

DIY Alexa 智能音箱：基于 **Raspberry Pi + Snowboy + TrulyHandsFree + AVS**

Phodal Huang

September 8, 2017

目录

硬件准备	3
Go	3
步骤 1: 设置你的 Pi	3
步骤 2: 注册一个 Amazon 开发者帐户	4
步骤 3: 创建设备和安全配置文件	4
步骤 4: 克隆示例应用程序	5
步骤 5: 使用你的凭据更新安装脚本	5
步骤 6: 运行安装脚本	6
步骤 7: 运行您的 Web 服务, 示例应用程序和唤醒文字引擎	6
终端窗口 1	7
终端窗口 2	7
终端窗口 3	8
步骤 8: 与 Alexa 交谈	8
步骤 9: 如何注销示例应用程序	9

原文链接: <https://www.wandianshenme.com/play/use-snowboy-trulyhandsfree-raspberry-pi-avs-bu>

本文详细地介绍了,如何一步步地在 Raspberry Pi 上搭建 AVS (Alexa Voice Service) 服务。它演示了如何使用 Amazon 的 Java 示例应用程序 (在 Raspberry Pi 上运行), Node.js 服务器, 以及第三方唤醒文字引擎来访问和测试 AVS。您将使用 Node.js 服务来登录获取具有 Amazon (LWA) 授权码的登录, 这个可以通过您的 Raspberry Pi 上的 Web 浏览器来访问。

硬件准备

在你开始之前, 让我们来看看你需要的材料:

1. 推荐使用 Raspberry Pi 3, Pi 2 Model B 也是支持的
2. 用于 Raspberry Pi 上的 Micro-USB 充电线
3. 安装有操作系统 (建议使用 NOOBS) 的 Micro SD Card (至少要 8 Gb)
4. USB 2.0 Mini 麦克风
5. 外置扬声器
6. 一个 USB 键盘和鼠标, 以及 HDMI 显示器
7. 网络连接
8. (可选) 用于 Raspberry Pi 2 的 WiFi 无线适配器

另外, 我们还将向你展示如何远程 (SSH) 登录到你的设备, 而无需显示器、键盘和鼠标, 以及如何 tail 日志来排错。

Go

原先的用于 Raspberry Pi 上的 Alexa, 需要手动下载一系列的库、依赖、并更新一系列的配置, 这些对于人来说, 都是容易出错的。为了使过程更快更容易, 我们已经在该项目中添加了一个安装脚本, 该脚本将负责所有的重要操作。这不仅将 Raspberry Pi 3 上的安装时间缩短到不到一个小时, 而且它只需要开发人员, 在单个安装脚本中调整三个变量。

步骤 1: 设置你的 Pi

如果你没有在 Raspberry Pi 上安装 Raspbian Jessie, 那么你可以根据这个指南来完成: [Setting up the Raspberry Pi](#)。它将引导您下载、安装 Raspbian Jessie, 并连接硬件

(如果您不熟悉 Raspberry Pi, 我们强烈推荐您, 按照上面的指南来设置您的 Pi, 然后再进行下一步)。

步骤 2: 注册一个 Amazon 开发者帐户

如果你没有 Amazon 的开发者账户, 你需要到 developer.amazon.com 上注册一个, 并查看阅读 [Alexa Voice Service Terms and Agreements](#)。

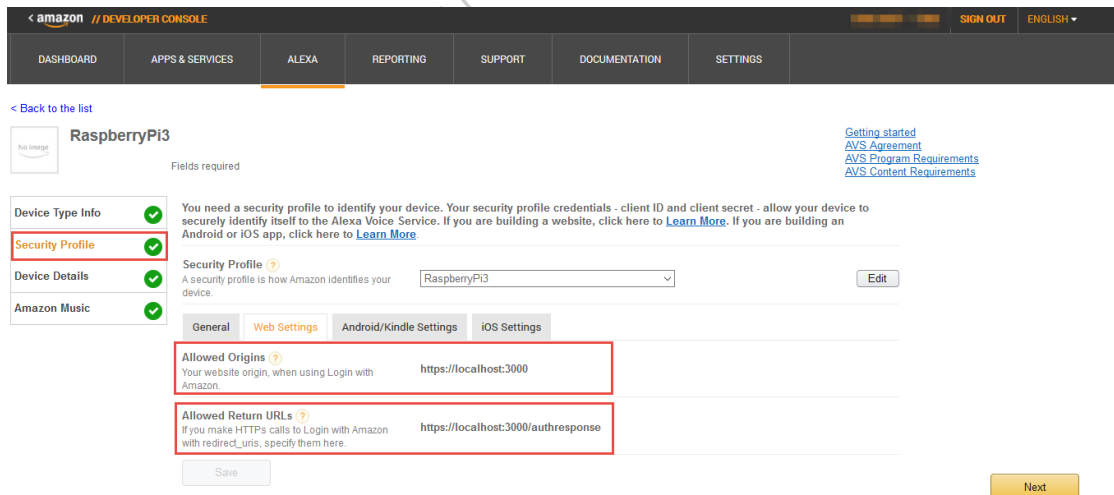
步骤 3: 创建设备和安全配置文件

首先, 按照此处的步骤 ([Create Security Profile](#)) 注册您的产品, 并创建一个安全配置文件。

并记下以下参数。您将需要在下面的“步骤 5”中使用到。

- ProductID
- ClientID
- ClientSecret

注意: 请确保你的 **Allowed Origins** 和 **Allowed Return URLs** 被设置在 **Security Profile > Web Settings** 之下 (详见: [Create a device and security profile](#)) :



The screenshot shows the Amazon Developer Console interface for configuring a device named 'RaspberryPi3'. The 'Web Settings' tab is active, and the 'Allowed Origins' and 'Allowed Return URLs' fields are highlighted with red boxes. The 'Allowed Origins' field contains 'https://localhost:3000' and the 'Allowed Return URLs' field contains 'https://localhost:3000/authresponse'. The 'Save' button is visible at the bottom left, and the 'Next' button is at the bottom right.

Allowed Origins: <https://localhost:3000> **Allowed Return URLs:** <https://localhost:3000/authresponse>

步骤 4: 克隆示例应用程序

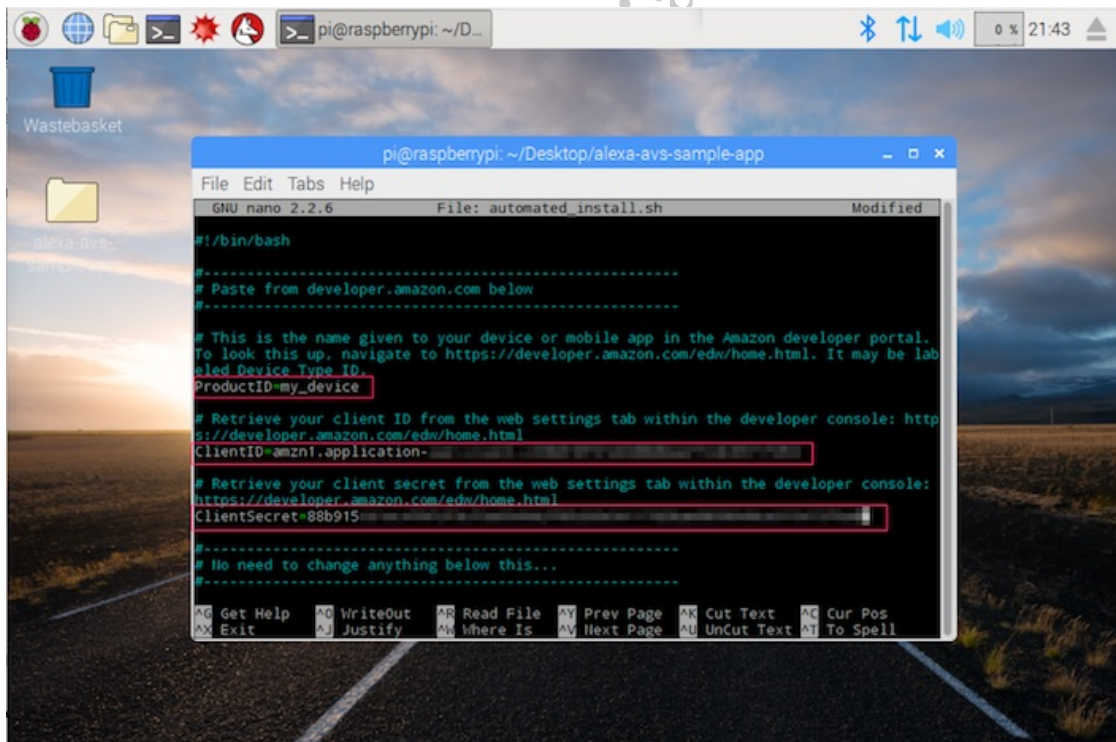
打开终端，并键入以下内容：

```
1 cd Desktop
2 git clone https://github.com/alexa/alexa-avs-sample-app.git
```

步骤 5: 使用你的凭据更新安装脚本

在运行安装脚本之前，您需要使用“步骤 3”中获得的凭据更新脚本：ProductID、ClientID、ClientSecret。在终端中键入以下内容：

```
1 cd ~/Desktop/alexa-avs-sample-app
2 nano automated_install.sh
```



粘贴您从上述“步骤 3”获得的 ProductID, ClientID 和 ClientSecret 的值。变更应如下所示：

```
1 ProductID="RaspberryPi3"
2 ClientID="amzn.xxxxxx.xxxxxxxxxx"
3 ClientSecret="4e8cb14xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx6b4f9"
```

输入 **ctrl-X**，然后输入 **Y**，然后按 **Enter** 键保存对文件的更改。

步骤 6: 运行安装脚本

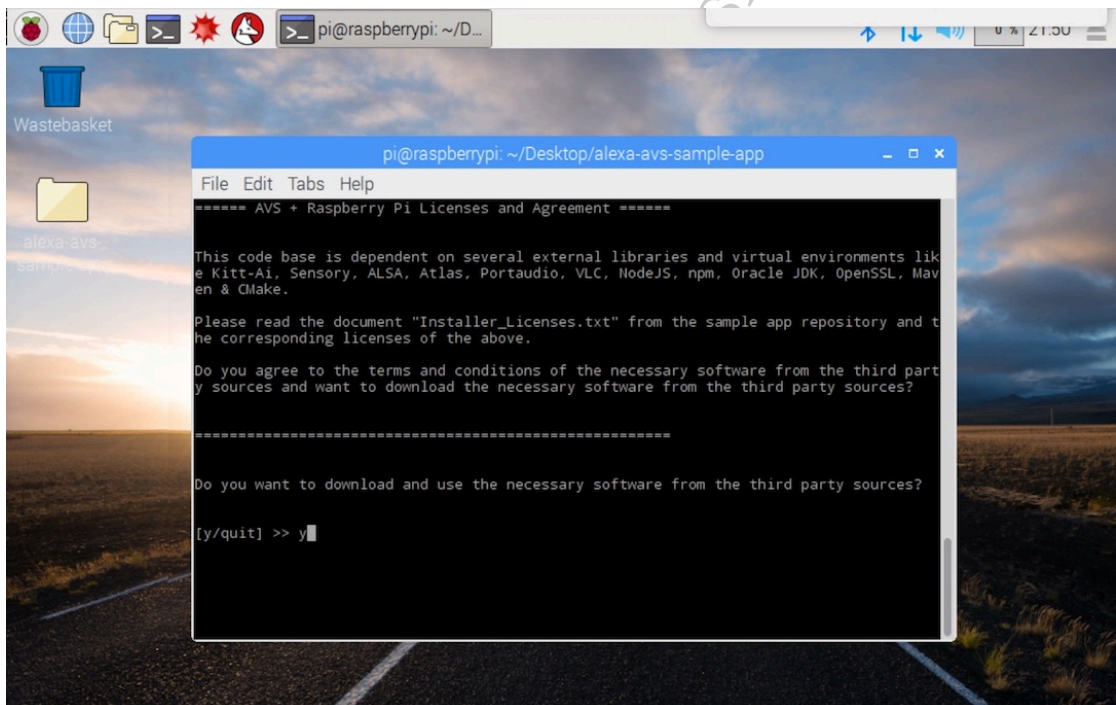
现在, 您可以运行安装脚本了。这将安装所有依赖项, 包括 **Sensory** 和 **KITT.AI** 中的两个唤醒字引擎。

注意: 安装脚本将在运行脚本的文件夹中, 安装所有项目文件。

要运行脚本, 请打开终端并导航到项目克隆的文件夹。然后运行以下命令:

```
1 cd ~/Desktop/alexa-avs-sample-app
2 . automated_install.sh
```

系统会提示您回答几个简单的问题。这些将有助于确保您在继续之前, 已经完成了所有必要的先决条件。



在法术开始的时候, 你可以去拿一杯咖啡 - 安装大约需要 **30** 分钟。

步骤 7: 运行您的 **Web** 服务, 示例应用程序和唤醒文字引擎

现在安装已经完成, 您需要在 **3** 个独立的终端窗口中运行三个命令:

- 终端窗口 **1**: 运行 **Web** 服务进行授权
- 终端窗口 **2**: 运行示例应用程序与 **AVS** 通信
- 终端窗口 **3**: 运行唤醒字引擎, 允许您使用短语 **"Alexa"** 开始交互。

注意: 这些命令必须按顺序运行。

终端窗口 1

打开一个新的终端窗口, 并键入以下命令来启动: 用于使用 **AVS** 授权您的示例应用程序的 **Web** 服务:

```
1 cd ~/Desktop/alexa-avs-sample-app/samples
2 cd companionService && npm star
```

服务器现在在 **3000** 端口上运行, 您已准备好启动客户端。

在 [API Overview > Authorization](#) 里, 你可以了解到更多关于授权的信息。

终端窗口 2

打开一个新的终端窗口, 并键入以下命令来运行, 与 **AVS** 通信的示例应用程序:

```
1 cd ~/Desktop/alexa-avs-sample-app/samples
2 cd javaclient && mvn exec:exec
```

从 [API Overview > Interfaces](#) 中可以了解到与 **AVS** 消息发送和接收的内容。

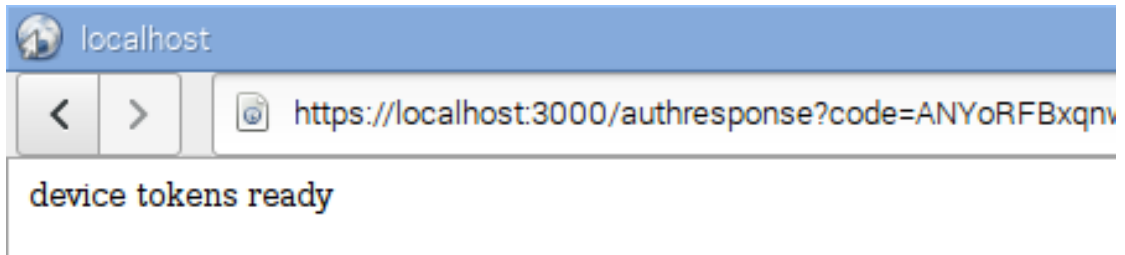
我们来看看与终端窗口 2 相关的后续几个步骤:

1. 当您运行客户端时, 窗口应弹出一条消息, 说:

```
Please register your device by visiting the following URL in a web
browser and following the instructions: https://localhost:3000/pro-
vision/d34of629bd685deeff28a917. Would you like to open the URL
automatically in your default browser?
```

点击 “**YES**” 来使用默认的浏览器来打开 URL。

2. 如果您使用 **Pixel** 桌面 (和 **Chromium** 浏览器) 运行 **Raspbian Jessie**, 则可能会从浏览器收到警告。您可以通过点击 **Advanced -> Proceed to localhost(unsafe)** 来解决。3. 您将被带到使用亚马逊的登录页, 输入您的 **Amazon** 凭证。4. 您将被带到 **Dev Authorization** 页面, 确认您希望您的设备能访问之前创建的安全性配置文件。点击 “**Okay**” 5. 现在将重定向到 `https://localhost:3000/authresponse?code=xxx` 的网页里。网页的正文将会: `device tokens ready`



6. 返回到 Java 应用程序，然后单击“OK”按钮。客户端现在可以接受 Alexa 的请求。

终端窗口 3

注意：跳过此步骤运行相同的应用程序，就不会有唤醒字引擎。

该项目支持两个第三方唤醒文字引擎：**Sensory** 的 **TrulyHandsFree** 和 **KITT.AI** 的 **Snowboy**。脚本中的 `-e` 参数用于选择代理，及支持两个值：`kitt_ai` 和 `sensory`。

打开一个新的终端窗口，并使用以下命令从 **Sensory** 或 **KITT.AI** 中引出一个唤醒字引擎。

唤醒文字引擎将允许您使用短语“**Alexa**”启动互动。

要使用 **Sensory** 唤醒字引擎，请键入：

```
1 cd ~/Desktop/alexa-avs-sample-app/samples
2 cd wakeWordAgent/src && ./wakeWordAgent -e sensory
```

输入下面的内容，则使用 **KITT.AI** 的唤醒字引擎：

```
1 cd ~/Desktop/alexa-avs-sample-app/samples
2 cd wakeWordAgent/src && ./wakeWordAgent -e kitt_ai
```

现在你有免费的 AVS 原型！

以下资源，能帮助你了解两个唤醒字引擎的更多信息：

- [Sensory](#)
- [KITT.AI](#)

步骤 8: 与 Alexa 交谈

您现在可以通过简单的使用“**Alexa**”来唤醒 Alexa。比如：

- 说“**Alexa**”，然后等待哔哔声。然后说“**what’s the time?**”

- 说“Alexa”，然后等待哔哔声。然后说“what’s the weather in Seattle?”

如果你愿意，你也可以点击“Listen”按钮，而不是使用唤醒字。点击“Listen”按钮，等待音频提示开始说话。听到音频提示可能需要一两秒钟。

步骤 9: 如何注销示例应用程序

参见: [Sample App Log Out Instructions](#)

原文链接: <https://github.com/alexa/alexa-avs-sample-app/wiki/Raspberry-Pi>

原文链接: <https://www.wandianshenme.com/play/use-snowboy-trulyhandsfree-raspberry-pi-avs-bu>