**Introduction to IOT**

**LAB ACTIVITY –** **3**

**Institute of Engineering and Technology**

**JK Lakshmipat University, Jaipur**

**SUBMITTED TO**

Mr. Divanshu Jain

**SUBMITTED BY**

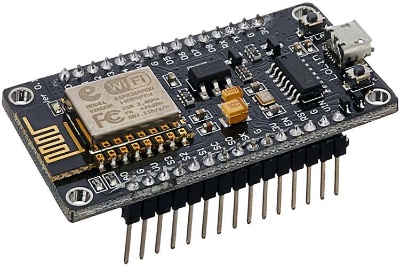
Garv Baheti

2020BTechCSE031

**September 8, 2022**

**OBJECTIVE**

WAP to upload the data of LM35 on ThingSpeak server. Also glow the lamp on server as temp go above some threshold. Also show the gauge value and plot the graph for last 20 readings.

**THEORY**

NodeMCU is an open-source Lua based firmware and development board specially targeted for IoT based Applications. It includes firmware that runs on the ESP8266 Wi-Fi SoC from Espressif Systems, and hardware which is based on the ESP-12 module. The **NodeMCU ESP8266 development board** comes with the ESP-12E module containing the ESP8266 chip having Tensilica Xtensa 32-bit LX106 RISC microprocessor. This microprocessor supports RTOS and operates at 80MHz to 160 MHz adjustable clock frequency. NodeMCU has 128 KB RAM and 4MB of Flash memory to store data and programs. Its high processing power with in-built Wi-Fi / Bluetooth and Deep Sleep Operating features make it ideal for IoT projects. NodeMCU can be powered using a Micro USB jack and VIN pin (External Supply Pin). It supports UART, SPI, and I2C interface.





**NodeMCU ESP8266 Specifications & Features**

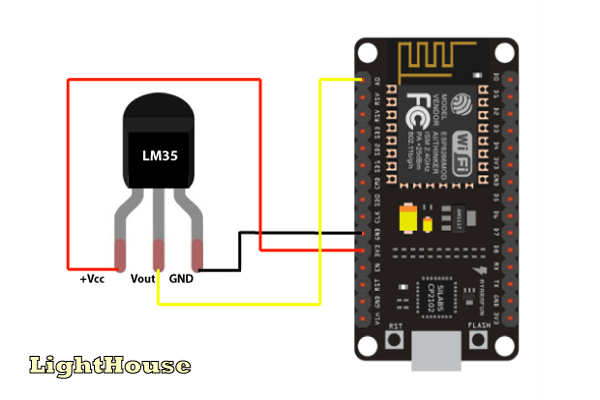
* Microcontroller: Tensilica 32-bit RISC CPU Xtensa LX106
* Operating Voltage: 3.3V
* Input Voltage: 7-12V
* Digital I/O Pins (DIO): 16
* Analog Input Pins (ADC): 1
* UARTs: 1
* SPIs: 1
* I2Cs: 1
* Flash Memory: 4 MB
* SRAM: 64 KB
* Clock Speed: 80 MHz
* USB-TTL based on CP2102 is included onboard, Enabling Plug n Play
* PCB Antenna
* Small Sized module to fit smartly inside your IoT projects

**NodeMCU Development Board Pinout Configuration**

|  |  |  |
| --- | --- | --- |
| **Pin Category** | **Name** | **Description** |
| Power | Micro-USB, 3.3V, GND, Vin | **Micro-USB:** NodeMCU can be powered through the USB port  **3.3V:** Regulated 3.3V can be supplied to this pin to power the board  **GND:** Ground pins  **Vin:**External Power Supply |
| Control Pins | **EN, RST** | The pin and the button resets the microcontroller |
| Analog Pin | A0 | Used to measure analog voltage in the range of 0-3.3V |
| GPIO Pins | GPIO1 to GPIO16 | NodeMCU has 16 general purpose input-output pins on its board |
| SPI Pins | SD1, CMD, SD0, CLK | NodeMCU has four pins available for SPI communication. |
| UART Pins | TXD0, RXD0, TXD2, RXD2 | NodeMCU has two UART interfaces, UART0 (RXD0 & TXD0) and UART1 (RXD1 & TXD1). UART1 is used to upload the firmware/program. |
| I2C Pins |  | NodeMCU has I2C functionality support but due to the internal functionality of these pins, you have to find which pin is I2C. |

**REQUIREMENTS**

* NodeMCU x 1
* Micro USB cable x 1
* PC x 1
* Software Arduino IDE(version 1.6.4+)
* LM35 Sensor
* Account on ThingSpeak.com

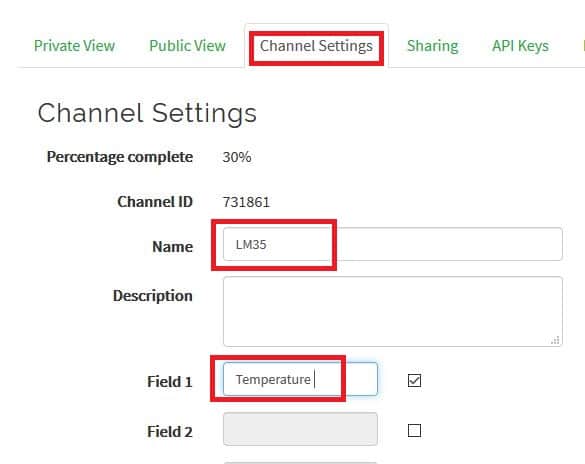


**SETTING UP THINGSPEAK ACCOUNT FOR LM35 SENSOR:**

Before we proceed towards construction details we need to setup our thingspeak account correctly to receive LM35 temperature data, this procedure need to be done for all three methods mentioned here.

You can sign up for *[thingspeak account here](https://thingspeak.com/users/sign_up)*, if you haven’t signed up yet.

* **You need to create a new channel** by clicking “New Channel” button and in the channel, you should do the changes as shown:



LM35 Data to Thingspeak

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

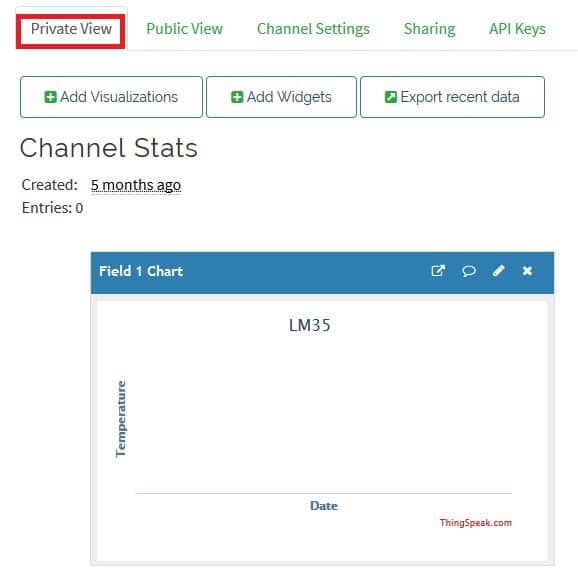
2) Enable Field 1 by checking the box and write the label as “Temperature”.

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.

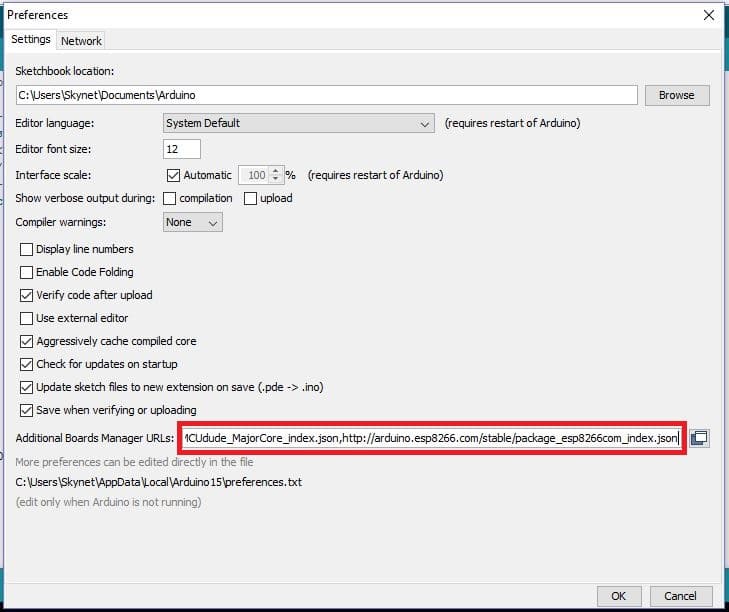


LM35 Data to Thingspeak

**Installing ESP8266 package to Arduino IDE:**

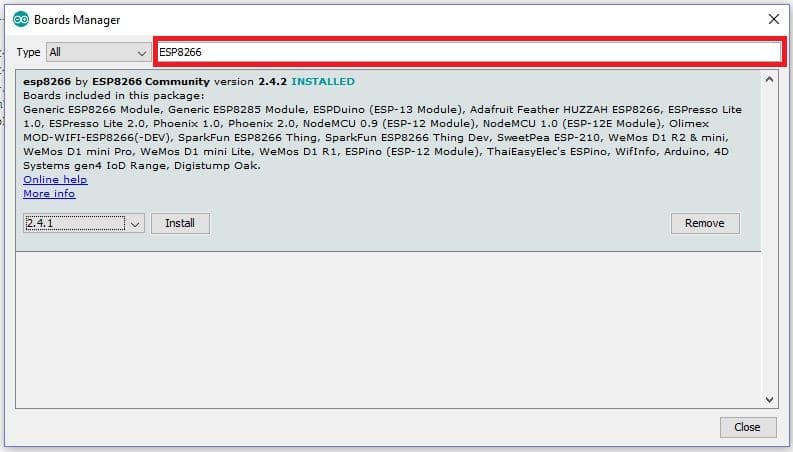
This process needs to be done if you are using NodeMCU or generic ESP8266 to connect to thingspeak. If you are going to use GSM modem to send LM35 data, this step is irrelevant for you, but **for NodeMCU and ESP8266 boards this step is mandatory.**

* 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:



Preferences

* Paste the URL on the box and click “OK”.
* Now go to **tools > Board > Boards Manager**.
* Now a window will popup:



boards manager

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

**CODE**

#include "ThingSpeak.h"

#include <ESP8266WiFi.h>

const int LM35 = A0;

//----------- Enter you Wi-Fi Details---------//

char ssid[] = "Galaxy M314A54"; //SSID

char pass[] = "kvkv6984"; // Password

//-------------------------------------------//

WiFiClient client;

unsigned long myChannelNumber = 1849469; // Channel ID here

const int FieldNumber = 1;

const char \* myWriteAPIKey = "1CK911OJZPU3JKDQ"; // Your Write API Key here

void setup()

{

Serial.begin(9600);

WiFi.mode(WIFI\_STA);

ThingSpeak.begin(client);

}

void loop()

{

if (WiFi.status() != WL\_CONNECTED)

{

Serial.print("Attempting to connect to SSID: ");

Serial.println(ssid);

while (WiFi.status() != WL\_CONNECTED)

{

WiFi.begin(ssid, pass);

Serial.print(".");

delay(5000);

}

Serial.println("\nConnected.");

}

int ADC;

float temp;

ADC = analogRead(LM35); /\* Read Temperature \*/

temp = (ADC \* 3); /\* Convert adc value to equivalent voltage \*/

temp = (temp / 10); /\* LM35 gives output of 10mv/°C \*/

Serial.print("Temperature = ");

Serial.print(temp);

Serial.println(" \*C");

delay(1000);

ThingSpeak.writeField(myChannelNumber, FieldNumber, temp, myWriteAPIKey);

delay(1000);

}

**RESULT**

Temperature above 37degree Celsius >> Lamp ON

Graphical user interface, application

Description automatically generated

Temperature below 37degree Celsius >> Lamp OFF

Graphical user interface, application

Description automatically generated

PLOTTING GRAPH WITH THE HELP OF EXPORTED CSV FILE