

使用 **Amazon Echo** 物联网智能音箱 控制 **RGB LED** 灯条

Phodal Huang

September 8, 2017

目录

步骤一：材料准备	3
步骤 2：用 Raspberry Pi 设置 RGB LED 灯条	3
步骤 3：搭建 Nginx	3
步骤 4：配置静态 IP	4
步骤 5：让 Raspberry Pi 和 Amazon Echo 一起工作	4
步骤 6：用 Pi 完成 RGB LED 条	5
步骤 7：设置您的 Amazon Echo	6
步骤 8：Amazon Echo Code 如何工作	7
步骤 9：在 Amazon Developer Service 配置 Amazon Echo	8
步骤 10：连接你的 Raspberry Pi 与 Amazon Echo	11

玩点什么：<https://www.wandianshenme.com>

原文链接：<https://www.wandianshenme.com/play/smart-speaker-amazon-echo-control-rgb-led-strip>

本文是一篇关于如何从您的 Amazon Echo 控制 RGB LED 灯条的指南。您将能够告诉 Echo 打开和关闭 LED 条，同时改变其亮度和颜色之间的渐变。除了此，如果进一步自定义，还可以执行更多操作。

以下是这个教程的视频 DEMO：

步骤一：材料准备

- Amazon Echo
- Raspberry Pi B+ 及面包板套件
- RGB LED 灯条
- 三个 MOSFET（最大阈值电压 3.3v）
- 电源（12V DC □ 2A）
- 电源插孔
- 面包板
- 跳线（公对公）和（公对母）

你可能会发现上面的内容只会使用到一些，而不是购买上面的整个套件。但无论如何，这是一个很好的方式进入电子及 Raspberry Pi 的世界。

步骤 2：用 Raspberry Pi 设置 RGB LED 灯条

首先，我们将从 Raspberry Pi 开始设置 RGB LED 灯条——这个需要在开始搭建之前完成。所以我们可以按照 Popoklopsi 的这个指南：<http://dordnung.de/raspberrypi-ledstrip/>

指南里完全解释了每一部分！唯一的区别是，我的红线从 RGB 条连接到我的 Raspberry Pi 的 GPIO 上的引脚 17，蓝色连接到引脚 24，绿色连接到 27。一旦可以成功地在红色，绿色，蓝色或蓝色之间切换，就可以使用命令“pigs p 17 255”来混合任何其他颜色。

接下来让我们进入下一部分。

步骤 3：搭建 Nginx

所以，我们现在可以从 Raspberry Pi 的命令行来改变灯条的颜色。但是，我们想做的是从 Amazon Echo 来改变颜色。

首先让我们设置一个 Web 服务器, 让 Amazon Echo 与 Pi 进行通信。按照此官方指南在您的 Raspberry Pi 上设置 nginx: <https://www.raspberrypi.org/documentation/remote-access/web-server/nginx.md>

确保也安装好了 PHP!

您现在应该能够将您的 Pi 的 IP 输入到您的 URL, 并且可以从您家中的任何计算机连接到该网站。网站将显示 nginx 默认屏幕或默认的 PHP 页面。

步骤 4: 配置静态 IP

我们需要确保这个 IP 保持不变, 所以我们将使您的 Raspberry Pi 具有静态 IP。这是一个简单的指南: <https://www.modmypi.com/blog/tutorial-how-to-give-your-raspberry-pi-a-static-ip-address>

如果指南已经过时, 那么您可能需要自行搜索; 如果您使用的是 WiFi, 可能会略有不同。

好了, 现在, 您的 Raspberry Pi 正在托管一个具有静态 IP 的 Web 服务器, 因此您始终知道托管的位置。

步骤 5: 让 Raspberry Pi 和 Amazon Echo 一起工作

接下来, 我们将在 python 中编写一个脚本, 来检查 Raspberry Pi 上的文本文件的内容。我们将这个脚本称之为 controller.py, 对应的文本文件将称之为 settings.txt。controller.py 脚本将不断循环地检查 settings.txt 文件的内容。根据 settings.txt 文件的内容, controller.py 脚本将会通过 Raspberry Pi 上安装的 pigpio 库来控制 Raspberry Pi 上的 GPIO 引脚 (17,24,27)。这样一来, LED 的变化将类似于我们使用 pig p 命令控制。

settings.txt 文件将会有 5 个数字, 前 3 个用于确定灯的亮度, 从 0 到 255 的任何值。第四个数字将为 0 或 1, 用以指示淡入淡出效果是否活动, 以便于在不同颜色之间渐变 (0 表示关闭, 1 表示打开)。最后一个数字将表示颜色 0-9, 同时 controller.py 脚本将根据您分配的数字, 将 GPIO 引脚更改为显示来自 LED 条的某种颜色。

controller.py 脚本将不断循环地检查 settings.txt 文件的值。它从文件中获取 5 个数字, 并解析它们以正确读取信息。然后使用 pigpio 库——根据文件中的数字运行 pig p 命令, 以修改 LED 的颜色。

最后一件事是, 为了让 controller.py 脚本获取 settings.txt 文件的内容, 它必须向 php 文件发送一个请求。我们将把这个文件称为 getColor.php, 这个文件将简单地为我们抓住 settings.txt 的内容。

<https://github.com/adza45/echo/tree/master/Raspberry%20Pi%20Code>

您需要将这些文件，下载到 Web 服务器上与 Nginx 相同的位置。通过会放在：

```
1 cd /var/www/html
```

步骤 6: 用 Pi 完成 RGB LED 条

当前的配置使淡绿色的灯变为全亮度，同时关闭淡入淡出效果。因为 `settings.txt` 文件中的前三个数字是 `255`，这是最大亮度；第四个数字是淡入淡出，它为零，所以淡入淡出设置是关闭的；最后一个数字是 `1`，在 `controller.py` 代码中可以看到，如果该数字为 `1`，则将引脚 `27` 设置为高电平为绿色。

一旦文件 `settings.txt` 在该目录中，我们可以测试这部分应用程序。通过键入以下命令运行 `controller.py python` 脚本：

```
1 python controller.py
```

如果 `settings.txt` 文件的当前内容是 `25501`，则指示灯应变为绿色。打开第二个终端并导航到 `settings.txt` 文件：

```
1 cd /var/www/html
```

打开 `settings.txt` 文件：

```
1 sudo nano settings.txt
```

修改文件中的数字，并确保在修改后保存并退出，这样 LED 灯条时相应地更改颜色。记住第一个数字是强度，第四个是你是否要渐变，最后一个是相应颜色的数字。

现在，我们可以根据 `settings.txt` 文件的内容改变 LED 的颜色。但是我们还没有碰到 Amazon Echo，不过我们快接触它了。

So，我们需要 Amazon Echo 能够与 Pi 所在的 Web 服务器进行通信，以便修改 `settings.txt` 文件的内容，可以改变 LED 条颜色。我们可以通过我们的 Web 服务器实现这一点。

首先，我们需要在目录 `/var/www/html` 中更改 `index.php` 文件的内容。我们需要这个文件，能允许来自外部来源的获取请求，以修改 `settings.txt` 的内容。`index.php` 文件位于与之前相同的代码库中，因此请确保您下载该文件，并使用该文件覆盖您的 `index.php`。

步骤 7: 设置您的 **Amazon Echo**

在我们的 Web 服务器上, 托管的 `Index.php` 可以让某人写入内容到 `settings.txt` 文件。你可以看到这个文件有变色、强度和淡入淡出, 这些值写入 `settings.txt` 文件。但是我们如何将变量设置为相等的东西? 由于我们在 Web 服务器上托管此代码, 因此我们可以通过查询字符串设置值。然后, 代码从我们编写的查询字符串中获取值。例如, 我们可以键入:

```
1 http://192.168.1.43/index.php?intensity=255&fade="0"&color="0"&get=1
```

在我们的浏览器中, 服务器将根据上面输入的内容设置亮度、淡入淡出和颜色变量。这时需要确保 **Raspberry Pi** 上仍运行 `controller.py` 脚本, 然后在您家中的任何计算机上, 键入上面的行到任何浏览器。需要确保您为 **Raspberry Pi** 设置的静态 IP, 替换到上述 URL 中的 IP。这时灯应该发生变化! 然而实际上他们可能不会, 因为没有足够的权限写入 `settings.txt` 文件。为了允许我们这样做, 需要到 **Pi** 上的 `/var/www/html` 目录, 然后键入以下命令:

```
1 sudo chmod 757 settings.txt
```

而这是超级不安全的, 因为任何人都可以编辑文件。不过, 由于我们只是测试一个灯开关, 他们不能做太多危险的操作。只是不要用它 (**Raspberry Pi**) 来存储非常重要的东西。你可以研究其他方式来保证它的安全。但现在的方式是最快的, 可以让初学者轻松上手。

好的, 现在返回并重新运行你的 `controller.py` 脚本。现在输入您的浏览器的请求, 你的灯应该相应改变。如果不能确保 `settings.txt` 文件, 能从请求中更改文本内容。请检查并查看您的 `controller.py` 是否停止运行, 以及是否显示错误。

对于上一部分, 可以看相应的 Youtube 的内容 <https://www.youtube.com/watch?v=n6u-H1sza-w>

虽然与我的代码不同, 但是他提供了将文本文件、循环 python 脚本、Web 服务器相结合的知识, 以用于修改文本文件的内容, 并最终改变颜色。

好的, 现在是使用 **Amazon Echo** 的时候了。**Amazon Echo** 允许您创建 `skills`, 如您可以购买比萨饼或二十个问题游戏的 `skill`。我们将创造自己的 `skill`, 来改变灯光的颜色。如果你从来没有创造过 **Amazon Echo** 的 `skill`, 你可以阅读如下的资料:

<https://developer.amazon.com/public/solutions/alexa/alexa-skills-kit/getting-started-guide>

我们为 **Echo** 编写的代码, 需要向我们的 Web 服务器发送一个获取请求, 以修改指示灯。这是我过去使用的代码:

<https://github.com/adza45/echo/tree/master/Alexa%20Code>

步骤 8: Amazon Echo Code 如何工作

基本上,这段代码与您的网络浏览器一样,也可以开/关开灯,其颜色和淡入淡出效果。在 `exports.js` 中,我有不同的功能,用于根据你所说作出响应。所以,如果你说“Alexa, turn on the light”,它会运行点亮灯的指令,即在 `exports.js` 文件中运行 `LightsOn`。代码如下:

```
1 function LightsOn(intent, session, callback) {
2     var cardTitle = intent.name;
3     var repromptText = "";
4     var sessionAttributes = {};
5     var shouldEndSession = true;
6     var speechOutput = "";
7     var tools = require('./tools');
8
9     tools.ChangeColorRequest("255", "pass", "pass", function(result) {
10         speechOutput = "The lights are now on.";
11         repromptText = "The lights are now on.";
12         callback(sessionAttributes,
13             buildSpeechletResponse(cardTitle, speechOutput, repromptText,
14                 shouldEndSession));
15     });
16 }
```

点亮灯的过程中,将使用 `tools.js` 文件来向 Raspberry Pi 上托管的 `index.php` Web 服务器发送请求。我传递值为 255 的灯光强度,并为颜色和渐变赋值为“pass”,以便只引起 `settings.txt` 文件中的亮度变化。所以,当你在下次打开灯时关掉它,就和之前的一样。其他功能的工作方式也是相似的,当更改颜色时,我键入 `pass` 的强度和渐变,然后传递一个数字为最后一个选项,将 `settings.txt` 文件中的最后一个数字更改为正确的颜色。

在 `tools.js` 中,需要确保将主机名更改为您的 ISP 给您的 IP。只需输入谷歌“我的 IP 是什么”,这个结果就是你想要的内容。

我使用 `test.js` 文件来测试:能从我的电脑更改 `settings.txt` 文件的值。我在 `notepad++` 中输入了这个代码,并添加了一个插件,以让我能运行 `node.js` 代码。以下是允许 `notepad++` 运行 `node.js` 代码的方法:

<http://blog.aguskurniawan.net/post/notepadjs.aspx>

要使此代码与 Amazon Echo 一起工作, 您需要从此站点 (Amazon Management Console) 创建一个 lambda 函数:

<https://us-west-2.console.aws.amazon.com/>

请在 amazon 管理控制台中, 在设置中将 “handler” 更改为 “exports.js”, 因为我们调用主文件 exports。

当您将此代码上传到亚马逊管理控制台时, 您必须制作一个 zip 文件, 因为我们正在上传多个文件。所以将 exports.js 和 tools.js 文件压缩在一起, 并将其压缩文件夹导出。然后将该压缩文件, 上传到 amazon 管理控制台中的 lambda 函数。

步骤 9: 在 Amazon Developer Service 配置 Amazon Echo

为了使用 Amazon Echo 中的 Skill, 我们需要访问:

<https://developer.amazon.com/>

点击 Alexa 标签 -> 点击 Alexa Skills Kit。在这里, 您将设置您的 Alexa Skill 如何更改灯条。您也将链接到刚才做的 Lambda 函数, 以便让程序知道要运行哪一个代码:

在交互模式下, 当您创建 skill 时, 您将需要一个 intent schema。如下:

```
1 {
2   "intents": [
3     {
4       "intent": "ChangeColor",
5       "slots": [
6         {
7           "name": "Color",
8           "type": "COLOR"
9         }
10      ]
11    },
12    {
13      "intent": "ChangeIntensity",
14      "slots": [
15        {
16          "name": "Intensity",
```



```
17         "type": "AMAZON.NUMBER"
18     }
19 },
20 {
21     "intent": "LightsOn"
22 },
23 {
24     "intent": "LightsOff"
25 },
26 {
27     "intent": "DimLights"
28 },
29 {
30     "intent": "BrightenLights"
31 },
32 {
33     "intent": "MaxIntensity"
34 },
35 {
36     "intent": "MinIntensity"
37 },
38 {
39     "intent": "FadeOn"
40 },
41 {
42     "intent": "FadeOff"
43 },
44 {
45     "intent": "Exit"
46 }
47 ]
48 }
```

以下是我的说话样本:

```
1 ChangeColor change color to {Color}
2 ChangeColor change to {Color}
```

```
3 ChangeColor turn lights to {Color}
4 LightsOn turn lights onLightsOff turn lights off
5 LightsOn lights on
6 LightsOff lights off
7 Exit exit
8 Exit quit
9 Exit quit god
10 Exit goodbye god
11 Exit goodbye
12 ChangeIntensity change intensity to {Intensity}
13 ChangeIntensity change brightness to {Intensity}
14 ChangeIntensity change intensity {Intensity}
15 ChangeIntensity change brightness {Intensity}
16 DimLights dim the lights
17 BrightenLights brighten the lights
18 DimLights intensity down
19 BrightenLights intensity up
20 DimLights brightness down
21 BrightenLights brightness up
22 MaxIntensity turn to maximum intensity
23 MaxIntensity turn to maximum brightness
24 MaxIntensity maximum intensity
25 MaxIntensity maximum brightness
26 MinIntensity turn to minimum intensity
27 MinIntensity turn to minimum brightness
28 MinIntensity minimum intensity
29 MinIntensity minimum brightness
30 FadeOn turn fade on
31 FadeOn fade on
32 FadeOff turn fade off
33 FadeOff fade off
```

另外对于交互模式 (**interaction modal**) 中的自定义插槽类型 (**custom slot types**), 我创建了我自己的“**COLOR**”, 其中的值如下所示:

```
1 RED
2 BLUE
```

- 3 PURPLE
- 4 GREENYELLOW
- 5 AQUA
- 6 WHITE
- 7 BLACK
- 8 ORANGE
- 9 CYAN

步骤 10: 连接你的 **Raspberry Pi** 与 **Amazon Echo**

好的，所有的一切似乎都很好，但我们需要做最后一件事。

由于您的 **Amazon Echo** 代码托管在云端，如果您尚未设置端口转发，则无法在您的 **Web** 服务器上访问 **index.php**。因此，我们在您的路由器中执行此操作，但是对于每个路由器都是不同的，希望您不必通过多个路由器，因为这可能会使事情变得复杂。无论如何，从浏览器到主路由器主页，最有可能是 **192.168.1.1** 或 **192.168.200**。并在搜索端口转发选项卡并配置它。

并确保将 **IP** 地址，修改为 **Raspberry Pi** 配置的静态 **IP**。还要祈祷 **ISP** 不阻止访问端口 **80**，这就并不是什么大问题。只需要更改正在托管的 **nginx** 端口，并更改上面对应的端口。可以轻松地通过 **Google** 来解决这个问题。

好的，所以现在你可以，从 **Amazon Developer Service** 在 **test** 标签下测试你的 **skill**。如果一切都奏效，你应该可以通过调用 **skill**，来运行我上面提供的任何一个示例语句！在 **amazon** 开发者服务的技能信息选项卡中，我命名了调用名称“**lights**”。所以我必须说的是“**Alexa, ask lights to turn on.**”或者“**Alexa, ask lights to change to cyan.**”。很酷吧！

祝你好运！

原文地址：<http://www.instructables.com/id/LED-Strip-Controlled-by-Amazon-Echo/>

原文链接：<https://www.wandianshenme.com/play/smart-speaker-amazon-echo-control-rgb-led-strip/>