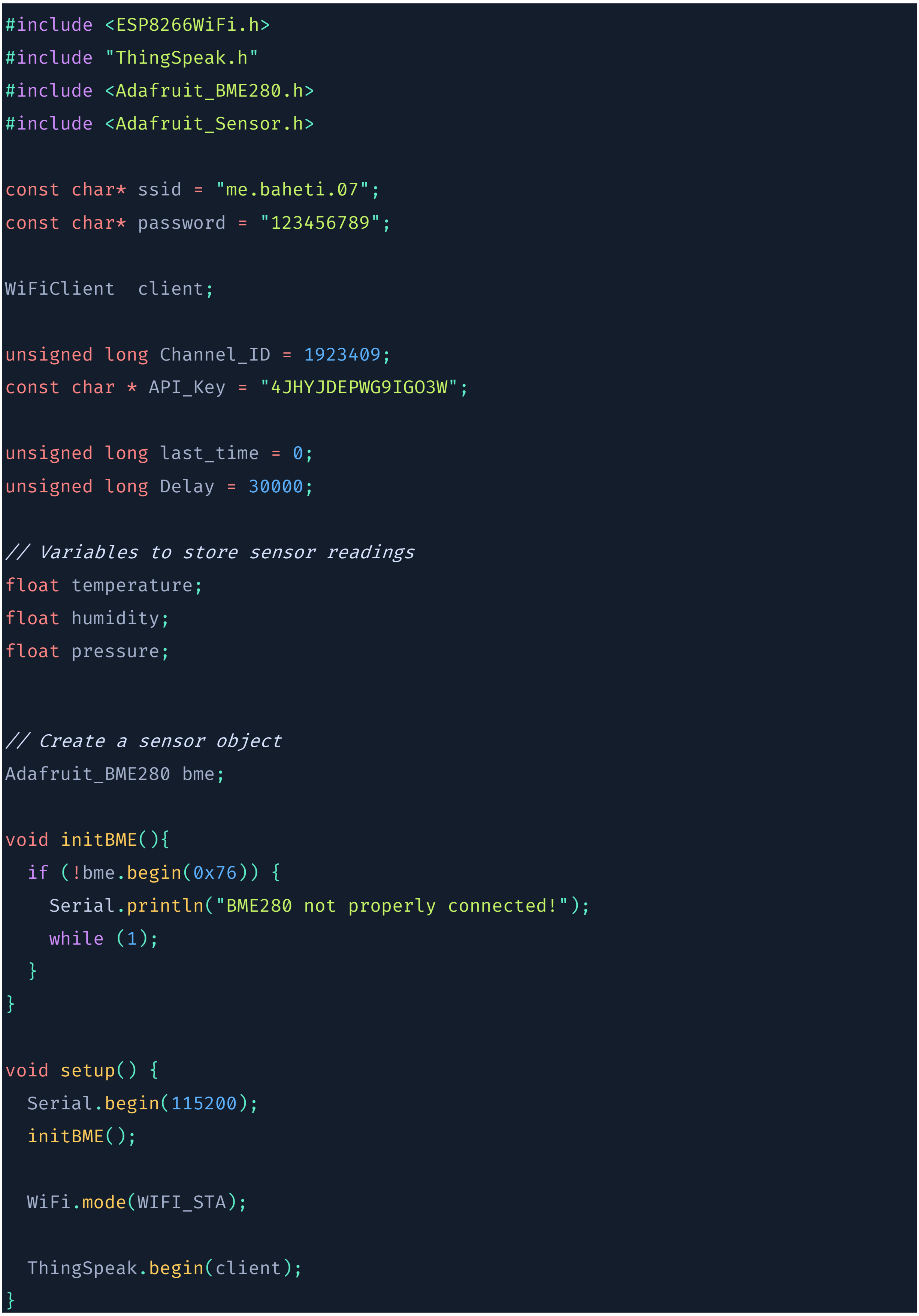
More info

CIRCUIT

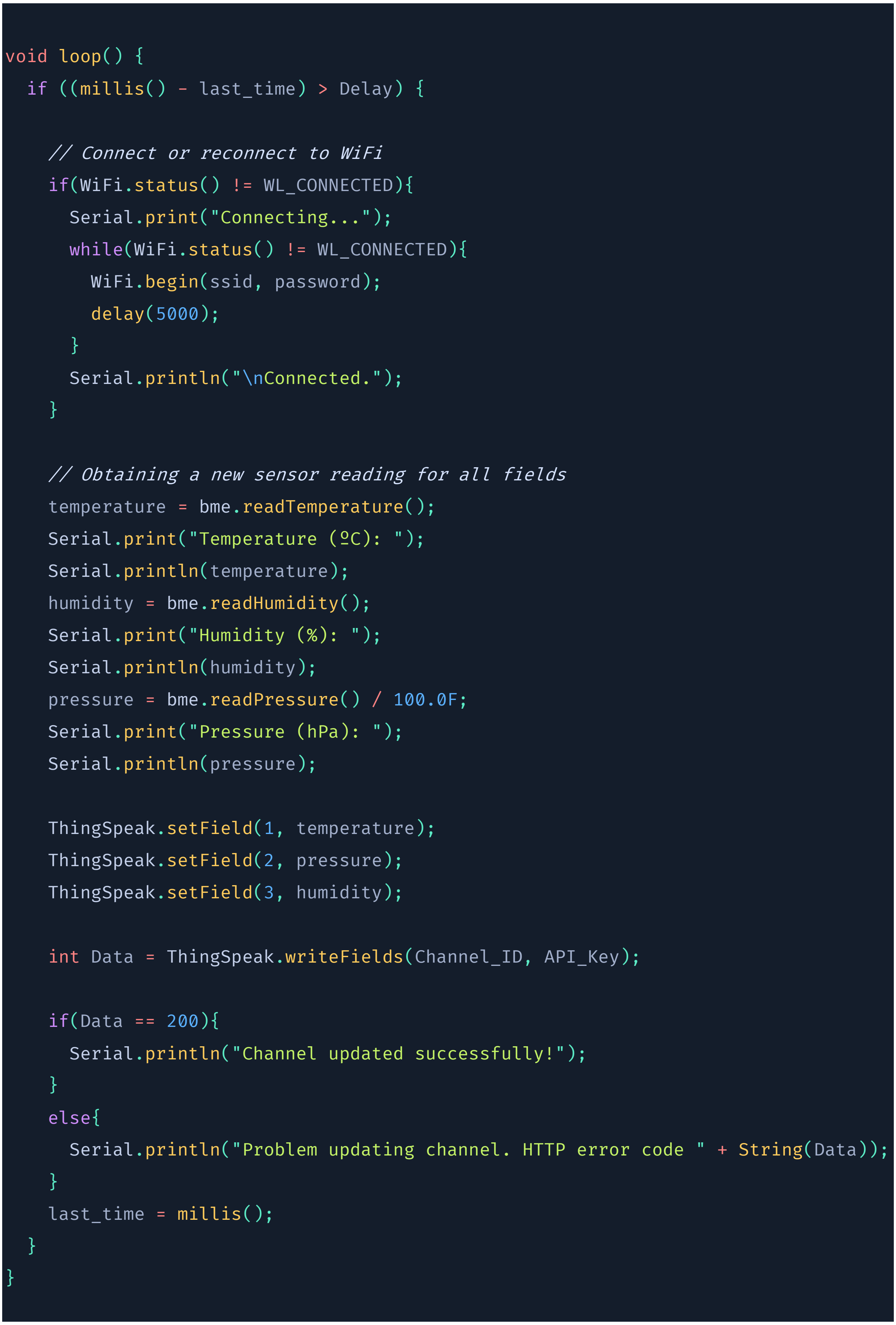

GPIO5 (D1) SCL
GPIO4 (D2) SDA
VIN
GND
SCL
SDA
BME / BMP280
D0
D1
D2
D3
D4
3V3
GND
D5
D6
D7
D8
RX
TX
GND
3V3
A0
RSV
RSV
SD3
SD2
SD1
CMD
SDO
CLK
GND
3V3
EN
RST
GND
Vin
RST
FLASH
MODEL ESP8266MOD
VENDOR AI/THINKER
ISM 2.4GHz
PA +25dBm
802.11b/g/n
SILABS CP2102
AYARAFUN
AM1117



CODE

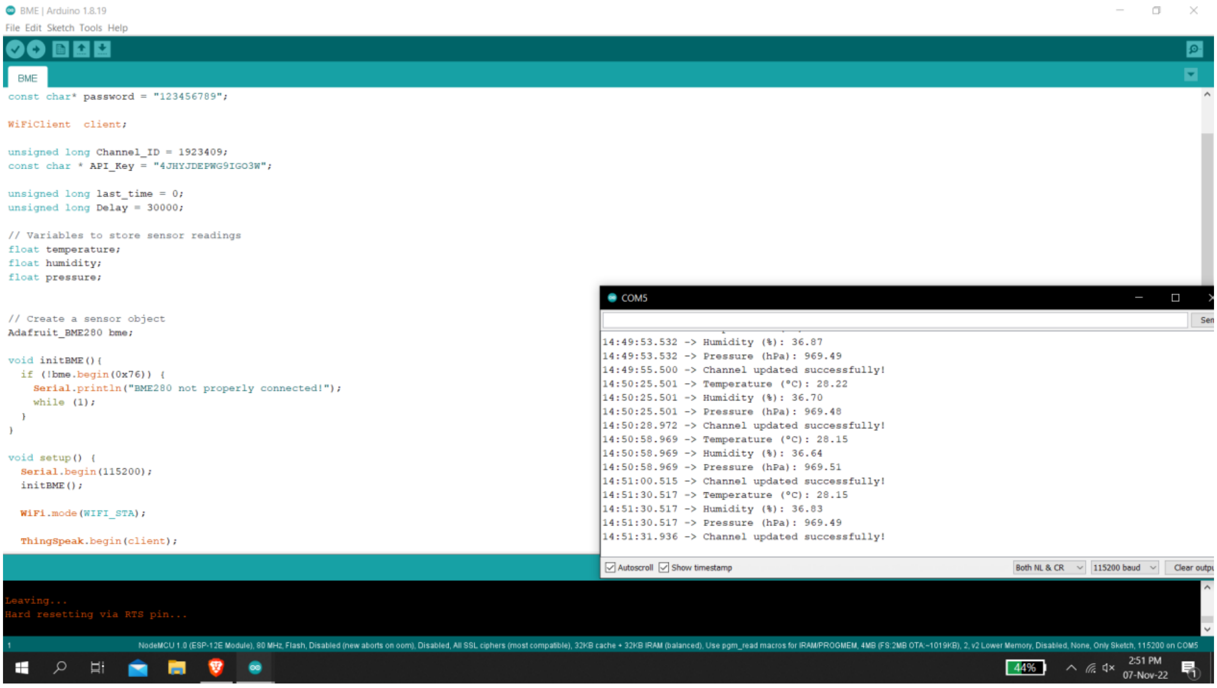


```
#include <ESP8266WiFi.h>
#include "ThingSpeak.h"
#include <Adafruit_BME280.h>
#include <Adafruit_Sensor.h>

const char* ssid = "me.baheti.07";
const char* password = "123456789";

WiFiClient  client;

unsigned long Channel_ID = 1923409;
const char * API_Key = "4JHYJDEPWG9IGO3W";

unsigned long last_time = 0;
unsigned long Delay = 30000;

// Variables to store sensor readings
float temperature;
float humidity;
float pressure;


// Create a sensor object
Adafruit_BME280 bme;

void initBME(){
  if (!bme.begin(0x76)) {
    Serial.println("BME280 not properly connected!");
    while (1);
  }
}

void setup() {
  Serial.begin(115200);
  initBME();

  WiFi.mode(WIFI_STA);

  ThingSpeak.begin(client);
}
```



```
void loop() {
  if ((millis() - last_time) > Delay) {

    // Connect or reconnect to WiFi
    if(WiFi.status() != WL_CONNECTED){
      Serial.print("Connecting...");
      while(WiFi.status() != WL_CONNECTED){
        WiFi.begin(ssid, password);
        delay(5000);
      }
      Serial.println("\nConnected.");
    }

    // Obtaining a new sensor reading for all fields
    temperature = bme.readTemperature();
    Serial.print("Temperature (ºC): ");
    Serial.println(temperature);
    humidity = bme.readHumidity();
    Serial.print("Humidity (%): ");
    Serial.println(humidity);
    pressure = bme.readPressure() / 100.0F;
    Serial.print("Pressure (hPa): ");
    Serial.println(pressure);

    ThingSpeak.setField(1, temperature);
    ThingSpeak.setField(2, pressure);
    ThingSpeak.setField(3, humidity);

    int Data = ThingSpeak.writeFields(Channel_ID, API_Key);

    if(Data == 200){
      Serial.println("Channel updated successfully!");
    }
    else{
      Serial.println("Problem updating channel. HTTP error code " + String(Data));
    }
    last_time = millis();
  }
}
```

OUTPUT



```
const char* password = "123456789";

WiFiClient  client;

unsigned long Channel_ID = 1923409;
const char * API_Key = "4JHYJDEPWG9IGO3W";

unsigned long last_time = 0;
unsigned long Delay = 30000;

// Variables to store sensor readings
float temperature;
float humidity;
float pressure;


// Create a sensor object
Adafruit_BME280 bme;

void initBME(){
  if (!bme.begin(0x76)) {
    Serial.println("BME280 not properly connected!");
    while (1);
  }
}

void setup() {
  Serial.begin(115200);
  initBME();

  WiFi.mode(WIFI_STA);

  ThingSpeak.begin(client);
```

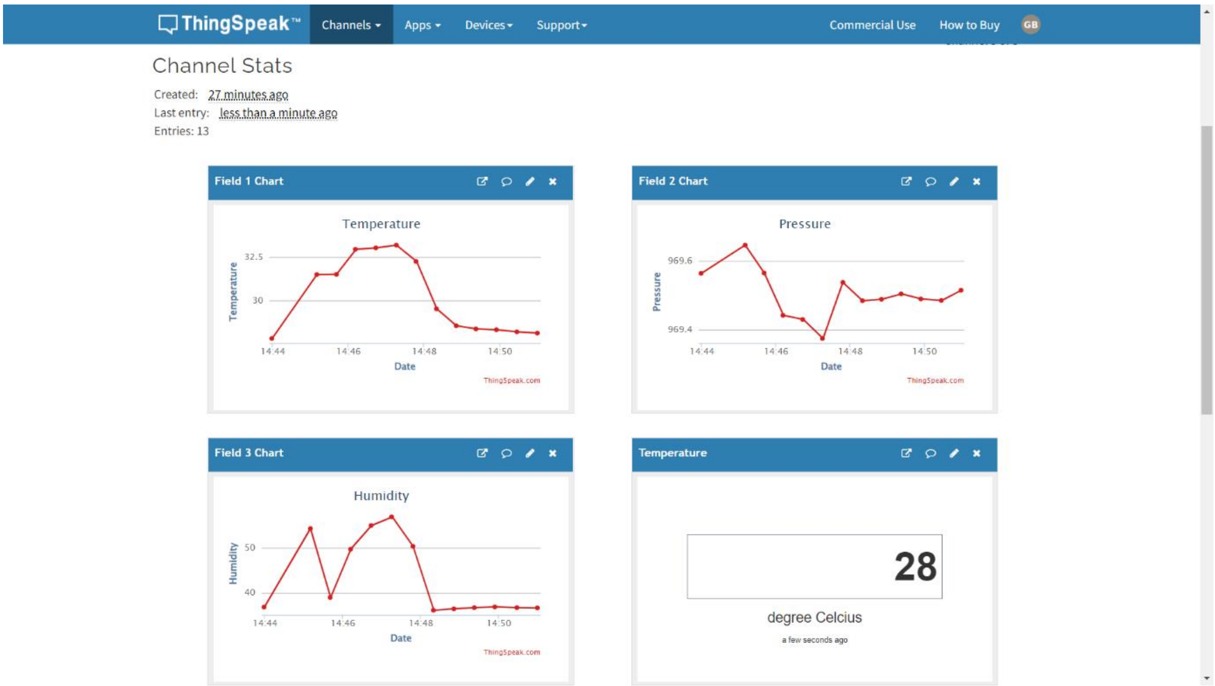
```
14:49:53.532 -> Humidity (%): 36.87
14:49:53.532 -> Pressure (hPa): 969.49
14:49:55.500 -> Channel updated successfully!
14:50:25.501 -> Temperature (°C): 28.22
14:50:25.501 -> Humidity (%): 36.70
14:50:25.501 -> Pressure (hPa): 969.48
14:50:28.972 -> Channel updated successfully!
14:50:58.969 -> Temperature (°C): 28.15
14:50:58.969 -> Humidity (%): 36.64
14:50:58.969 -> Pressure (hPa): 969.51
14:51:00.515 -> Channel updated successfully!
14:51:30.517 -> Temperature (°C): 28.15
14:51:30.517 -> Humidity (%): 36.83
14:51:30.517 -> Pressure (hPa): 969.49
14:51:31.936 -> Channel updated successfully!
```

☑ Autoscroll ☑ Show timestamp Both NL & CR 115200 baud Clear outpu

```
Leaving...
Hard resetting via RTS pin...
```

1 NodeMCU 1.0 (ESP-12E Module), 80 MHz, Flash, Disabled (new aborts on oom), Disabled, All SSL ciphers (most compatible), 32KB cache + 32KB IRAM (balanced), Use pgm_read macros for IRAM/PROGMEM, 4MB (FS:2MB OTA:~1019KB), 2, v2 Lower Memory, Disabled, None, Only Sketch, 115200 on COM5

THINGSPEAK DASHBOARD

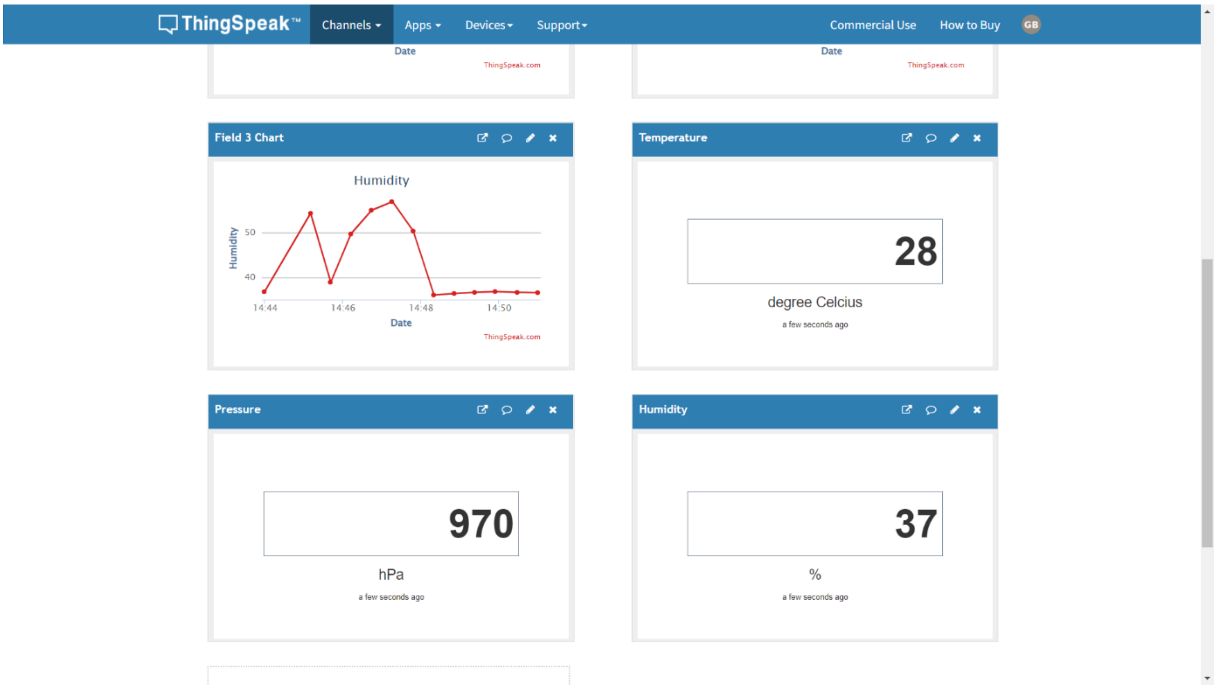


Channel Stats

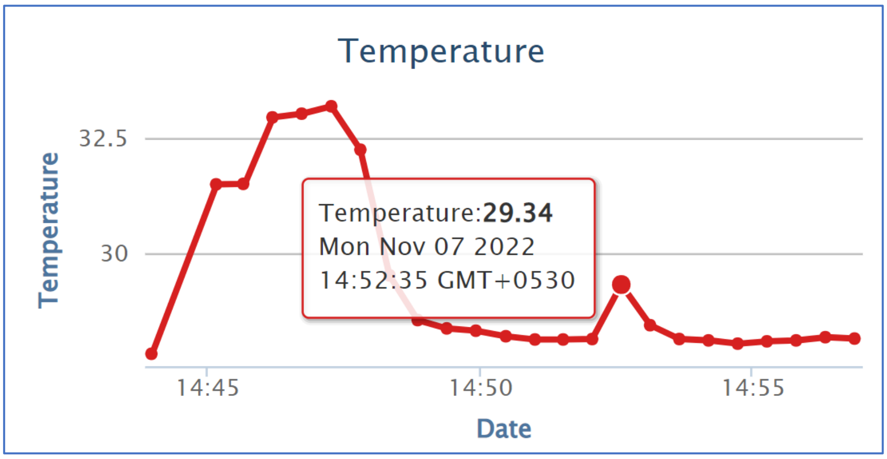
Created: 27 minutes ago

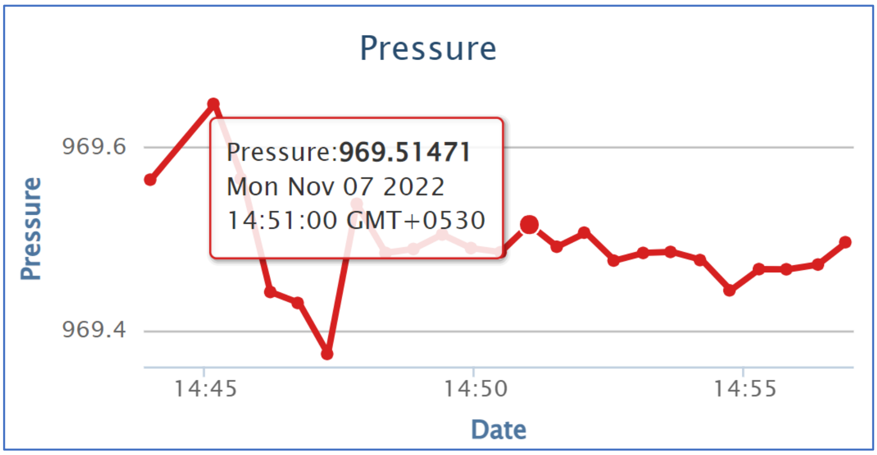
Last entry: less than a minute ago

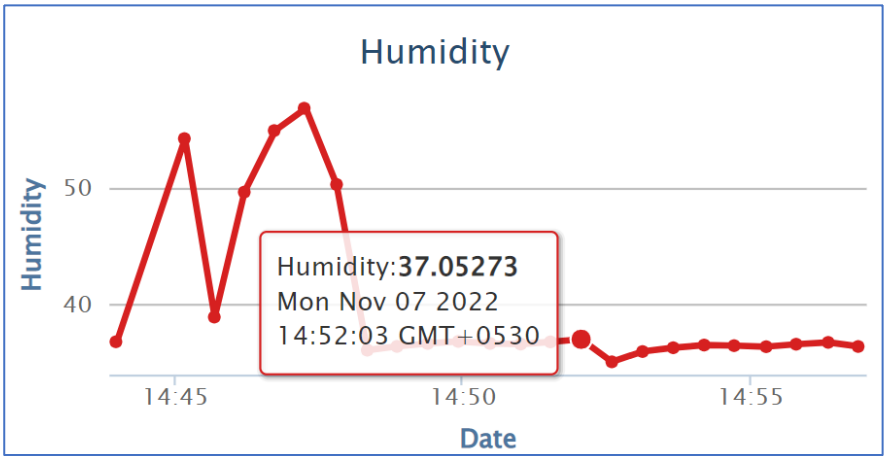
Entries: 13


ThingSpeak™
Channels
Apps
Devices
Support
Commercial Use
How to Buy
GB
Date
ThingSpeak.com
Date
ThingSpeak.com
Field 3 Chart
Humidity
Humidity
50
40
14:44
14:46
14:48
14:50
Date
ThingSpeak.com
Temperature
28
degree Celcius
a few seconds ago
Pressure
970
hPa
a few seconds ago
Humidity
37
%
a few seconds ago

GRAPHS


Temperature
32.5
30
Temperature
Temperature:29.34
Mon Nov 07 2022
14:52:35 GMT+0530
14:45
14:50
14:55
Date


Pressure
969.6
969.4
Pressure
Pressure:969.51471
Mon Nov 07 2022
14:51:00 GMT+0530
14:45
14:50
14:55
Date


Humidity
50
40
Humidity
Humidity:37.05273
Mon Nov 07 2022
14:52:03 GMT+0530
14:45
14:50
14:55
Date