

# 物联网智能音箱 **Amazon Echo** 打造 智能家居

玩

September 8, 2017

## 目录

步骤一：搭建你的服务器 .....	3
步骤 2：控制 <b>EVERYTHING!</b> .....	4
步骤 3：使用 <b>MQTT</b> 来控制设备 .....	5
步骤 4：使用多个 <b>Echo</b> 和处理 <b>Echo</b> .....	5
知道 <b>Echo</b> 在哪里 .....	5
来自其他 <b>Echo</b> 的 <b>Echoes</b> .....	6
步骤 5：小结 .....	6

玩点什么: <https://www.wandianshenme.com>

原文链接：<https://www.wandianshenme.com/play/amazon-echo-build-home-automation-with-wem>

上个月，亚马逊发布了 **Amazon Echo** - 一个能够强大的语音识别的小型汽缸。

**Amzon Echo** 本身是封闭源码的。但是，一旦你找出了 **Echo** 的秘密，它将成为一个非常强大的语音控制工具——使您能够通过声音来控制家庭或公寓中的任何内容，而无需担心人声中的复杂性。（见：[如何做声音控制 pre-Echo](#) - 提示：难度理大）

在本教程中，我将介绍如何使用 **Amazon Echo** 通过 **WiFi** 或 **MQTT** 来控制任何设备。

材料清单：

- **Amazon Echo**
- **Raspberry Pi**
- 初始代码：[GitHub](#)

难度提示：您需要了解一些 **Python** 编程来了解这些示例。所需时间：**Alexa** 开箱后 **10** 到 **30** 分钟。注意：此教程仅在 **Linux** 系统上尝试过。如果你有兴趣，可以想在其他平台上尝试。

## 步骤一：搭建你的服务器

在你的 **Amazon Echo** 运行起来的这段时间里，让我们谈谈黑客和家庭自动化。

具体来说，我们即将，创建一个能假装成为 **WeMo** 设备的服务器。

**Echo** 集成了 **WeMo** 家庭自动化协议 - 只需说“**Alexa, discover my devices**”，它将在本地网络上搜索 **WeMo** 设备。所以我们给它一个设备，**Echo** 就能发现它！

1. 安装要求的软件：**Python 2.7**，**pip** 和必需的 **Python** 模块

```
1 sudo apt-get install python-pip ; sudo pip install requests
```

1. 下载项目的源码

```
1 wget "https://github.com/toddmcdema/echo/archive/master.zip"
```

1. 解压源码文件：

```
1 unzip master.zip -d echo
```

1. 移动到新建的文件夹中

```
1 cd echo/echo-master
```

5. 启动服务器 - 您应该能看到一些调试文本, 上面写着服务器已经开始轮询。

```
1 python example-minimal.py
```

1. 对你的回声 “**Alexa, discover my devices**”。她会花几秒钟找到新设备。如果她第一次找不到它, 请尝试重新启动并重新启动该过程, 再次发现设备。
2. 现在试试, 说: “**Alexa: turn off device**”, 或 “**Alexa, device on**”。您将看到 **True** 或 **False** 用于开/关视频, 以及 **Echo** 的 IP 地址。

脚注:

**Echo** 还可以通过 **Amazon Cloud** 发送命令——使用他们的新服务 **Lambda**。对于 **Amazon Echo** 新手来说, 这是种方便的方式。但是对于智能家居来说, 这不友好。它要求每个命令都包括程序的名称, 即 “**Alexa, ask HOME to turn on the lights**”, 它还涉及到远程服务器的网络往返, 从而减慢响应时间。

我们真正想要的是在本地控制本地设备, 没有前缀 - 即 “**Alexa, turn on the lights**” - 这就是为什么我使用 **WeMo** 协议。

## 步骤 2: 控制 **EVERYTHING!**

这就是它的一切。您现在有一台服务器连接到您的 **Amazon Echo**。您现在可以使用您的声音:

- 控制灯, 空调, 甚至你的电视!
- 在您的计算机上运行脚本
- 切换整个家庭自动化模式 (“电影模式”, “派对模式” 等)
- 恶心你的室友
- 通过更轻松地关闭您未使用的设备来节省电力

我们设置了我们的公寓, 在每个房间里运行一个 **Amazon Echo**, 通过友好的 **MQTT** 协议控制所有的家庭自动化设备。

请继续阅读, 以了解如何用 **MQTT** 以及多个 **Echo** 来控制设备。

### 步骤 3: 使用 **MQTT** 来控制设备

**MQTT** 使用“发布商-订阅者”或“pub-sub”框架，来使控制智能家居中的设备变得轻而易举。使用 **pub-sub**，您可以将消息发布到主题（**topic**），任何监听这些主题的设备都会收到这些消息。所以，例如，你可以有一个主题（**topic**）卧室灯，并发布“1”或“0”打开或关闭灯。

以下是使用我将 **Amazon Echo** 连接到 **MQTT** 的方式：

1. 使用 **pip** 来安装 **MQTT**，通过打开命令窗口并键入：`pip install paho-mqtt`。
2. 您现在可以运行 **MQTT** 测试文件进行测试：`python fauxmo_mqtt_example.py`
3. 打开 **Web** 浏览器并浏览到 [HiveMQ MQTT Websocket 客户端](#)
4. 点击“Connect”
5. 点击“Add New Topic Subscription”。在“testtopic/#”的地方输入“livingroom”，并点击 **Subscribe**
6. 对着你的 **Amaon Echo** 说：“Alexa, turn on the lights”
7. 我应该能在主题里看到一个“True”的消息

我会把它作为一个练习，来发现示例代码使用的其他主题（**topic**）。

免责声明：示例代码中使用公共 **MQTT** 服务器，是为了可能快地开始运行。所以，不要将您的社交安全号码、手机号等等发布到测试主题！

### 步骤 4: 使用多个 **Echo** 和处理 **Echo**

如果你爱上了新的 **Echo** 的能力，你可能会对使用多个 **Echo** 感兴趣。你会遇到两个问题：以下是这两个问题的解决方案：

知道 **Echo** 在哪里

你不想让卧室的 **Echo** 打开厨房的灯光，但这些东西并不完全是内置的 **GPS**，而是可以使用每个 **Echo** 的 **IP** 地址来提供不同的命令及控件。

查看 `fauxmo_mqtt_example.py` 文件，看至一个解决方案。我记录了当我运行 `fauxmo_minimal.py` 时看到的 **IP** 地址，并将其保持为常量。然后，当我想要不同的 **Echo** 对同一个命令有不同的响应时，我将在处理程序的 `act ()` 函数中的 **IP** 与 `client_address` 进行比较。

## 来自其他 **Echo** 的 **Echoes**

**Echo** 的麦克风非常敏感，他们经常会从一个房间里响应你，即使是在播放音乐！虽然是令人难以置信的方便，但当您在房屋周围有多个 **Echos** 时，也会成为一个问题。

为了解决这个问题，我在代码中添加了一个“去抖动”功能（请参阅 `debounce_handler.py`），以防止相同的命令被快速连续调用多次。

所以，如果卧室和厨房有多个 **Echo** 听到你，只有第一个提交命令将执行。无论哪个 **Echo** 听起来你最响亮、最接近你、处理消息更快、提交网络请求更快、并处理其请求。因而，所有的 **Echo** 听到更安静的 **Echoses**，则需要更长的时间来处理和去抖动。瞧！去抖动功能是内置的，所以实际上没有什么额外的你需要做这里 - 它开箱即用。

## 步骤 5: 小结

您现在有能力能用声音控制您家庭中任何设备。去做不可思议的事情！

原文链接: <http://www.instructables.com/id/Hacking-the-Amazon-Echo/>

原文链接: <https://www.wandianshenme.com/play/amazon-echo-build-home-automation-with-wem>