

使用 **ESP8266 WiFi** 模块制作简单的物联网项目

Phodal Huang

September 8, 2017

目录

步骤 1: 设置 ESP8266	3
步骤 2: Arduino 连接 LM35 温度传感器	5
步骤 3: 总结	8

玩点什么: <https://www.wandianshenme.com>

原文链接:<https://www.wandianshenme.com/play/esp8266-arduino-create-simple-iot-project>

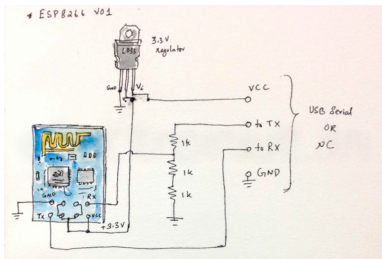
ESP8266 是一个成本不到 5 美元的 WiFi 模块。它能使得我们可以将传感器放在网上。(当然 75 美元的 Arduino Yun + 你的每个传感器 - 这也不是不可行的)。目前互联网上有很多关于这个传感器的点子, 让人很举。人们做了惊人的工作来破译, 这个来自中国的设备, 仅仅得到一个模糊的命令结构。似乎有三种使用这个模块的方式:

1. 使用 AT 命令, 在计算机上通过 USB 发送到串行适配器。这对测试和设置非常有用。
2. 与 Arduino 或任何其他微控制器对接, 并使用该板作为外围设备。
3. 直接对模块进行编程, 并使用其 GPIO 引脚与传感器进行通信, 因此不需使用第二个控制器。

我已经探索了上面的第 1 点和第 2 点, 这是我们要在这里谈论的内容。

步骤 1: 设置 ESP8266

为了用 ESP8266, 要做的第一件事 (和任何外星人一样) 是建立沟通。为此, 您将一个 USB 转 TTL 适配器连接到该模块, 并使用串行端口终端应用程序 (如 CoolTerm) 与之通信。连接此模块时要注意的一件事是: 该模块工作在 **3.3 V** 电压上 - 即使串行线路也不应超过此电压。以下是我连接 ESP8266 的方式:



在上面的电路中, 您可以看到我使用 **3.3V** 稳压器为电路供电, 并在 RX 线路上使用电阻分压器来保持电压。下面的会话使用 CoolTerm。我的主板工作波特率为 9600, 因为其固件已经升级到 0.9.2.2。您可能需要尝试其他波特率 - 例如 115200。

- 1 `<.p...p. (.....û.ÿ`
- 2 `[System Ready, Vendor:www.ai-thinker.com]`

现在, 您可以发送 AT 命令。您可以在此处看到完整的命令参考 ([ESP8266](#))。首先让我们发送“AT”:

- 1 AT
- 2

```
3
4 OK
```

然后, 让我们获取固件版本:

```
1 AT+GMR
2
3 0018000902
4
5 OK
```

现在, 我们来看看操作模式 (operation mode):

```
1 AT+CWMODE?
2
3 +CWMODE:3
4
5 OK
```

上面的 3 意味着我们处于单机 (Standalone) + 接入点 (Access Point) 模式。没关系。现在, 让我们做一些有趣的事情。我们列出所有可用的 WiFi 接入点:

```
1 AT+CWLAP
2
3 +CWLAP: (0, "Cisco15010-guest", -84, "ce:d7:19:0b:80:19", 1)
4 +CWLAP: (4, "TRENDnet", -86, "c8:d7:19:0b:80:19", 1)
5 +CWLAP: (2, "ASingh", -87, "00:25:5e:c5:d1:88", 1)
6 +CWLAP: (4, "HMMA", -66, "80:ea:96:f1:27:7a", 11)
7 +CWLAP: (4, "dlink-30B8", -91, "c8:d3:a3:6e:30:b8", 6)
8 +CWLAP: (4, "Deepak Mullick", -93, "80:a1:d7:77:a1:e4", 6)
9 +CWLAP: (1, "MGmnt", -91, "80:a1:d7:77:a1:e5", 6)
10 +CWLAP: (3, "rajat_linksys", -67, "20:aa:4b:31:43:9d", 6)
11 +CWLAP: (2, "smk385", -89, "00:1f:33:bc:02:f6", 6)
12 +CWLAP: (4, "Apoorv", -91, "ec:1a:59:17:53:93", 7)
13 +CWLAP: (3, "AkhIshir", -88, "08:60:6e:cb:63:b0", 11)
14 +CWLAP: (2, "sunil", -77, "94:d7:23:f5:fc:34", 9)
15 +CWLAP: (1, "MGmnt", -77, "94:d7:23:f5:fc:35", 9)
16 +CWLAP: (1, "Tanu", -92, "00:1d:7e:1e:68:4e", 11)
17
```

18 OK

然后，让我们连接上我的 WiFi:

```
1 AT+CWJAP="HMMA", "NOTTELLINGMYPASSWD"
2
3
4 OK
```

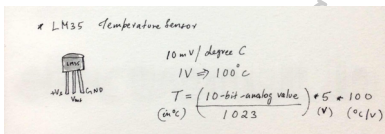
我们检查一下，我们是否真的获得了一个 IP 地址:

```
1 AT+CIFSR
2
3 192.168.4.1
4 192.168.4.10
5
6 OK
```

Whoohoo! 我们已经连接上网络了。实际上，我们会在下一节中使用互联网。此外，上述的设置已经存储在模块中。即使您打开和关闭电源，它将自动连接到此网络。

步骤 2: Arduino 连接 LM35 温度传感器

现在，我们将模块放在网络上，我们通过创建一个小的物联网（物联网）设备来利用它。我在 [ThingSpeak](#) 上写了一篇关于绘制传感器数据的文章。这一次，我会用一个 LM35 温度传感器。我将使用 Arduino Mini Pro 克隆版与 ESP8266 通信。这是 LM35 的样子:



在如下的代码中，我使用 `SoftwareSerial` 库与 ESP8266 通信。我使用硬件 `Serial` 进行调试。（您可以尝试相反的方法。）我也假设 ESP8266 已经设置为连接到 WiFi 网络:

```
1 // esp8266_test.ino
2 //
3 // Plot LM35 data on thingspeak.com using an Arduino and an ESP8266 WiFi
4 // module.
5 //
6 // Author: Mahesh Venkitachalam
```

```
7 // Website: electronut.in
8 #include <SoftwareSerial.h>#include <stdlib.h>
9
10 // LED
11 int ledPin = 13;
12 // LM35 analog input
13 int lm35Pin = 0;
14
15 // replace with your channel's thingspeak API key
16 String apiKey = "T2RJXWQAVXG4ZV39";
17
18 // connect 10 to TX of Serial USB
19 // connect 11 to RX of serial USB
20 SoftwareSerial ser(10, 11); // RX, TX
21
22 // this runs once
23 void setup() {
24   // initialize the digital pin as an output.
25   pinMode(ledPin, OUTPUT);
26
27   // enable debug serial
28   Serial.begin(9600);
29   // enable software serial
30   ser.begin(9600);
31
32   // reset ESP8266
33   ser.println("AT+RST");
34 }
35
36
37 // the loop
38 void loop() {
39
40   // blink LED on board
41   digitalWrite(ledPin, HIGH);
42   delay(200);
```

```
43  digitalWrite(ledPin, LOW);
44  // read the value from LM35. // read 10 values for averaging.
45  int val = 0;
46  for(int i = 0; i < 10; i++) {
47      val += analogRead(lm35Pin);
48      delay(500);
49  }
50
51  // convert to temp:
52  // temp value is in 0-1023 range
53  // LM35 outputs 10mV/degree C. ie, 1 Volt => 100 degrees C
54  // So Temp = (avg_val/1023)*5 Volts * 100 degrees/Volt
55  float temp = val*50.0f/1023.0f;
56
57  // convert to string
58  char buf[16];
59  String strTemp = dtostrf(temp, 4, 1, buf);
60
61  Serial.println(strTemp);
62
63  // TCP connection
64  String cmd = "AT+CIPSTART=\"TCP\", \"";
65  cmd += "184.106.153.149"; // api.thingspeak.com
66  cmd += "\",80";
67  ser.println(cmd);
68
69  if(ser.find("Error")){
70      Serial.println("AT+CIPSTART error");
71      return;
72  }
73
74  // prepare GET string
75  String getStr = "GET /update?api_key=";
76  getStr += apiKey;
77  getStr += "&field1=";
78  getStr += String(strTemp);
```

```
79  getStr += "\r\n\r\n";
80  // send data length  cmd = "AT+CIPSEND=";
81  cmd += String(getStr.length());
82  ser.println(cmd);
83
84  if(ser.find(">")){
85    ser.print(getStr);
86  }
87  else{
88    ser.println("AT+CIPCLOSE");
89    // alert user
90    Serial.println("AT+CIPCLOSE");
91  }
92
93  // thingspeak needs 15 sec delay between updates
94  delay(16000);
95 }
```

在上面的代码中，我读取模拟引脚来计算温度，并通过 **GET** 请求将该信息发送给 **ThingSpeak** 的频道（channel）。随后，你就可以在 **ThingSpeak** 上看到温度曲线。

步骤 3: 总结

我认为这个模块是非常有价值的，也是将传感器放在互联网上的非常实用的方式。尽管这个模块可以直接编程，但是由于可以轻松地使用它，所以我可能会将其用作外设。

原文链接：<https://www.wandianshenme.com/play/esp8266-arduino-create-simple-iot-project>