

**Cloud.Node User Manual**

**By SeeedStudio**

VERSION: 0.9

2013.May.31

目录

[1. 概述 3](#_Toc358180958)

[2. 硬件 4](#_Toc358180959)

[3. 注册Xively账号 4](#_Toc358180960)

[4. 配置Cloud.Node 5](#_Toc358180961)

[4.1 传感器设置 5](#_Toc358180962)

[4.2 Wi-Fi 设置 6](#_Toc358180963)

[5. 配置使用Atom 7](#_Toc358180964)

[6. 一个使用例子 8](#_Toc358180965)

[6.1 Step 0: 注册Xively 9](#_Toc358180966)

[6.2 Step 1: 配置Cloud.Node 12](#_Toc358180967)

[6.3 Step 2：配置Atom.Node 15](#_Toc358180968)

[6.4 Step 3：查看设备列表 16](#_Toc358180969)

[7. 参考链接 18](#_Toc358180970)

# 概述

Node是基于Arduino的物联网节点平台。该平台配合SeeedStudio的Grove电子模块系列和Xbee等通讯模块，用户可以快速地组建智能家居网络，并实现一定逻辑控制。使用Node平台来实现一个智能家居系统，例如远程人体监测报警、远程测距等，将变得非常方便和容易。

Node分为Atom.Node和Cloud.Node。

Atom主要用于局域网的通信。Atom通过Xbee或RFbee等无线模块进行相互通信，通信可靠，通讯距离高达100m；在配置上，用户使用SeeedStudio公司开发的基于Android平台的App，使用可见光通信的方式对Atom进行配置，配置过程简单直观；在使用上，Atom可以被单独使用来实现简单的IFTTT功能，也可以配对使用以实现远程无线控制和监控功能。

Cloud配备Xbee接口以及Wi-Fi功能，所以Cloud除了可以与Atom通信之外，也可以与互联网进行通信。配合Cloud，用户可以很方便的将Atom采集到的信号数据等上传至云端，或从云端对Atom进行简单控制。这将是一个很方便易用的智能物联网解决方案。

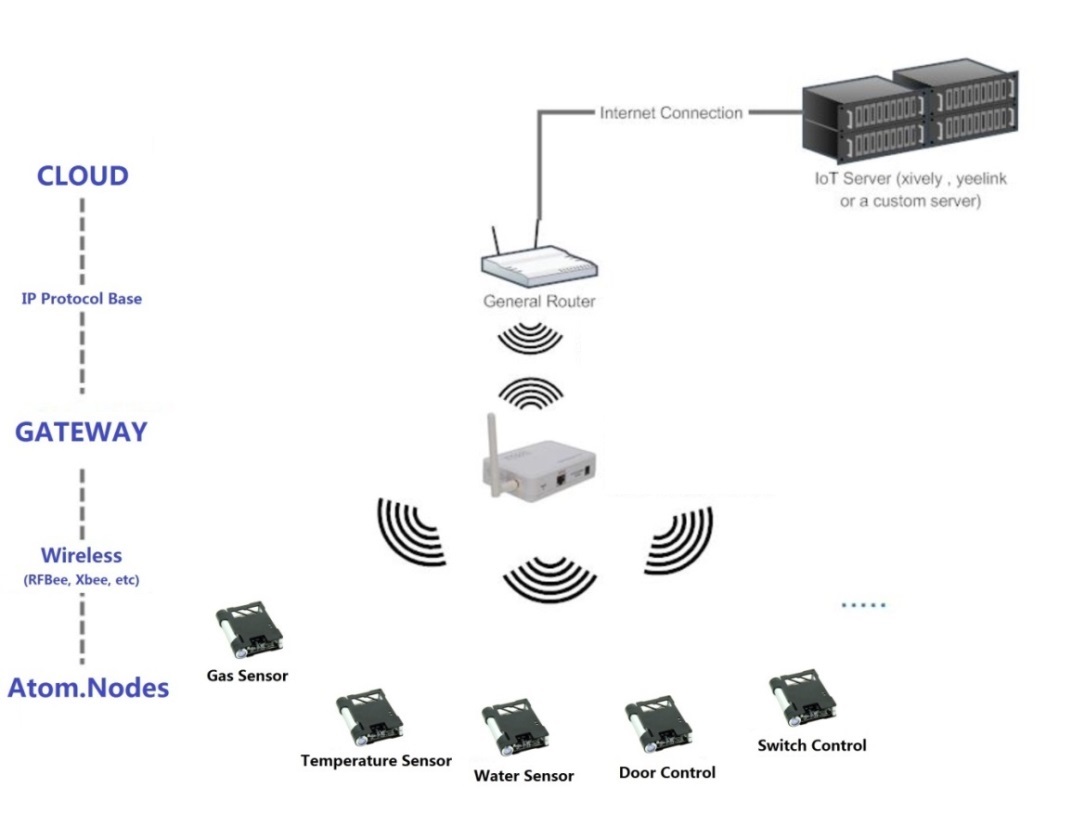


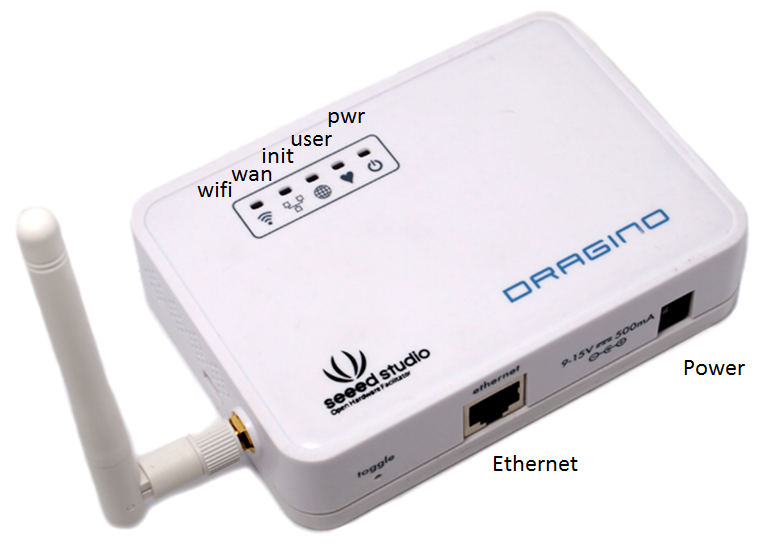
图 1‑1 Node工作示意图

通过阅读本手册，你可以：

* 了解Cloud.Node及其工作流程
* 使用Cloud.Node与Xively以远程查看Node网络的传感器数值

下面将会逐步的说明使用的方法，最后有一个Example，帮你更加深刻的理解如何让Cloud.Node运转起来，**我们建议你先把前五章看完，再一步一步的跟着第六章的Example进行操作**。

# 硬件



* Ethernet：以太网口，用于和电脑连接
* Power：电源输入
* Wi-Fi led: Wi-Fi工作灯，闪的时候表示Wi-Fi在工作
* Wan Led：灯亮表示Ethernet连接到电脑
* Init Led：初始化灯，灯亮表示正在初始化，灯灭表示初始化完成
* User：用户灯，目前没有定义
* Pwr Led：电源灯，上电即亮

# 注册Xively账号

Xively是一个通用物联网平台，主要提供传感器数据的接入，存储和展现服务，为所有的开源软硬件爱好者、制造型企业，提供一个物联网项目的平台，使得硬件和制造业者能够在不关心服务器实现细节和运维的情况下，拥有交付物联网化的电子产品的能力。Xively是现阶段的Cloud.Node所用到的主要的云端数据存储平台。

# 配置Cloud.Node

电脑通过网线连接Dragino，上电，等待Dragino中间的提示灯（上面有个地球的）熄灭。

注：这个过程大概需要2分钟，请耐心等待

网页输入<192.168.255.1>，登录到Dragino配置页面。需要设置Sensor及Wi-Fi里面的内容即可让Cloud.Node运转起来。熟悉其他选项的内容可以为你带来方便，但不是必须的。

## 传感器设置

点击Sensor选项：

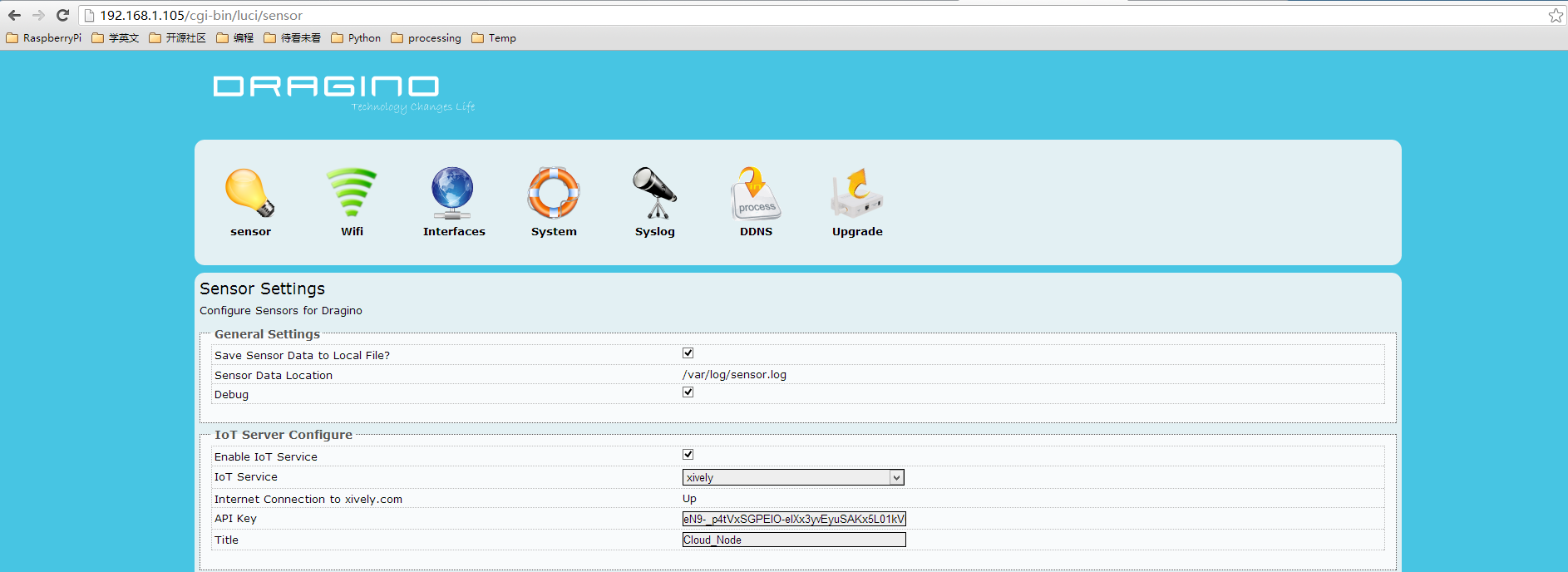


图 ‑1 配置传感器

**Main Settings**:

* **Save Sensor Data to Local File**:是否把传感器数值保存到本地文件
* **Sensor Data Location**: 如果勾选上面的选择，则传感器数据保存到/var/log/sensor.log

**IoT Server Configure:**

* **Enable IoT Service:** 是否开启云端服务
* **IoT Service:** 选择云端服务器，Xivey
* **API Key:** 第一章提到的APIKEY，注册帐号时生成
* **Title:** 上传到云端的设备名称前缀



图 2‑2 Devices List

**Devices List:**

列出了已经添加的Atom.Node的列表，从列表可以看到，该Cloud已经添加了4个设备，可以看到每个设备的Device Id, Sensor Name, Actuator Name已经Post地址。Post地址是Xively不需要对它进行操作。

点击右下角的Save按钮保存，之后进行Wi-Fi的配置。

注：Devices List里面的所有内容不需要用户进行编辑，当Atom.Node加入时，会自动添加

## Wi-Fi 设置

首先，假设Cloud.Node工作的环境有Wi-Fi网络存在，并且你知道Wi-Fi的密码。点击Wi-Fi选项，弹出以下界面：

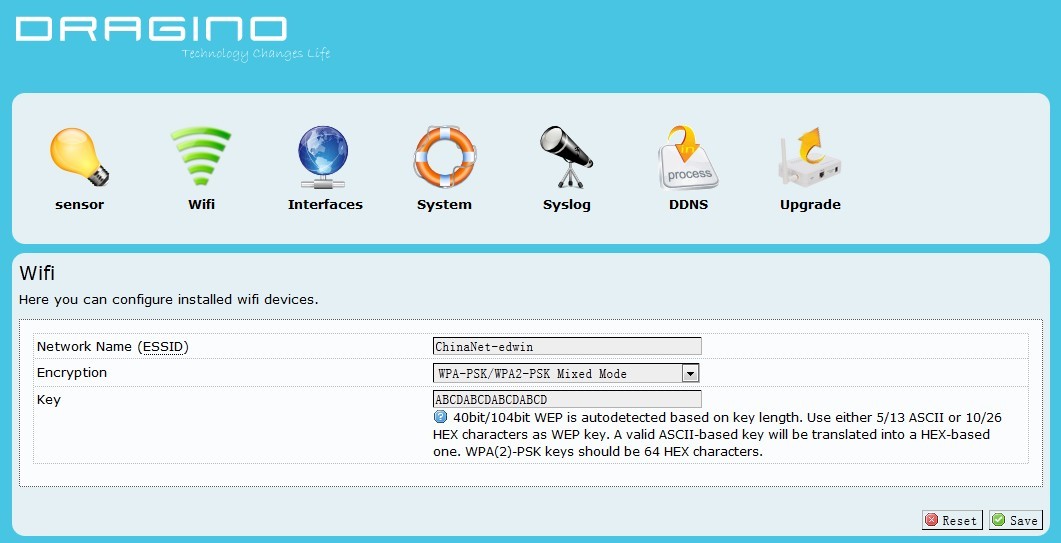


图 ‑3 Wi-Fi配置页面

**Wi-Fi Settings:**

* **Network Name(ESSID):** 输入Wi-Fi的SSID
* **Encryption:** 加密方式
* **Key:** 密码

注：点击右下角的保存后，等待页面刷新，这个时间可能要几十秒到1分钟,请耐心等待。

保存完成后，可以看到最外侧的Wi-Fi led闪烁，表示已经成功连上Wi-Fi。

至此，Cloud.Node的配置已经完成。

# 配置使用Atom

Cloud.Node已经配置完毕，可以让Atom出场了，如果你不熟悉Atom的使用，可以参考：<http://www.seeedstudio.com/wiki/Atom_Node_V1.0>

Atom工作起来后，Cloud会自动接收Atom的传感器数值，并且在网页配置页面以及Xively添加一个设备列表，当有数据广播时把数据推送到Xively。

# 一个使用例子

为了让你更容易地使用Cloud.Node，这里将会从头开始一步一步进行详细的讲解。

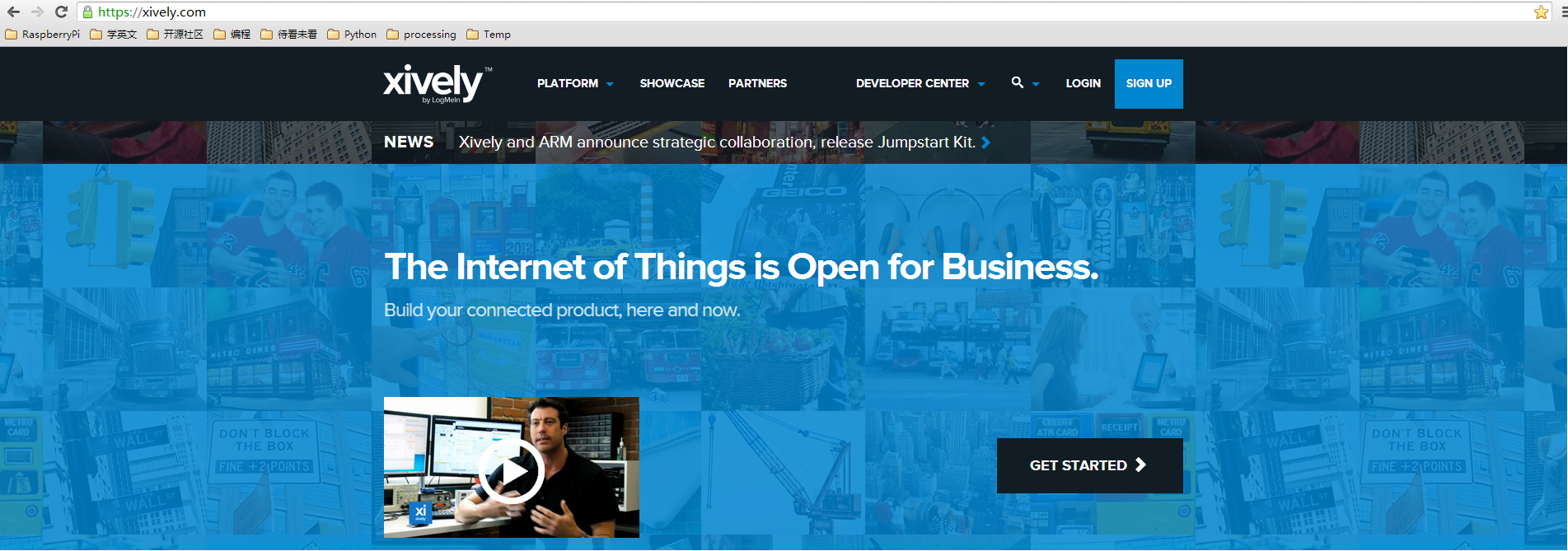
我想知道室外一天24小时的光线变化，这样我就可以大概了解什么时候天亮，什么时候开始天黑了。这是个很有趣的过程，你将会发现，大自然的变化虽然很缓慢，但是变化却是持续的；黑夜虽然是漫长的，但光明终将到来。

假设你已经会使用Atom.Node，并且你还有一块Grove-Light Sensor。你上周到SeeedStudio买了一块Cloud.Node，今天上午终于到了，你满怀希望地打开包装，深深的为包装的精美所吸引，实在太漂亮了！于是你迫不及待地想让它立马工作起来。那么，开始吧。

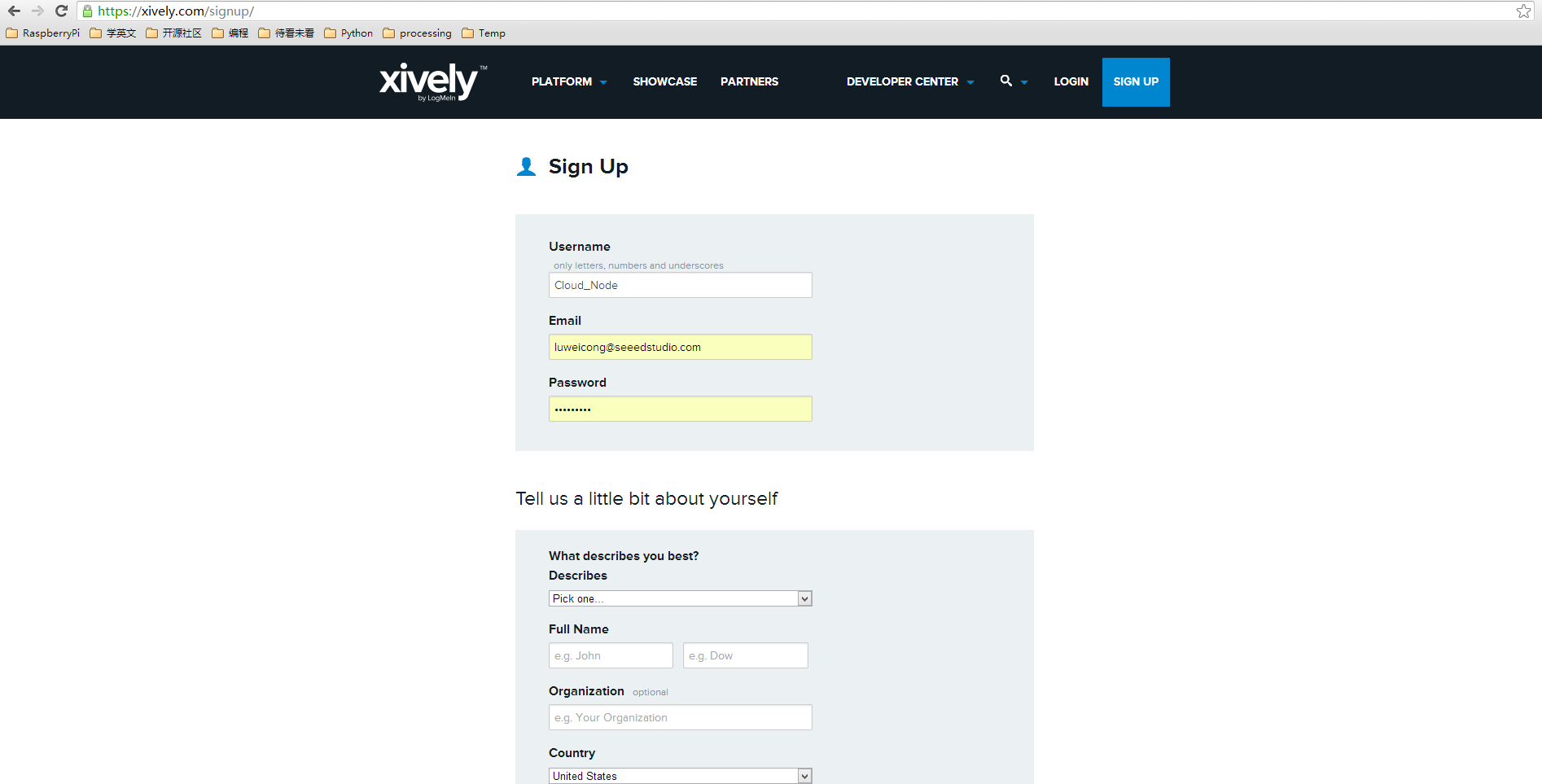


## Step 0: 注册Xively

打开网页浏览器，输入[www.xively.com](http://www.xively.com)



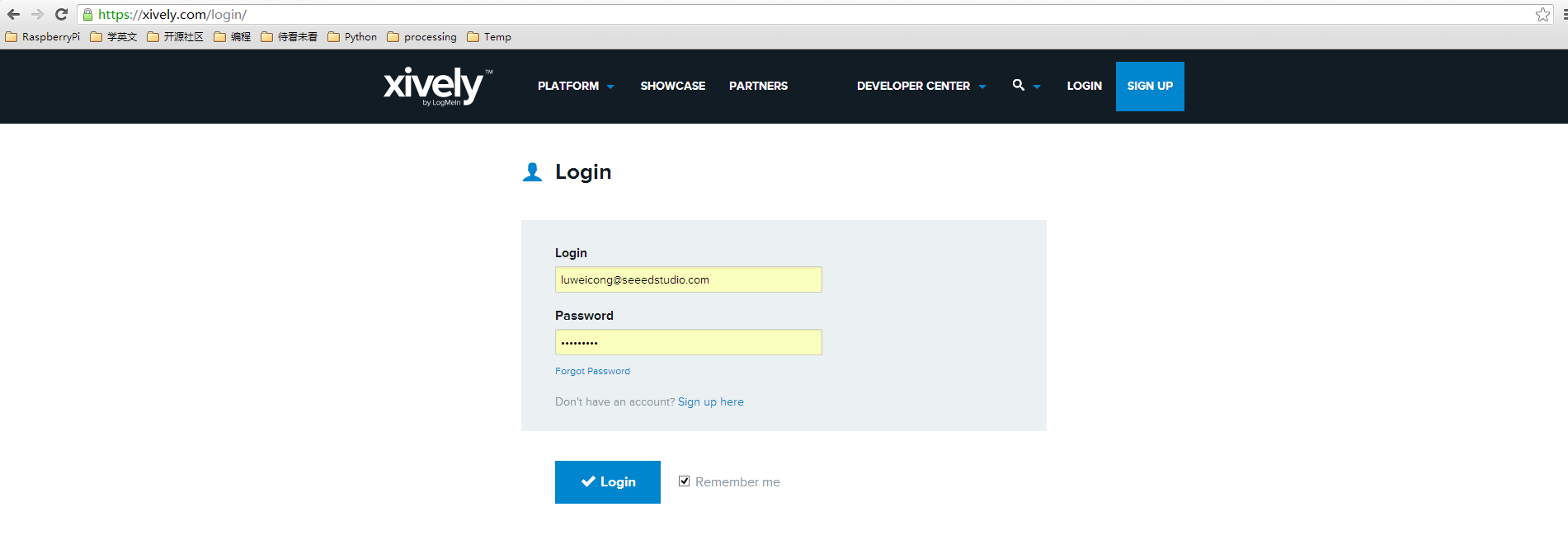
点击右上角的SIGN UP：



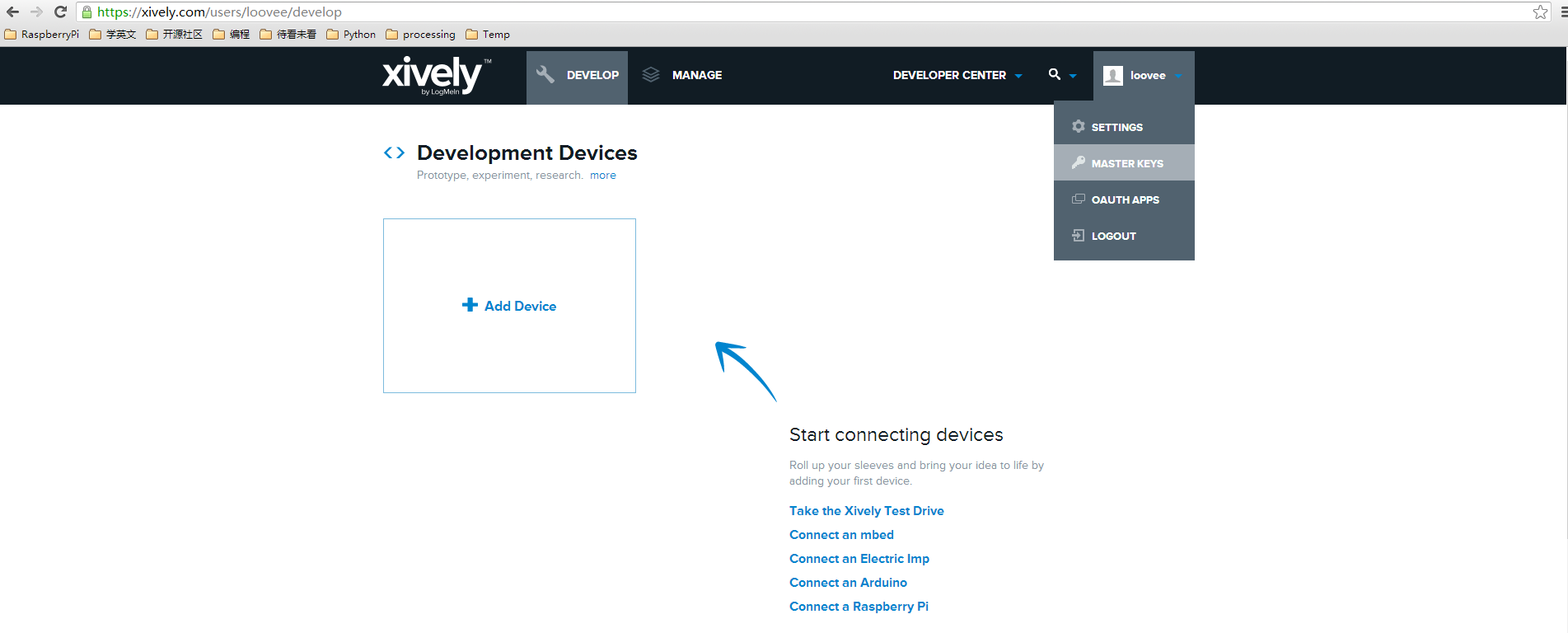
填入相关信息，完成注册。

注册完成后，进入你注册时所用的邮箱，收到一封来自Xively点击里面的链接，完成注册。

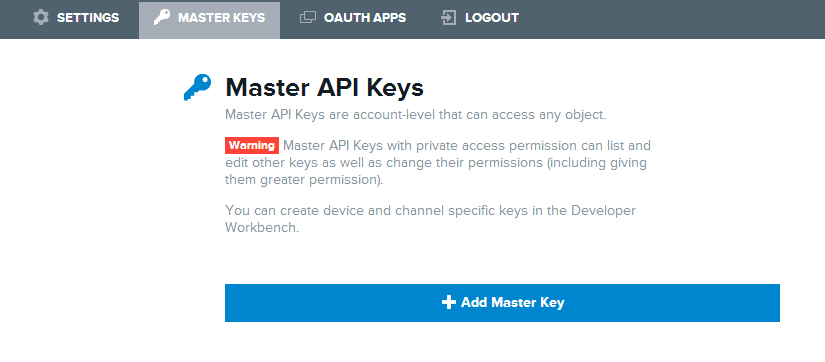
再次登陆[www.xively.com](http://www.xively.com)，点击右上角的登陆按钮，输入刚才注册的信息，并点Login，登录到Xively.



这样，你完成了Xively的注册。接下来，需要生成APIKey。把鼠标光标移到右上角的用户名，弹出下拉框，点击MASTER KEYS



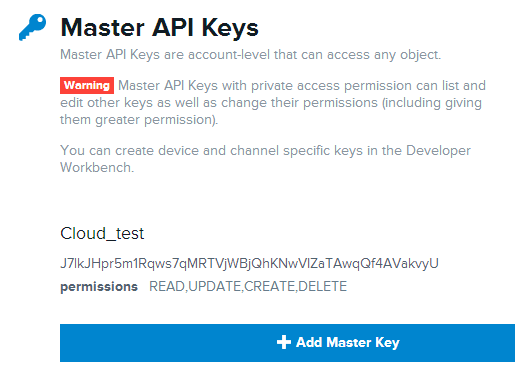
弹出：



点击



在上面随便填入一个名字，并且在Permissions下按照图中所示勾选，然后点击Add Key:



可以看到，已经成功添加了一个API Key，其实是一串奇奇怪怪的很长的字符，J7IKJ……等会要用到。

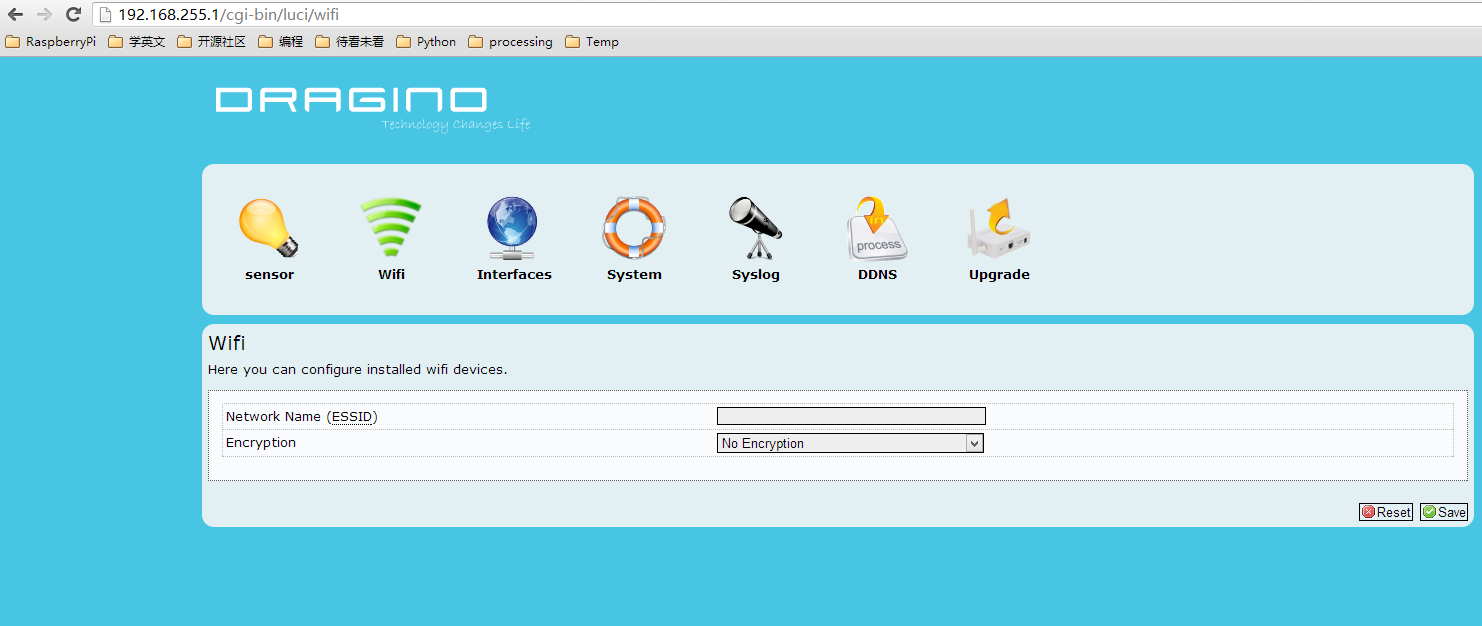
## Step 1: 配置Cloud.Node

用网线连接Cloud.Node与电脑：网线的一端插入电脑的网口，另一端插入Cloud的Ethernet口。

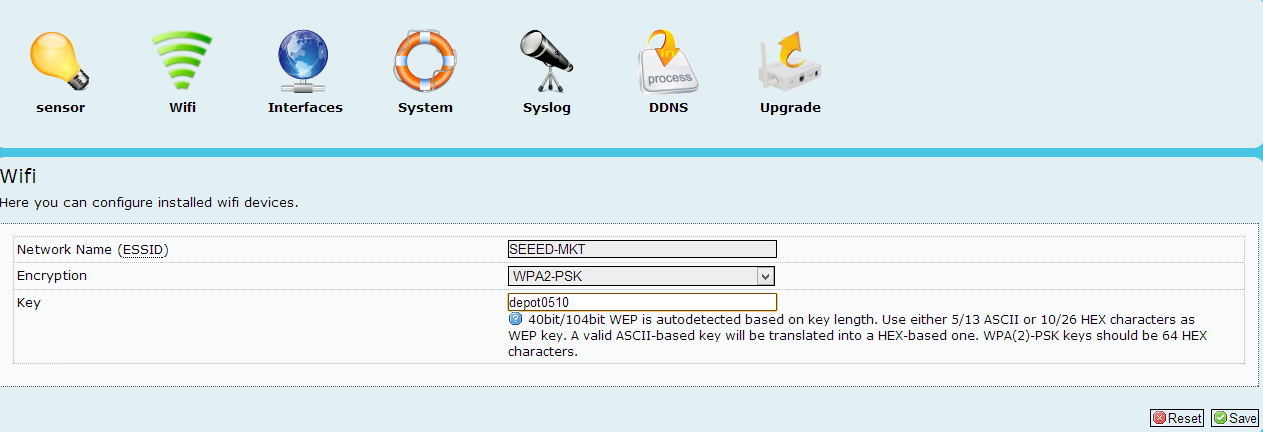
Cloud.Node上电，可以看到电源等亮起。上电大约5秒后init灯亮，表明Cloud正在启动。大约1分钟后，init灯熄灭，表示初始化完成，可以进行下一步的操作。

注:在等待init灯熄灭的过程中，其他灯闪烁的状态不予理会

Cloud.Node启动完成后，打开网页浏览器，输入192.168.255.1，回车，登陆到Cloud.Node的配置页面：

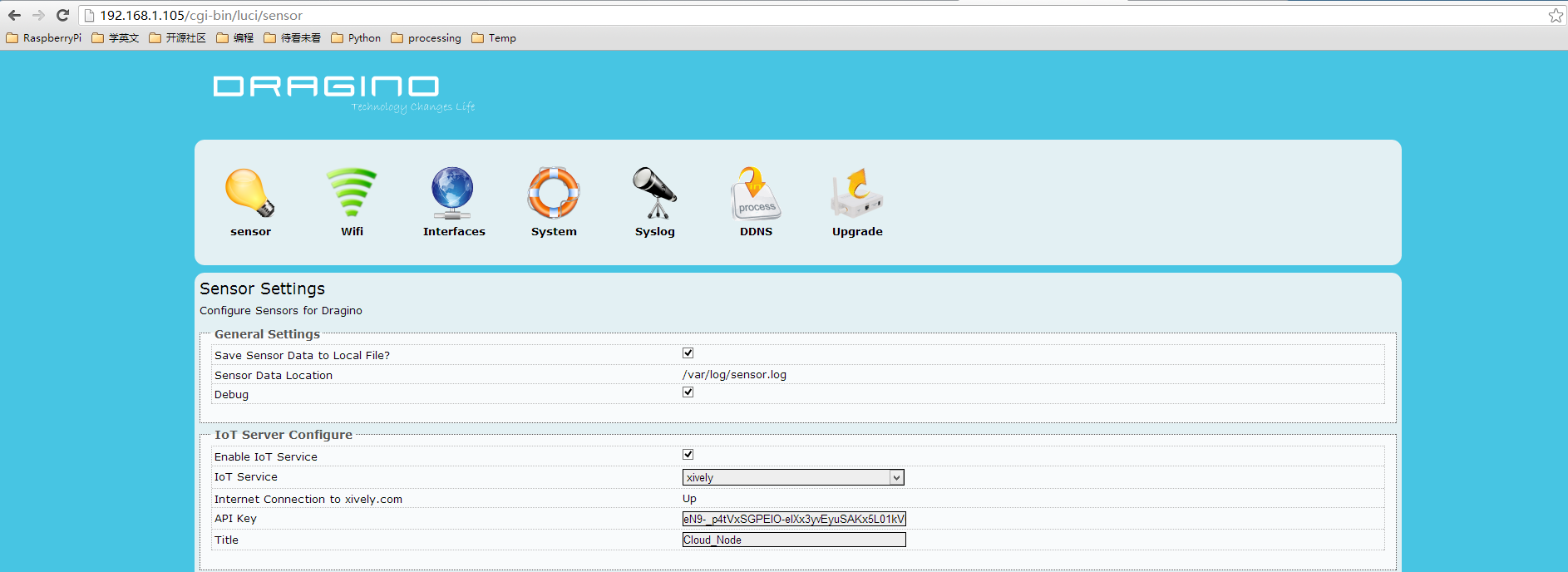


在Network Name(ESSID)输入Wi-Fi的SSID，Encryption选择加密方式，以及输入Key，选择Save。



等待网页刷新，刷新完毕尝试登录[www.google.com](http://www.google.com), 如果成功登录表示配置成功，否则检查信息是否正确填入。

之后点击Sensor按钮：

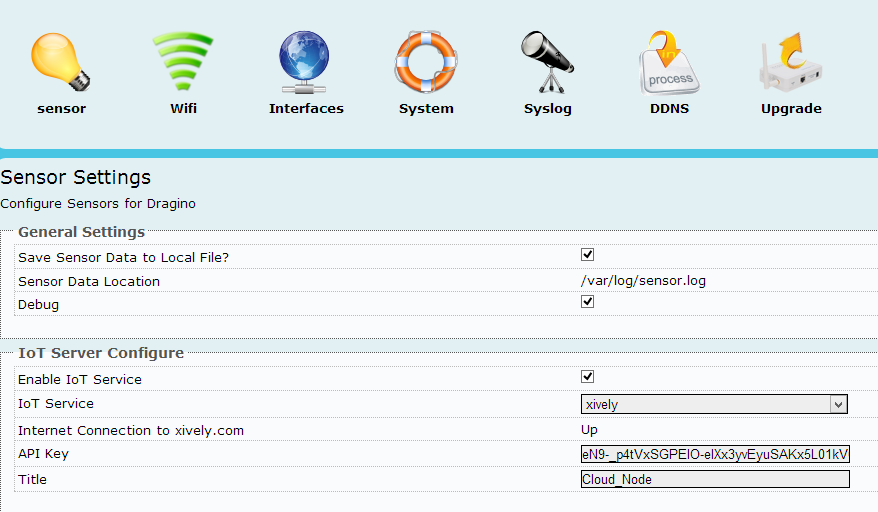


你需要填写的是API Key和Title。

把Step0末尾提到的API KEY复制到网页配置Sensor里面的API Key。

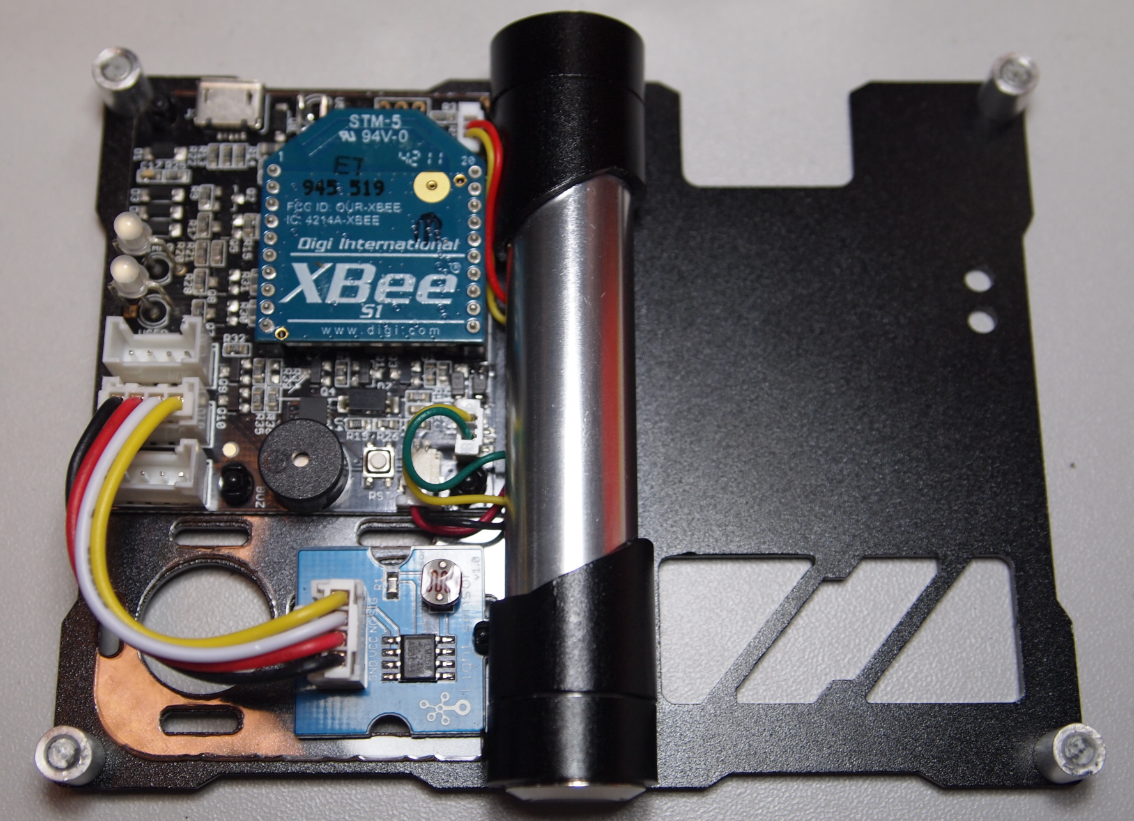
Title随便填入一个名字，这里填的是Cloud\_Node，稍后你就会知道这个名字的用途。

点击右下角的Save完成配置，等待页面刷新完成。

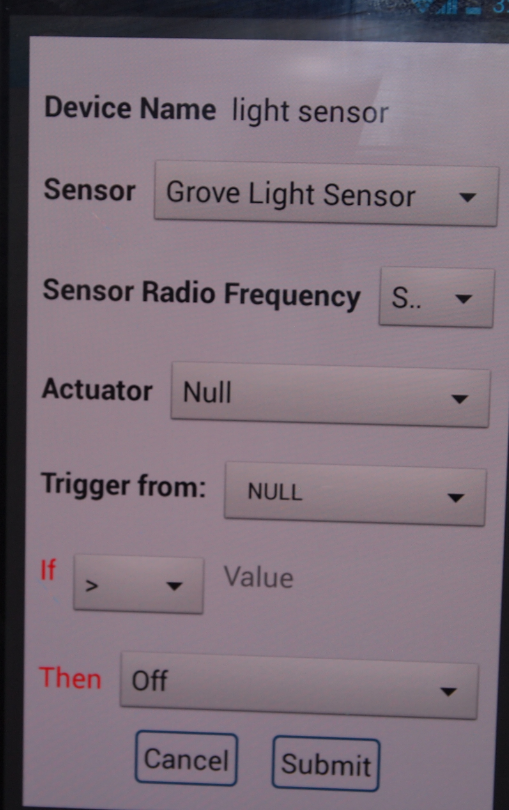


## Step 2：配置Atom.Node

把Grove-Light Sensor接入Atom.Node：



打开配置Atom的手机APP，对Atom.Node进行配置，每1s广播一次：

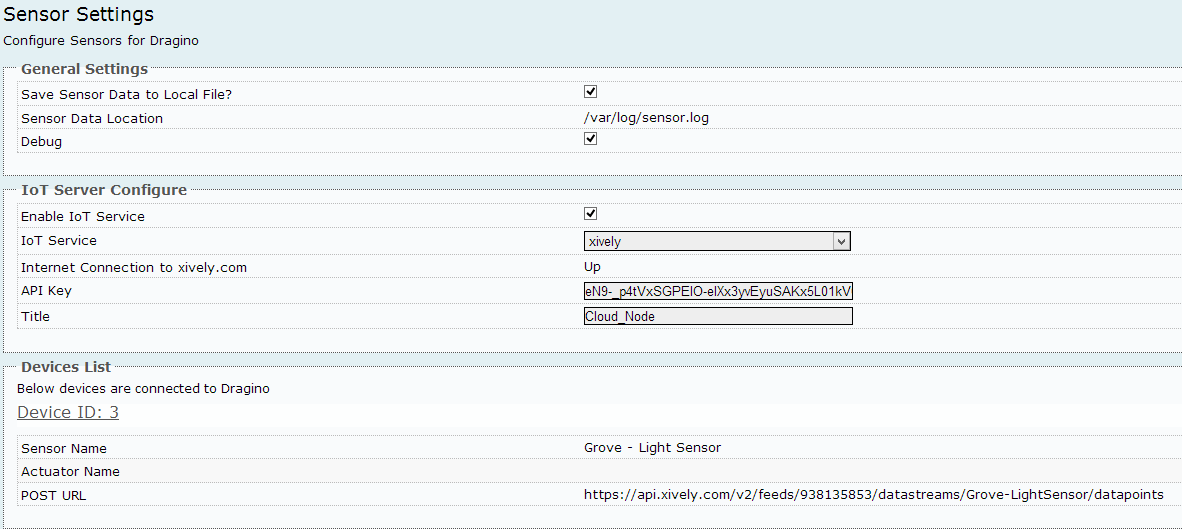


配置完毕，Atom.Node开始广播数据，Cloud.Node内置Xbee接收到广播数据后，自动创建了一个设备，并且接收到传感器值后，自动把数据推送到Xively

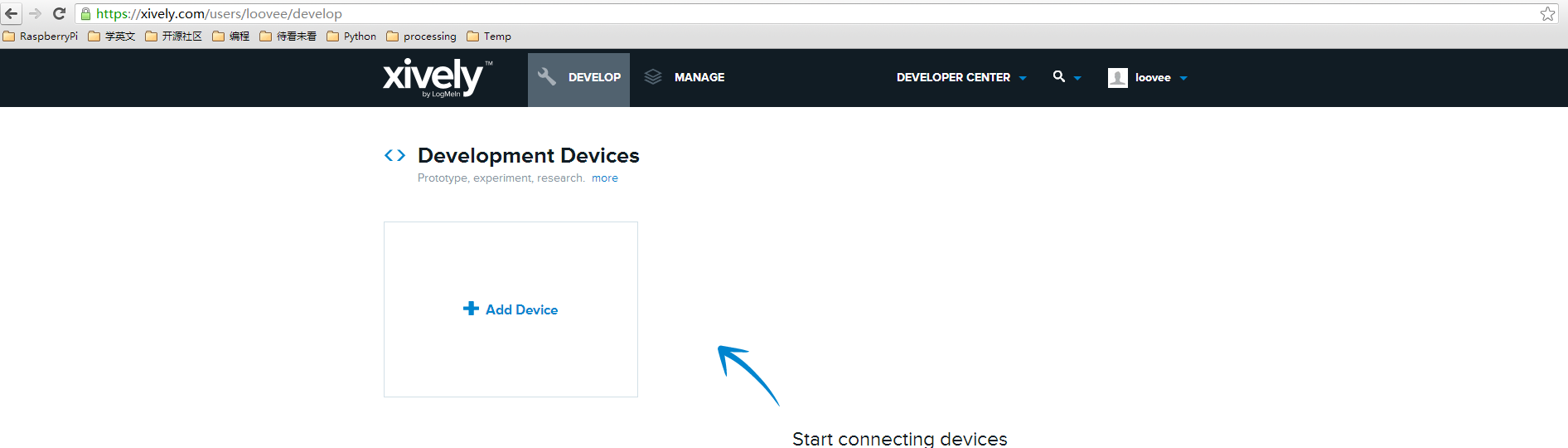
把配置好的Atom.Node放置到阳台或者窗台，让它能够感受到室外的光线，就可以不用管它了。

## Step 3：查看设备列表

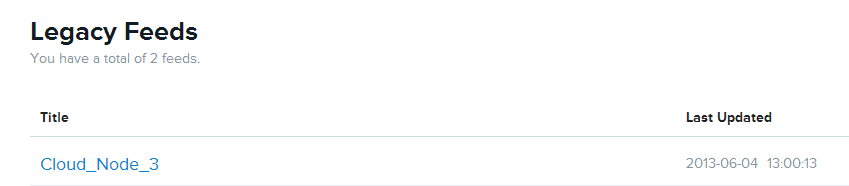
进入网页配置端，点sensor，你会看到，Cloud的Device List里面已经出现了Light Sensor的列表：



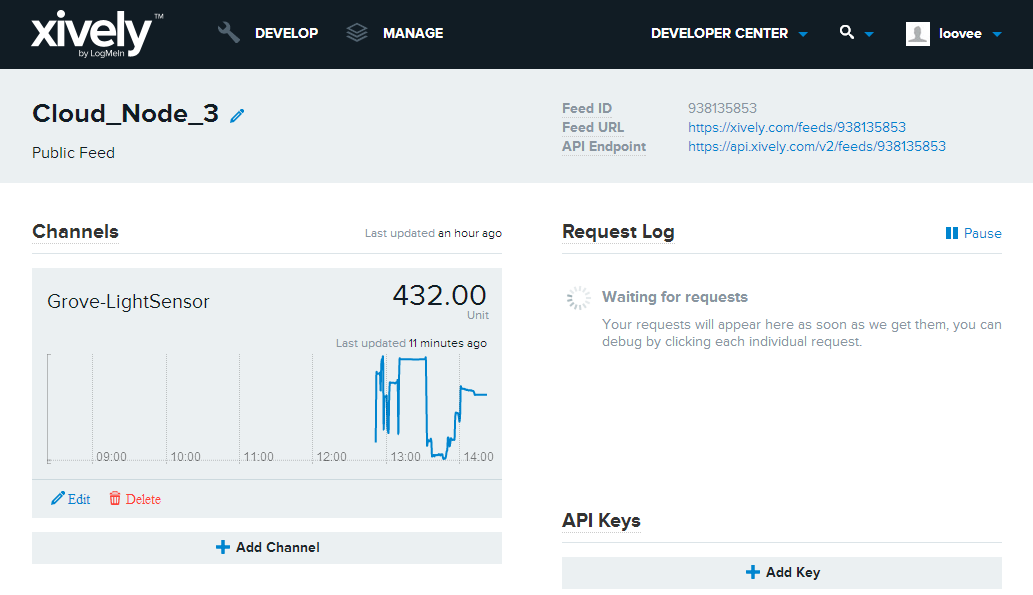
同时Xively上也添加了该设备，切换到Xively的界面，点击DEVELOP。



可以在网页的下方看到：



Cloud.Node已经在Xively上添加了这个设备了，如果想看里面的数据，点击Cloud\_Node\_3：



看到了我们刚才上传的数据。

这标志着你的Cloud.Node已经初步工作起来了。这里仅仅是简单介绍了Xively的使用，Xively本身功能是非常强大的，你可以自己尝试进行更加深入了解。

# 参考链接

* [www.seeedstudio.com](http://www.seeedstudio.com) : Dragrove vendor, more info about Dragrove and its development kit can be found here.
* [www.openwrt.org](http://www.openwrt.org): Embedded linux used in Dragino.
* [wiki.dragino.com](http://wiki.dragino.com/index.php?title=Main_Page): General software/hardware design info for Dragino MS12
* [www.xively.com](http://www.xively.com): A public IoT RESTful server.
* [www.yeelink.com](http://www.yeelink.com): A public IoT RESTful server used in China.
* <https://github.com/reeedstudio/Cloud_Dragrove>: Daughter board firmware code
* <https://github.com/reeedstudio/Cloud_Dragino_Firmware>：Dragino Firmware