

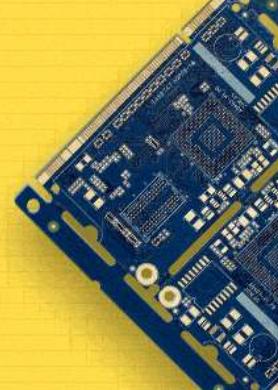
NextPCB

Try Free PCB Prototype

0\$ for 1-4 layer PCB Prototype

- 1-4 layers, green mask, Fr-4
- Fast lead time as 24 hours

Order Now



Arduino Projects IoT Projects

Cellular IoT: Send SIM800/900 GPRS Data to Thingspeak with Arduino



Admin Last Updated: August 21, 2022 18 comments 37,322 views 4 minutes read



In this project we will learn about **IoT using GSM Module**. We will use **SIM900/800 GSM Module** with **Arduino** and send the **DHT11 Sensor** data to **Thingspeak Server**.

Table of Contents [[hide](#)]

- [1 Overview](#)
- [2 Bill of Materials](#)
- [3 SIM900A GSM/GPRS Module
 - \[3.1 SIM900A GSM/GPRS Modem Features\]\(#\)](#)
- [4 Circuit/Connection between GSM Module Arduino & DHT11](#)
- [5 Setting up Thingspeak](#)
- [6 Source Code/Program](#)
- [7 Results & Observations](#)
- [8 Video Tutorial & Explanation](#)

Overview

Nowadays we use **ESP8266** or **ESP32** or any other **wifi Modules** to send any **sensor data** to the Internet wirelessly. Hence **Wifi** comes into action and thus we need **Wifi Connection** for **wireless communication** with any **server**. But the disadvantage of using **Wifi** is, it is not available everywhere. The **wifi signal** is limited to certain locations and to a certain range up to a few meters. For example, in order to use **IoT Connectivity** and to get data from the farmer's fields, we can't rely on **Wifi** due to unavailability. Similarly forest, river zone, mountains are the areas where **wifi connection** is not available.

So, **GSM GPRS** is the only alternative left as per the present scenario and current technology. **GSM GPRS Module** allows you to add **location-tracking, voice, text, SMS, and data** to your application. The big advantage of **GSM/GPRS Connectivity** is, it covers a wide area and signal/connectivity is available almost everywhere.

Làm chủ vóc dáng BodyKey

Văn Mai Hương làm chủ vóc dáng
BodyKey - Dễ mà Vui. Giờ đến

BodyKey

So in this project, we will learn about the **Internet of Things** using **GSM GPRS Module**, i.e Cellular IoT. We will take **SIM900A** as a reference **GSM GPRS Module** and Interface it with Arduino Board. We will sense the surrounding **humidity** and **temperature** using **DHT11 Humidity/Temperature Sensor**. The humidity & temperature data will be sent to **Thingspeak server** using **AT Commands** for GSM Module. **ThingSpeak server** is an open data platform and API for the Internet of Things that enables you to collect, store, analyze, visualize, and act on data from sensors. So let us learn all about **GSM Module with Thingspeak & Arduino**.

You can check one of our posts related to a similar system:

1. [Arduino SIM: Cellular Connectivity Service for Arduino IoT Cloud](#)
2. [Send SIM800/900 GSM/GPRS Data to Thingspeak with STM32](#)

If you want to send the data in **JSON format** using any **API Server** you can check our post here: [SIM900/800 HTTP post request in JSON Format with Arduino](#)

Bill of Materials

Following are the components required for learning the **Cellular IoT project**. All the components can be easily purchased from Amazon.

The component purchase link is given below.

S.N.	COMPONENTS	DESCRIPTION	QUANTITY	
1	Arduino UNO	Arduino UNO R3 Development Board	1	https://amzn.to/3LqfDyA
2	GSM Module	SIM800/SIM900 GSM GPRS Module	1	https://amzn.to/3LqfDyA
3	DHT11 Sensor	DHT11 Humidity Temperature Sensor	1	https://amzn.to/3LqfDyA
4	12/9V Power Supply	-	1	https://amzn.to/3LqfDyA

5	Connecting Wires	Jumper Wires	10	https://amzn.to/3LqfDyA
6	Breadboard	-	1	https://amzn.to/3LqfDyA



SIM900A GSM/GPRS Module

The **SIM900A** is a readily available GSM/GPRS module, used in many mobile phones and PDA. The module can also be used for developing **IoT (Internet of Things)** and **Embedded Applications**. It works on frequencies **900/ 1800 MHz**. The Modem comes with an **RS232 interface**, which allows you to connect PC as well as a microcontroller with RS232 Chip(MAX232). The baud rate is configurable from **9600-115200** through **AT command**. The **GSM/GPRS Modem** is having internal **TCP/IP** stack to enable you to connect with the internet via GPRS. It is suitable for SMS, Voice as well as DATA transfer applications in the M2M interface.



The onboard Regulated Power supply allows you to connect a wide range of unregulated power supply. Using this modem, you can **make & receive audio calls, Send & Read SMS, GPRS Internet, etc** through simple **AT commands**.

Làm chủ vóc dáng với Bc

Văn Mai Hương làm chủ vóc dáng mỏ
Dễ mà Vui. Giờ đến lượt bạn.

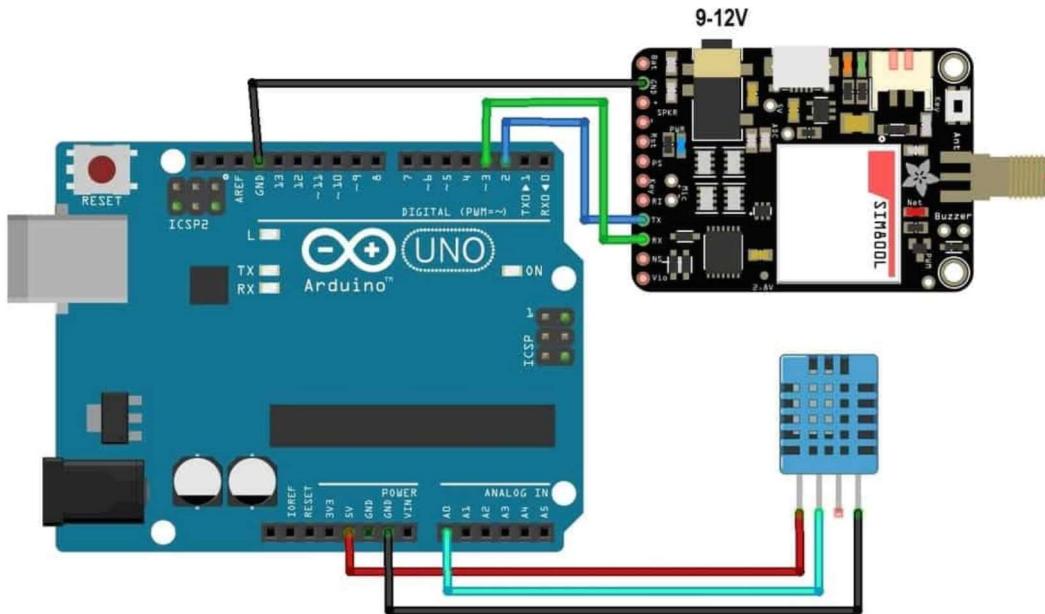
BodyKey

SIM900A GSM/GPRS Modem Features

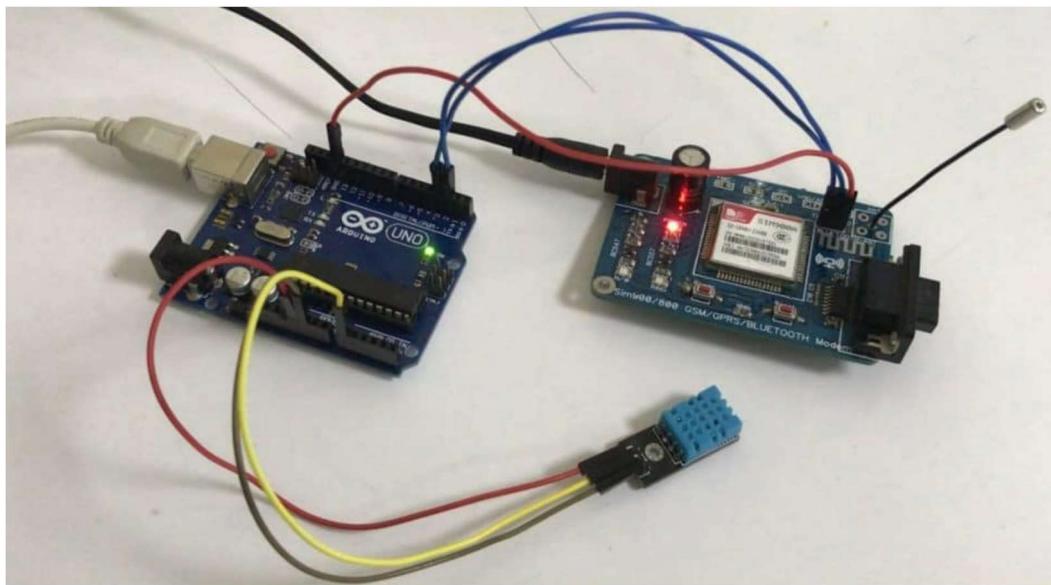
1. *Input Voltage: 12V DC*
 2. *Supports MIC, Audio Input & Speakers*
 3. *Dual-Band GSM/GPRS 900/ 1800 MHz*
 4. *RS232 interface for direct communication with computer or MCU kit*
 5. *Configurable baud rate*
 6. *Wire Antenna (SMA connector with GSM Antenna Optional)*
 7. *SIM Cardholder*
 8. *Built-in Network Status LED*
 9. *Inbuilt Powerful TCP/IP protocol stack for internet data transfer over GPRS*
01. *DATA GPRS: download transfer max is 85.6KBps, Upload transfer max 42.8KBps*

Circuit/Connection between GSM Module Arduino & DHT11

Here is a circuit diagram for interfacing SIM800/900 GSM Module with Arduino & DHT11 for Sending data to Thingspeak Server. The connection is fairly simple as shown below.



SIM800/900 is a **UART Module**. We use the Software Serial command for Serial Communication. So, connect its TX & RX Pins to Arduino pin 2 & 3 respectively. Supply the GSM Module with **9V/12V Power supply**. Similarly, connect the VCC & GND Pin of DHT11 Sensor to Arduino 5V & GND. Connect its output pin to A0 of Arduino.





Cùng BodyKey làm chủ vóc dáng

Mở

Setting up Thingspeak

ThingSpeak provides very good tool for IoT based projects. By using ThingSpeak site, we can monitor our data and control our system over the Internet, using the Channels and web pages provided by ThingSpeak. So first you need to sign up for ThingSpeak. So visit <https://thingspeak.com> and create an account.



Then create a new channel and set up what you want. The tutorial in the video below. Follow the video for more clarifications.

The screenshot shows the 'Channel Settings' section of the ThingSpeak web interface. A new channel is being created with the following details:

- Percentage complete:** 30%
- Channel ID:** 1002353
- Name:** DHT11 Humidity Temperature
- Description:** (empty)
- Field 1:** Temperature (checked)
- Field 2:** Humidity (checked)
- Field 3:** (unchecked)
- Field 4:** (unchecked)

On the right, there is a detailed description of the channel settings and a list of tips:

- Percentage complete:** Calculated based on data entered into the various fields of a channel. Enter the name, description, location, URL, video, and tags to complete your channel.
- Channel Name:** Enter a unique name for the ThingSpeak channel.
- Description:** Enter a description of the ThingSpeak channel.
- Field:** Check the box to enable the field, and enter a field name. Each ThingSpeak channel can have up to 8 fields.
- Metadata:** Enter information about channel data, including JSON, XML, or CSV data.
- Tags:** Enter keywords that identify the channel. Separate tags with commas.
- Link to External Site:** If you have a website that contains information about your channel, enter the URL here.

Then create the API keys. This key is required for programming modifications and setting your data.

The screenshot shows the 'API Keys' section of the ThingSpeak web interface. It includes:

- Write API Key:** A form where a new key ('013AOCHYYNU2LQ19') has been generated and is displayed.
- Help:** A section explaining what API keys are used for (writing data to a channel or reading data from a private channel) and noting they are auto-generated when a new channel is created.
- API Keys Settings:** A list of instructions for using write and read API keys.
- Read API Keys:** A form where an existing key ('W1QRX0K73W1PZHY2') is listed.
- API Requests:** Examples of API requests for writing to a channel feed and reading a channel feed.

Now click on channels so that you can see the online data streaming

Source Code/Program

The source code/program to Send GSM SIM800/900 GPRS Data to Thingspeak with Arduino is given below. No libraries are required. You can simply copy the code and upload to the Arduino Board.

But before that, you have to make two changes in the code. Change your APN Services first. In my case, I am using Airtel SIM and the default APN for Airtel is *airtelgprs.com*. You can check your SIM APN from your google or cellular provider.

```
gprsSerial.println("AT+CSTT=\\"airtelgprs.com\\\""); //start task and setting the APN,
delay(1000);
```

You also need to make changes to the API key from thingspeak.

```
String str="GET https://api.thingspeak.com/update?api_key=013AOCHYYNU2LQ19&field1=" + String(t) +"&field2="+String(h);
Serial.println(str);
```

```
#include <SoftwareSerial.h>
SoftwareSerial gprsSerial(2,3);

#include <String.h>
#include <DHT.h>

#define DHTPIN A0

DHT dht(DHTPIN, DHT11);

void setup()
{
    gprsSerial.begin(9600);                      // the GPRS baud rate
    Serial.begin(9600);      // the GPRS baud rate
    dht.begin();

    delay(1000);
}

void loop()
{
    float h = dht.readHumidity();
    float t = dht.readTemperature();
    delay(100);

    Serial.print("Temperature = ");
    Serial.print(t);
```

```
Serial.println(" °C");
Serial.print("Humidity = ");
Serial.print(h);
Serial.println(" %");

if (gprsSerial.available())
    Serial.write(gprsSerial.read());

gprsSerial.println("AT");
delay(1000);

gprsSerial.println("AT+CPIN?");
delay(1000);

gprsSerial.println("AT+CREG?");
delay(1000);

gprsSerial.println("AT+CGATT?");
delay(1000);

gprsSerial.println("AT+CIPSHUT");
delay(1000);

gprsSerial.println("AT+CIPSTATUS");
delay(2000);

gprsSerial.println("AT+CIPMUX=0");
delay(2000);

ShowSerialData();

gprsSerial.println("AT+CSTT=\\"airtelgprs.com\\\"");
delay(1000);

ShowSerialData();

gprsSerial.println("AT+CIICR");//bring up wireless connection
delay(3000);

ShowSerialData();

gprsSerial.println("AT+CIFSR");//get local IP address
delay(2000);
```

```
ShowSerialData();

gprsSerial.println("AT+CIPSPRT=0");
delay(3000);

ShowSerialData();

gprsSerial.println("AT+CIPSTART=\\"TCP\\", \"api.thingspeak.com");
delay(6000);

ShowSerialData();

gprsSerial.println("AT+CIPSEND");//begin send data to remote
delay(4000);
ShowSerialData();

String str="GET https://api.thingspeak.com/update?api_key=01
Serial.println(str);
gprsSerial.println(str);//begin send data to remote server

delay(4000);
ShowSerialData();

gprsSerial.println((char)26);//sending
delay(5000);//waitting for reply, important! the time is bas
gprsSerial.println();

ShowSerialData();

gprsSerial.println("AT+CIPSHUT");//close the connection
delay(100);
ShowSerialData();
}

void ShowSerialData()
{
    while(gprsSerial.available()!=0)
        Serial.write(gprsSerial.read());
        delay(5000);

}
```



Results & Observations

Once the code is uploaded to Arduino UNO Board, the module will start responding. You can open the Serial Monitor and see the following responses there.

The screenshot shows the Arduino Serial Monitor window titled "COM10". The window displays a series of AT commands sent to the SIM800 module and its responses. The commands include AT, AT+CPIN?, AT+CREG?, AT+CIICR, AT+CIFSR, AT+CIPSPRT=0, AT+CIPSTART="TCP","api.thingspeak.com","80", and AT+CIPSEND. The module responds with OK for most commands and provides status information like Temperature = 23.80 °C and Humidity = 45.00 %. It also shows the connection details for the Thingspeak API calls. The bottom of the window includes checkboxes for Autoscroll and Show timestamp, and settings for Newline, 9600 baud, and Clear output.

```
Temperature = 23.80 °C
Humidity = 45.00 %
AT
OK
AT+CPIN?
+CPIN: READY
OK
AT+CREG?
+CREG: 0,AT+CSTT="airtelgprs.com"
OK
AT+CIICR
OK
AT+CIFSR
10.90.139.102
AT+CIPSPRT=0
OK
AT+CIPSTART="TCP","api.thingspeak.com","80"
OK
CONNECT OKAT+CIPSEND
GET https://api.thingspeak.com/update?api_key=013AOCHYYNU2LQ19&field1=23.80&field2=45.00
GET https://api.thingspeak.com/update?api_key=013AOCHYYNU2LQ19&
SEND OK
3
Temperature = 23.80 °C
Humidity = 40.00 %
AT+CIPSHUT
SHUT OK
 Autoscroll  Show timestamp
Newline 9600 baud Clear output
```

Similarly, you can open the Thingspeak private view and see the temperature and humidity logged data there.



So this is how you can **Send GSM SIM800/900 GPRS Data to Thingspeak with Arduino.**

If you want to send the data in JSON format using any API Server you can check our post here: [SIM900/800 HTTP post request in JSON Format with Arduino](#)

Video Tutorial & Explanation

Cellular IoT | Send Data to Thingspeak Server using SIM800/900 GSM/GPRS Module

[Watch this video on YouTube.](#)

Làm chủ vóc dáng với BodyKey

Ad BodyKey

18 Comments

Toni

July 27, 2020 at 7:18 PM

excuse me sir, i wanna ask about this line :

gprsSerial.println((char)26);//sending

what is (char)26 mean?

thanks before

Loading...

Reply

GAURAV PIPLANI

August 7, 2020 at 4:10 PM

i have connected the GSM modem to my PC and then i am send commands to the GSM Modem using terminal Software.

This is what i am sending:

AT

AT+CPIN?

AT+CREG?

AT+CGATT?

AT+CIPSHUT

AT+CIPSTATUS

AT+CIPMUX=0

AT+CSTT="www"

AT+CIICR

AT+CIFSR

AT+CIPSTART="TCP","api.thingspeak.com","80"

AT+CIPSEND

GET <http://api.thingspeak.com/update?>

api_key=X4L3QD19Q7YSLK3J&field1=125

#026

When i use GET instruction and then send 026 i get SEND OK from the modem and then CLOSED but thingspeak channel is not updated.

Please help

Loading...

Reply

bwengye duncan

November 18, 2020 at 9:23 PM

T

oni the line means you are terminating the message being sent. It basically tells the GS_m that the message ends here hence executing the send command

Loading...

Reply

Al Diman

February 1, 2021 at 5:42 PM

I have loaded the code to my arduino uno. Everything seems to work except that my thingspeak channel is not updating temperature and humidity data. Pease guide me on the issue.

Loading...

Reply

Mr. Alam

February 1, 2021 at 5:44 PM

Add some delays between the thingspeak data sending command in the code. It should fix the issue

Loading...

Reply

**Al Diman**

February 13, 2021 at 1:42 PM

Hi,

Thanks on your reply. The data on my channel are now updating. The data are updated every 1.5 minutes (about 88 seconds). Is it possible to have data updated every 60 seconds? What part in the code should I change? Thanks again for the help.

Loading...

Reply

**Andrew**

March 25, 2021 at 1:30 AM

Hi, I noticed that the GSM module needs an external power supply rather than being powered from the arduino board. What sort of current does the GSM module draw? I was hoping to power this from a battery and place the node in a remote area with small solar panel to trickle feed the battery.

Loading...

Reply

**Mr. Alam**

March 25, 2021 at 10:26 AM

The power supply varies depending upon the type of GSM Module used. The SIM800/900 GSM Module requires more than 1.5A of current. Hence 9V, 2A supply is good for the module.

Loading...

Reply

**Andrew**

March 26, 2021 at 7:21 PM

Thanks for the reply. I have got it up and running now but it seems to be updating things speak at random intervals with some gaps as big as 25 minutes. Is this because the delays are not long enough and it times out before making connection?

Loading...

Reply

**Andy**

March 29, 2021 at 7:29 PM

Hi, I have the same issue. Can you please confirm exactly where this extra delay should be?

Loading...

Reply

**Mr. Alam**

March 29, 2021 at 8:42 PM

delay between each and every AT Commands should be increased.

Loading...

Reply

**Andrew**

March 29, 2021 at 11:34 PM

Hi, I have uploaded the same code and thingspeak updates but at random intervals. What delays should I adjust to make sure I get data sent everytime the program runs. I have 'send Ok' back but thingspeak does not update and it tends to randomly update now and then.

Loading...

Reply

**srikanth**

April 1, 2021 at 11:26 AM

#026 command is used for termination, so it varies from terminal program to program. the following terminal it will work. Take a try
<https://sites.google.com/site/terminalbpp/>

Loading...

Reply

**Tapiwanashe Vincent Kufarimani**

April 4, 2021 at 5:21 PM

i am having +CME ERROR:50 , +CME ERROR:53 and +CME ERROR:58

Loading...

Reply

**stevejferry**

June 28, 2021 at 7:33 PM

Hello, is there a way to get the http status from ThingSpeak ie the 200 code ThingSpeak sends back when the post has been successful? I am using an A9 GSM/GPRS module with an ESP32 board to receive LoRa packets from my nodes and forward them to ThingSpeak. Thanks, Steve.

Loading...

Reply

**samarjeetchavan**

October 28, 2021 at 2:47 PM

can we post data to a google sheet with similar code?

Loading...

Reply

**Srikanth n**

June 30, 2022 at 1:35 PM

I am also facing the same issue. did you find a solution ??

Loading...

Reply

**Abuud**

October 18, 2022 at 3:23 PM

Thanks for the well-explained tutorial. I was wondering if the same project can be implemented using esp32. what will be the necessary changes in the code in term of communication between esp32 and sim800L.

Loading...

Reply