

如何使用 WisCore kit 打造一个 DIY 智能家居最小系统

该文章介绍如何配置 node-red 设备，从而允许 Alexa 不需要调用 Skill 就可以控制带有 Wi-Fi 功能的智能设备。没错，的确不需要通过 Skill 就可以控制哦。让我们一起深入研究一下具体的操作步骤吧

类似 Alexa 能够直接发现 Hue 设备一样，我们需要找出 node-red 如何能够提供一种手段来仿真这类设备。幸运的是，有一个非常棒的代码已经可以在网上找到，就是为了实现这个目的。在 Originally US 上面，已经有人开发了这样一个模块，叫做 Alexa-Local，下面是具体的链接：

<https://github.com/originallyus/node-red-contrib-alexa-local>。

让我们更深入的了解如何来进行设置，

硬件配置：

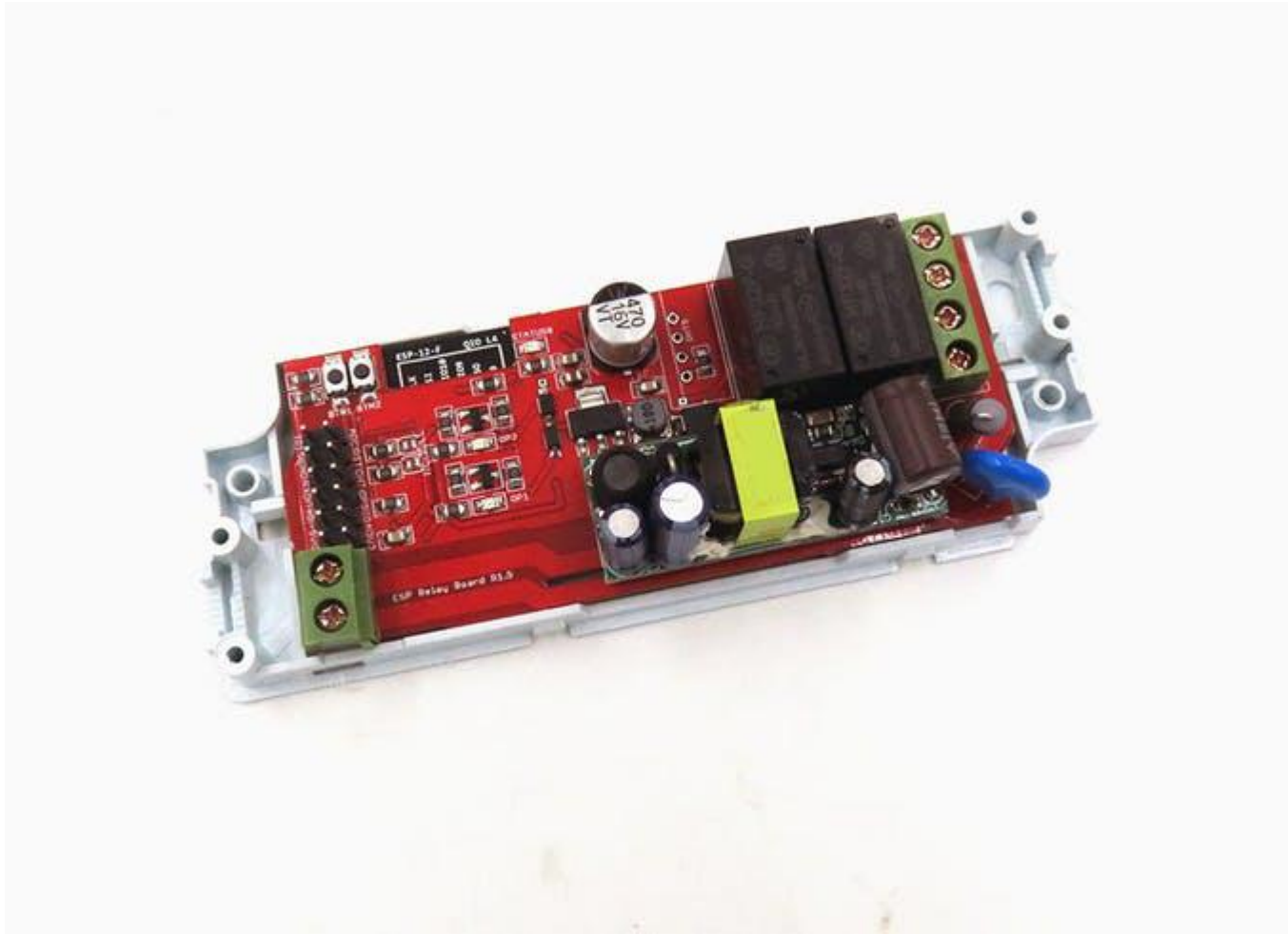
在硬件方面，我们需要 Raspberry PI 和一个 8G micro SD 卡，单独的一个 wifi 控制的开关。在我们的实验中，我们会用到 Electrodragonwifi 开关，链接如

下：<http://www.electrodragon.com/product/wifi-iot-relay-board-based-esp8266/>

至于如何配置上面的开关，可以下载 Tasmota firmware，链接如下：

- <https://www.hackster.io/naresh-krish/home-automation-using-wiscore-and-openhab-1ec6e4>
- <https://github.com/arendst/Sonoff-Tasmota>

PO 一张设备的靓照：



Raspberry PI 的板子会运行安装的 node-red : 一个非常精心制作的开发指南呈现给大家: <https://nodered.org/docs/hardware/raspberrypi>

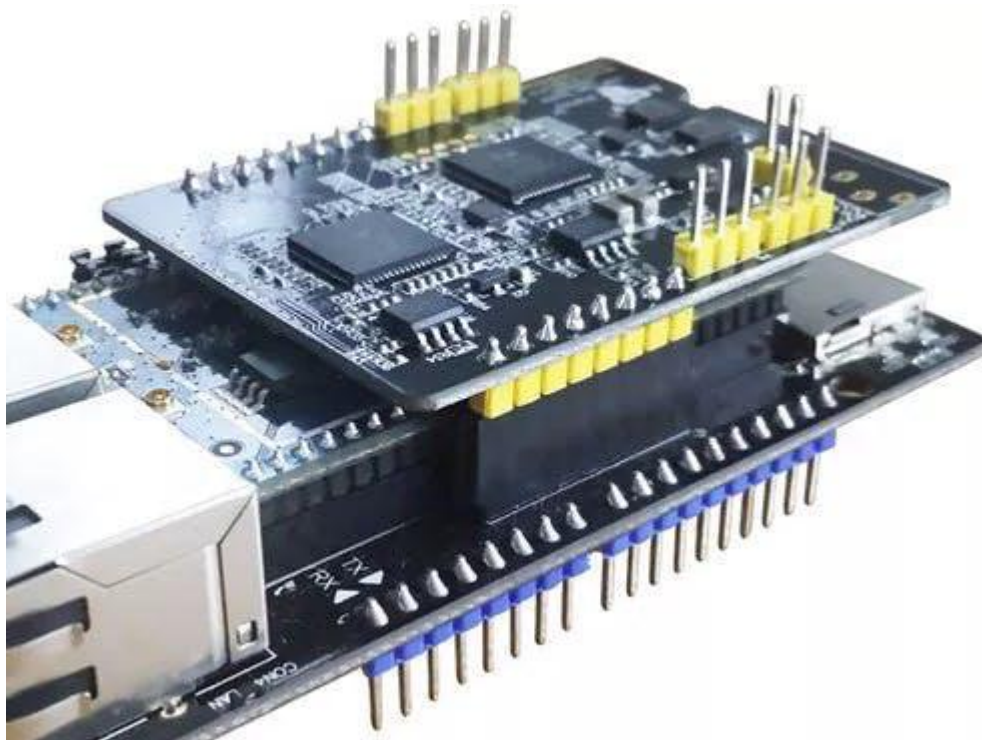


同时,我们还需要用到 RAK Wireless 公司生产的 WisCore Alexa Kit 开发套件,该套件集成了 Alexa Voice Service 功能,可以作为一个 Echo dot, Echo, Tap 来使用。WisCore Alexa Kit 是一款开源开发套件,核心芯片采用了联发科的 MT7628A 芯片,上面运行的是 OpenWRT 系统。使用该套件,可以直接通过套件自带的固件或者网上更新的固件

(<https://github.com/RAKWireless/WisCore/tree/master/release/wiscore>) 直接使用,也可以编译网上提供的 SDK (教程:<https://github.com/RAKWireless/WisCore/wiki/WisCore>), 自己在上面增加其他功能

很酷的套装来袭





一旦这两个部分都已经配置好之后，让我们开始我们最初的试验目标。

配置 RAK WisCore Alexa kit:

配置手上的 WisCore Alexa kit 链接到 Alexa Voice service, 具体步骤可以参考以下链接: <https://github.com/RAKWireless/wiscore/wiki/Login-Alexa-Through-App>

Node-red setup:

一旦你的 Raspberry PI 设置好 Node-Red。打开一个浏览器在你的桌面上，然后运行

```
http://<ip of rpi>:1880
```

你将会得到一个 Node-red 白板，有一些 nodes 已经上传并且准备好用于你的画板中

请注意：不是所有人都想要运行 node-red 在端口 1880 上。如果你需要更改端口，也一定要同步再上面的 URL 上进行更改

现在，我们需要安装一个很酷的 node，叫做 Alexa-Local node，由 folks 在 OriginallyUS 提供。这个 node 的作用是模拟你的节点作为一个 Hue 终端。Alexa 将会发现这个服务并且添加这个设备到他的清单里面。

手动安装 nodejs 和 node-red

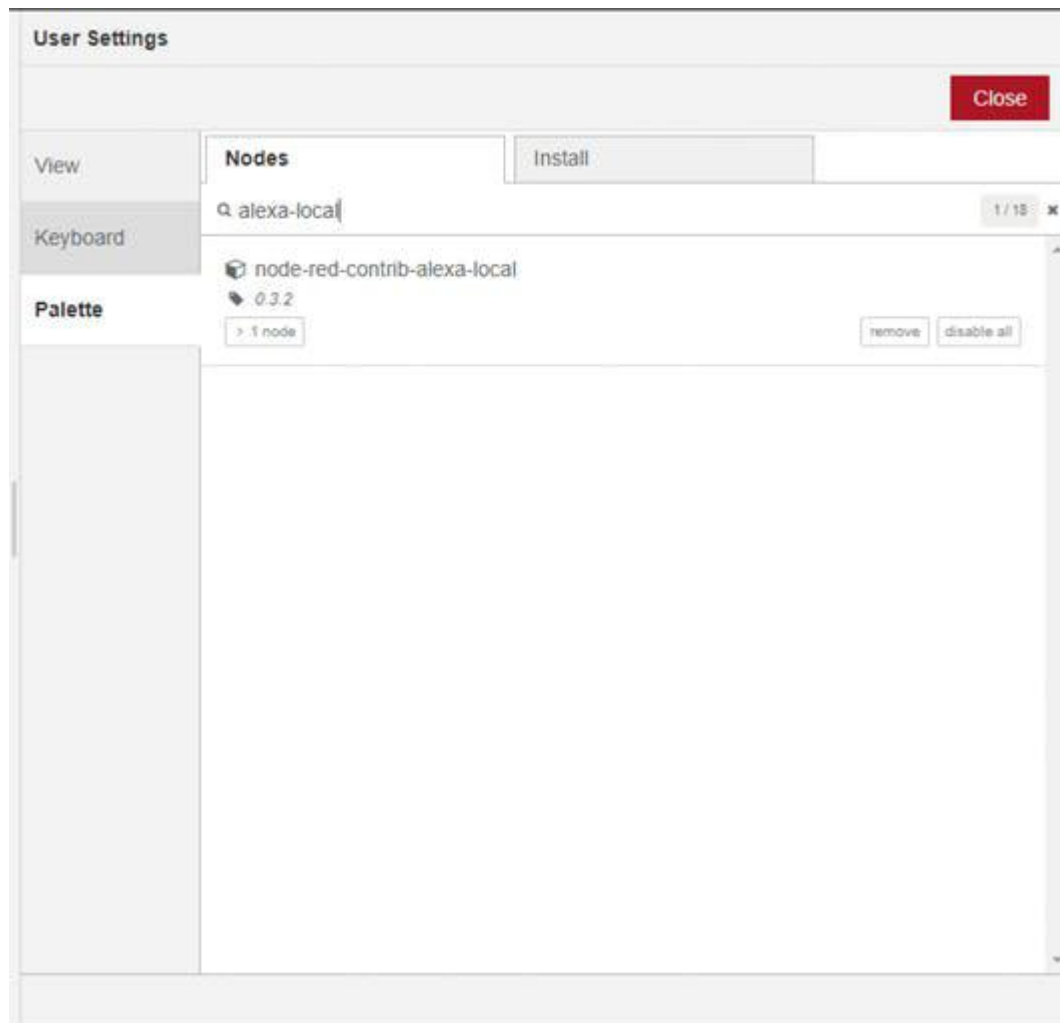
按照以下步骤，我们可以尝试手动安装 node-red
安装 nodejs:

```
wget http://node-arm.herokuapp.com/node_latest_armhf.deb  
sudo dpkg -i node_latest_armhf.deb
```

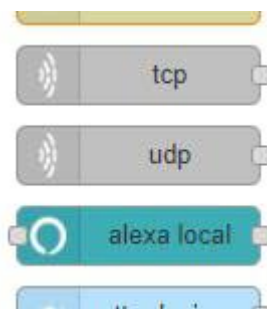
安装 node-red

```
sudo npm install -g --unsafe-perm node-red
```

点击右上角设置图标，进入节点板菜单



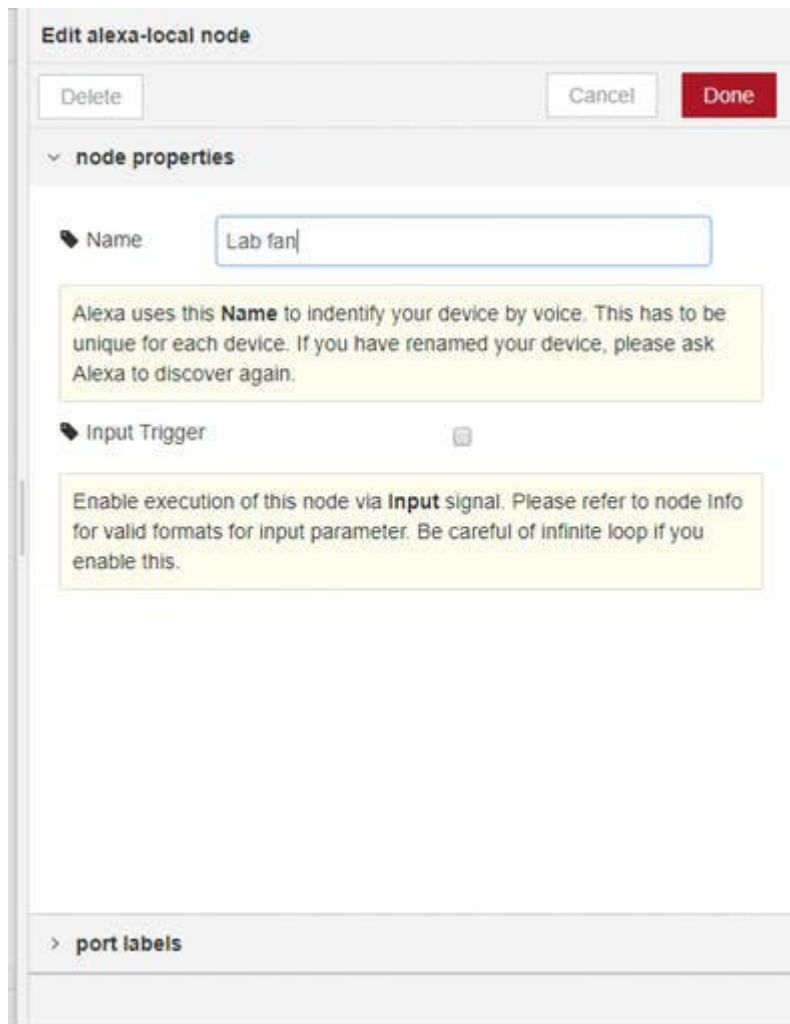
搜索 Alexa-local 并且点击安装 node。你可以看到一个 Alexa node 列表在在左侧的滚动条内，如下图所示：



拖放一个这个节点的实例如下所示。同时再拖一个调试用的 node 在下面这个状态，连接是一样的，如下所示



debug 节点的作用只是为了让我们知道，当 Alexa 被触发时会有相应。双击 Alexa-local node，可以弹出一个菜单页，如下图：



现在，我们把这个节点命名。根据你想要控制的类型来进行命名。在我的示例中，我叫这个节点为 Lab Fan。设置 input trigger 为 off。这样该节点就不会因为其他节点输出的信息而被触发。

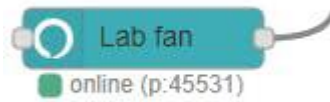
点击 node-red 界面的右上角 Deploy 按钮来完成。一旦配置好，通过语音对着 WisCore Kit 说：

Alexa, Discover devices.

Alexa 会需要一些时间来发现活跃的设备。由于我们的设备是一个本地设备，所以发现的过程会非常快。我们会收到回复说 Alexa 已经发现了一个或多个设备。

这个时候，我们就可以检查一下设备是否已经注册到你的 Alexa 账号下。在 Android 或 IOS 设备上打开 Alexa app，然后进入到 Smart Home 选项，你可以发现一个跟你之前命名 node 一样的设备显示在界面上。

请注意：如果你的 node 没有起来，请确认在配置好之后 node 会显示连接状态，如下图



这时，你的触发节点已经正常工作了。让我们连接一个 wifi 设备。在我的实验中，我使用了 Electrodragon 根据 SonoffTasmota 固件，针对 ESP8266 的模块。这个固件功能比较强大，支持 OTA 升级，基于 WEB API 的 HTTP/MQTT 我们会根据我们的需要选择 HTTP API。

想要控制一个开关，我们可以要求 ESP8266 通过以下指令进行控制

```
http://sonoff/cm?cmd=Power%20TOGGLE
http://sonoff/cm?cmd=Power%20On
http://sonoff/cm?cmd=Power%20off
http://sonoff/cm?user=admin&password=joker&cmd=Power%20Toggle
```

当我们的 IP 地址有变化时，Sonoff 可以重置

让我们把这个需求添加到我们的 node-red 工作流程当中。添加一个 HTTP 请求 node，把这个 node 链接到 Alexa node。增加一个 HTTP 返回 node 放在 HTTP 请求节点后面。这样，我们就能够输出该请求的状态，例如失败，成功等等。



双击 HTTP 请求 node，

Dialog box titled "Edit http request node" with buttons "Delete", "Cancel", and "Done".

Section: **node properties**

- Method: GET
- URL: http://sonoff/cm?cmd=Power%20TOGGLE
- Enable secure (SSL/TLS) connection:
- Use basic authentication:
- Return: a UTF-8 string
- Name: Name

port labels

由于这个是一个非常简单的 HTTP 请求，所以要关闭 SSL 和认证。输入我们刚刚在上面提到的 URL。

请注意：这里要把 sonoff 替换成 URL，显示 IP 地址

现在，所有的准备工作都已经完成。抑制不住激动的心情，让我们开始体验吧：当你对着你的 WisCore 说：

Alexa, toggle Lab Fan.

Alexa 会发送 TOGGLE 指令给到 Alexa local node，local node 将会触发 HTTP 请求来控制开关的开启或者关闭。

是否已经掌握了呢？你也可以尝试其他的一些智能设备，让我们用语音来开启新的世界吧！