

使用 **ESP8266** 和 **RN2483** 构建 **TTN Lora** 节点

Phodal Huang

September 8, 2017

目录

步骤 1: 连接零件	3
步骤 2: 准备 TTN 应用程序	4
步骤 3: 代码	4
步骤 +: 延伸和包装	6

玩点什么: <https://www.wandianshenme.com>

原文链接:<https://www.wandianshenme.com/play/esp-8266-rn2483-build-ttn-lora-node>

TheThingsNetwork 是一个全球性的社区, 其使命是将物联网世界更贴近人们, 使连接变得更加容易、完全免费。它通过提供开放式通信基站和分布式后端系统来处理所有的设备和目标系统的通信, 安全和数据可用性。

其使用的通信协议是 LoRaWAN。它允许低功率设备从远距离连接, 考虑到它们只能每秒发送或接收短脉冲数据。通信工作在开放频率, 这意味着它们不依赖监管机构或主要电信提供商。

最近我在斯科普 (Skopje) 里申请了一个当地的 TTN 社区, 得到了批准, 基础设施仍在发展中。对于第一个测试性项目, 我使用 Arduino Uno 和 Microchip RN2483 结合使用 LoRaWAN 连接。这是最基本的例子, 它工作得很好。然而, 真正的要求是有一个 ESP8266 主控制器与 RN2483 通话。这样做的原因是, 我正在进行的工作需要最终用户配置机制, ESP8266 集成了 WiFi 和 WebServer 就是一个物联网。

为了方便起见, 我使用了基于 ESP-12E 芯片的 NodeMCU 开发板。它类似于 Arduino, 具有直接的微型 USB 连接, 无需 FTDI, 并具有连接 RN2483 芯片及其他传感器、执行器所需的所有引脚。对于开发, 您可以使用 Arduino IDE 和 SDK 与 ESP8266 板卡和端口。我在这之前写过这个话题 (可能有点过时了...)

步骤 1: 连接零件

对于一个非常基础的搭建, 您将需要:

- 一个原型 PCB (或者一个原型板)
- NodeMCU 1.0 开发板
- RN2483 芯片
- 一些补丁用的线
- 坚实的核心线作为天线
- 焊丝

NodeMCU 具有比 RN2483 芯片更大的外形尺寸, 所以甚至可以通过使用一些插头将它们紧紧地焊在一起, 或者在另一个之间焊接。

两者之间的联线如下:

RN2483 TX (6) <-> NodeMCU D6 RN2483 RX (7) <-> NodeMCU D5
RN2483 VDD (12 或者 34) <-> NodeMCU 3.3V
RN2483 GND (20 或者 33 ...) <-> NodeMCU GND
RN2483 RESET (32) <-> NodeMCU D7
RN2483 RFH (23) <-> Antenna 或者 8.6cm 实心芯线

步骤 2: 准备 TTN 应用程序

如果你没有 **TheThingsNetwork** 的帐户, 你需要转到 **TTN** 的网站并创建一个帐户。并检查您所在的区域是否有 **LoRa TTN** 覆盖。

之后, 转到分段应用程序并创建一个新的应用程序。选择使用默认应用程序键激活设备, 以便能够使用 **OTAA** (这不是强制性的, 我只是在示例代码中使用 **OTAA**)。创建应用程序时, 请使用 **App EUI**, 然后在设置中获取默认应用程序键。之后你将需要他们的代码。离开主应用程序页面打开, 以便您可以看到之后发送的连接和数据。

步骤 3: 代码

编码之前, 你需要先做一些准备:

- 获取最新的 **Arduino IDE**
- 安装 **NodeMCU devkit** 驱动程序
- 将 **Esp8266** 板安装在板卡管理器中 (或更新到最新版本)
- 安装 **RN2483** 库
- 安装 **ESPSoftwareSerial** 库

我使用的代码是由 **jpmeiers** 完成的教程 ([How to build your first TTN node: Arduino + RN2483 - Hardware](#)) 修改的:

```
1 #include <rn2483.h>
2 #include <SoftwareSerial.h>
3
4 #define RST D7
5 SoftwareSerial mySerial(D6, D5); // RX, TX
6
7 //create an instance of the rn2483 library,
8 //giving the software UART as stream to use,
9 //and using LoRa WAN
10 rn2483 myLora(mySerial);
11
12 // the setup routine runs once when you press reset:
13 void setup() {
14
15 // Open serial communications and wait for port to open:
```

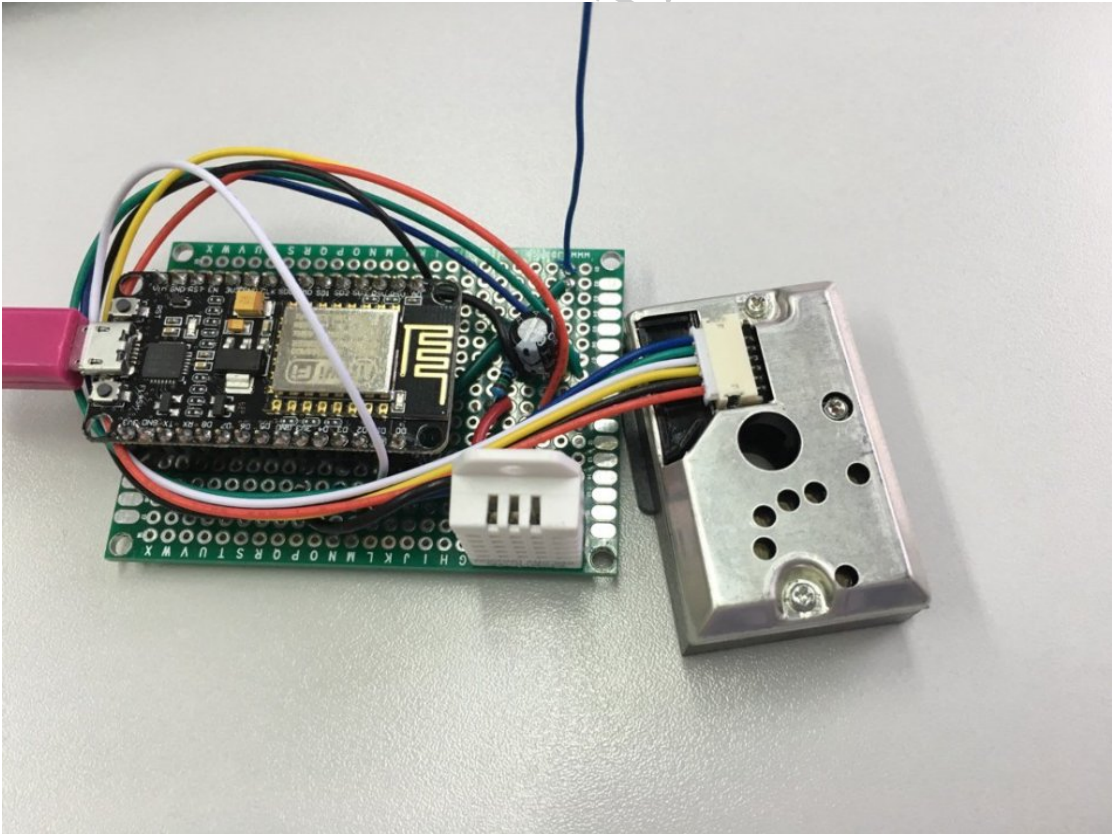
```
16 Serial.begin(57600);
17 mySerial.begin(9600);
18 Serial.println("Startup");
19
20 //reset rn2483
21 pinMode(RST, OUTPUT);
22 digitalWrite(RST, LOW);
23 delay(500);
24 digitalWrite(RST, HIGH);
25
26 //initialise the rn2483 module
27 myLora.autobaud();
28
29 //print out the HWEUI so that we can register it via ttncctl
30 Serial.println("When using OTAA, register this DevEUI: ");
31 Serial.println(myLora.hweui());
32 Serial.println("RN2483 version number:");
33 Serial.println(myLora.sysver());
34
35 //OTAA: init(String AppEUI, String AppKey);
36 myLora.init("<Insert you app EUI key here>", "<insert your default app
    key here>");
37
38 delay(2000);
39 }
40
41 // the loop routine runs over and over again forever:
42 void loop() {
43
44     //Do your stuff here
45     //Measure things if you're working with a sensor
46     //and pack them into an output string
47     String toSend = "DataHere";
48
49     Serial.println("TXing: " + toSend);
50     myLora.txUncnf(toSend ); //send the data
```

```
51  
52 Serial.println("TXed.");    delay(60000); //Send every minute (maybe you  
    need to tune this)  
53 }
```

这是为了开始工作所需要的最低限度的代码。我通过修改原始代码，然后删除其他部分，所以如果哪里有问题，请及时告诉我。

当您全部打包时，选择 **NodeMCU** 作为您当前的主板，点亮串行显示器，将其设置为 **57600** 波特率，并将代码烧录到设备上。如果一切正常，您应该在串行日志中看到好消息，并通过刷新 **ttn** 上的应用页面，您应该看到您的设备已注册，并且消息都会在每分钟到达。

步骤 +: 延伸和包装



我所做的最终产品是一个灰尘、温湿度 **TTN** 传感器节点。您可以通过结合 **DHT22** 和尖锐的灰尘传感器来实现同样的功能。然而，代码仍然看起来有点不舒服，并且灰尘传感器可能需要一些额外的工作和/或校准，所以我不敢发布它。

最后，我在 **OpenSCAD** 中设计了一个定制机箱（由 **b2vn** 组成的这个设计）和 **3D** 打印。我只是做得越来越长，为 **DHT** 和灰尘传感器开口放孔，让空气流通。

原文链接:<http://pance.mk/index.php/thethingsnetwork-lora-node-with-esp8266-and-rn2483/>

原文链接:<https://www.wandianshenme.com/play/esp-8266-rn2483-build-ttn-lora-node>

玩点什么: <https://www.wandianshenme.com>