

智能电源板

ESP8266 长期以来一直是物联网项目的首选工具，因为它价格便宜且支持不断增加。该项目包含两种创建互联网连接电源板的方法，该电源板与 Amazon Alexa / echo 设备连接。

在深入研究代码之前，请确保已安装必要的库：

Arduino ESP8266 电路板管理器：文件>首选项>其他电路板管理器 URL：

http://arduino.esp8266.com/stable/package_esp8266com_index.json

安装 ESP8266 库：草图>包含库>管理库...> ESP8266

智能家居方法（简易）



ESP Wemo 设备仿真图

此方法使用 FauxMo 库在 ESP8266 上模拟多个 WeMo 设备。

优点：

- 易于设置。编辑 Wifi 凭据，更新设备名称和闪存到 esp 板上。
- 简单的设备控制话语：'Alexa，打开/关闭{设备名称}'

缺点：

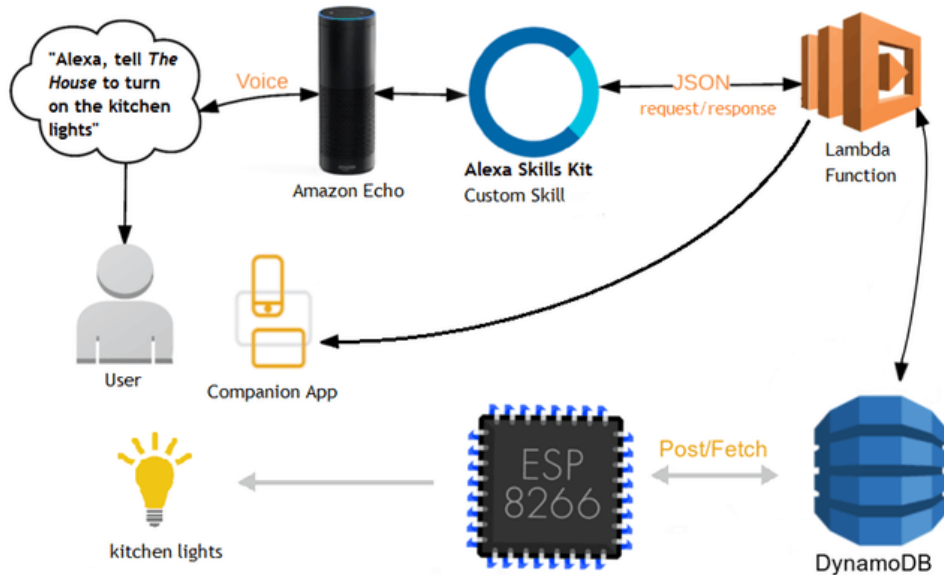
- 控制有限。只能切换开/关。
- 回复有限。Alexa 总是回答'ok'。

- 设备更新可能会失去支持
- 仅受控制 UDP 的设备支持（谷歌主页不起作用）

说明：

- 打开 smartPowerStrip \ code \ Smart Home \ multiFauxMo.ino 并编辑相关参数; ssid, 密码, 密码。
- 在 arduino IDE 和 flash 上选择正确的板。你现在准备摇滚了。您可以在 Alexa 应用程序中重命名开关。

Alexa 技能套件+ Lambda 函数+ DynamoDB（中级）



智能电源板的流程图

使用此方法，ESP 设备轮询 DynamoDB 以获取值，解析数据并处理必要的硬件外围设备。代码还能够更新值（或写入全新的条目）以处理用户想

要通过硬件更新值的情况。使用这种方法使我们能够将超过二进制状态传递给 ESP 设备，为各种应用程序打开了大门。

优点：

- 无限制的控制和定制。可以从 DynamoDB 读取多个值和数据类型。
- 不需要像 MQTT 代理那样的中间件。ESP 设备直接与云通信。
- 可以从 Alexa 获得定制的，精心的回复。

缺点：

- 更难设置。需要创建 DynamoDB，安装 ArduinoAWS 库和设置 Lambda 函数。
- 更长的话语：'Alexa，告诉{技能名称} {intent}'

常见问题：

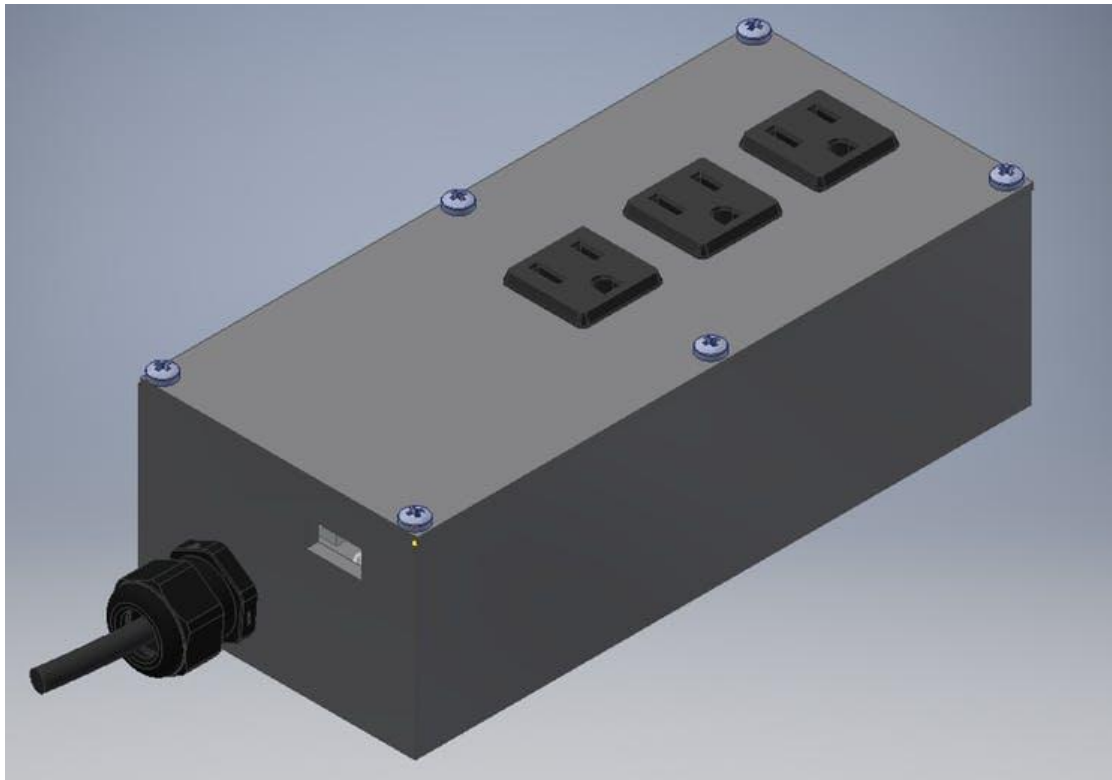
- DynamoDB 中的值不是字符串。检查值是否不是数字格式。

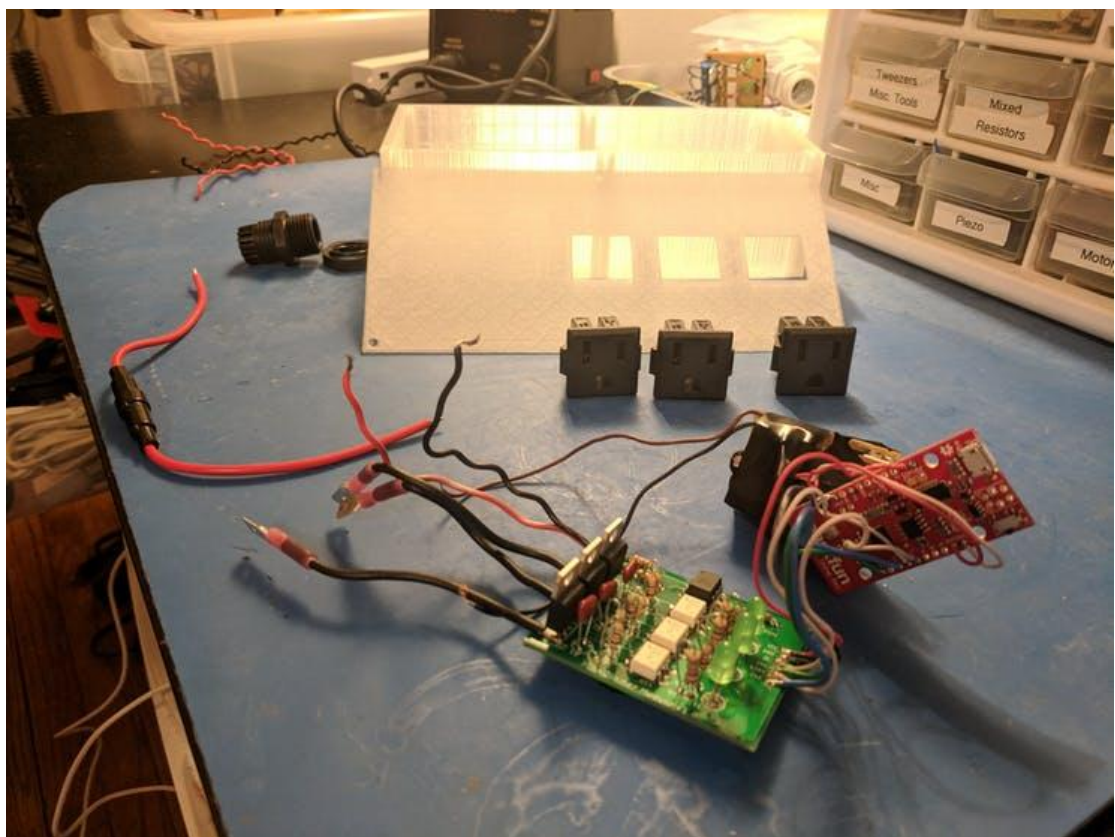
可调光，互联网控制的电源板（中级/高级）

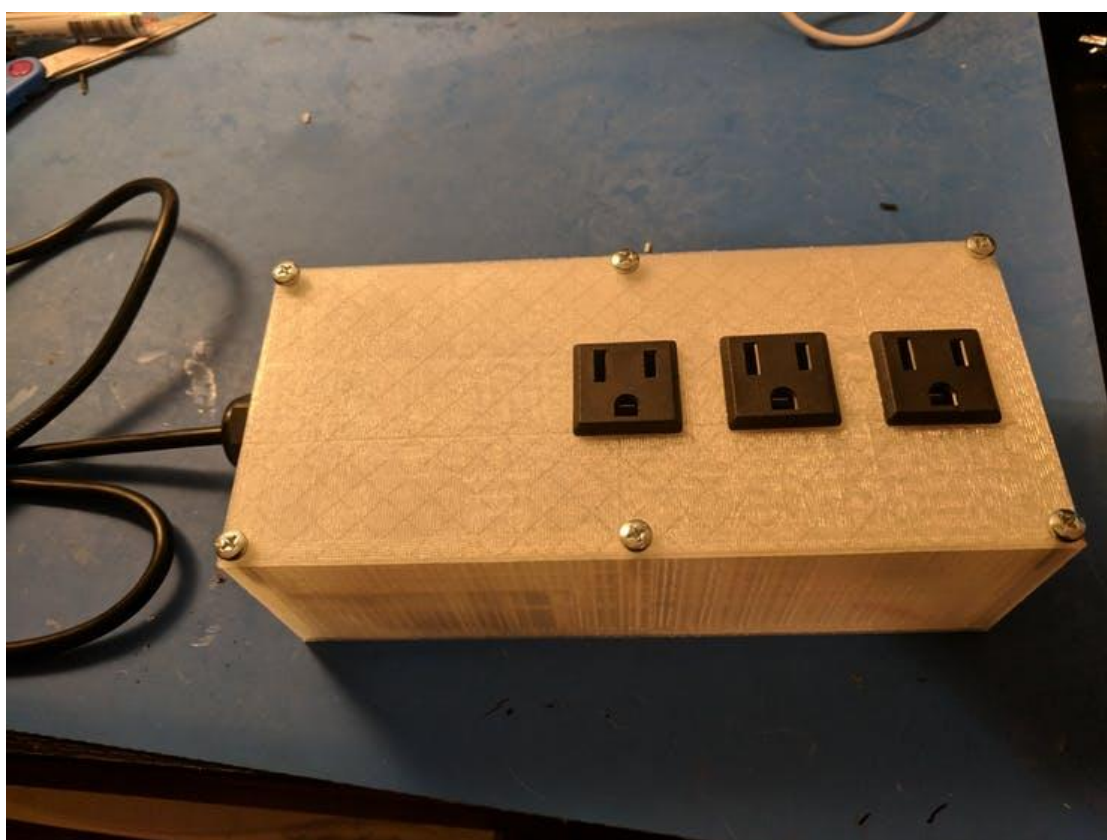
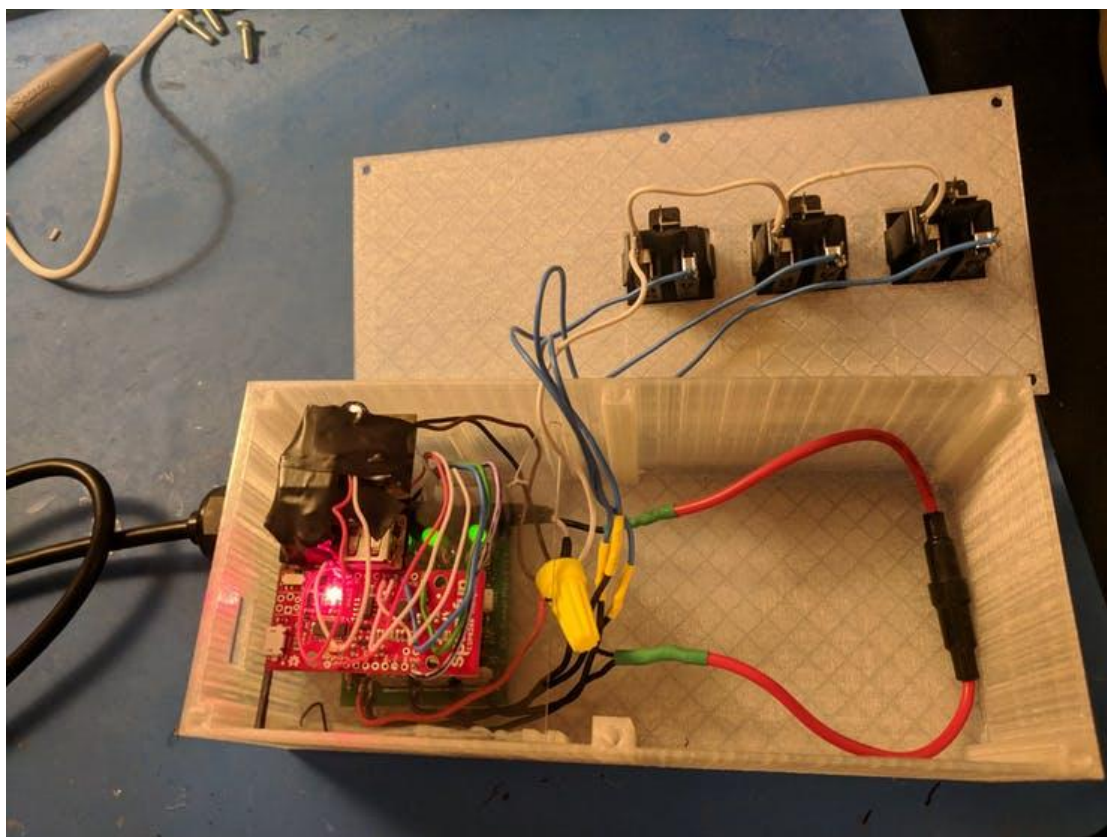
警告：如果您不方便处理 120VAC，请不要尝试此版本。连接到电源板的某些负载在变暗时可能会永久性损坏，系统使用白炽灯泡和加热器进行测试。

1.硬件：

外壳设计有搁架能力，可将高压元件与低压元件分开。将电源线放置在面的底角以允许安装应变消除并且添加槽以允许重新编程 ESP 芯片。在印刷之后，切割丙烯酸片材，在侧面上具有凹口以用作搁板支撑件并允许电线在水平面之间延伸。组装电子器件，将插座插入适当的槽中，然后焊接到来自调光模块的电源线。然后使用六个 M4 螺钉将盖子连接到外壳上。





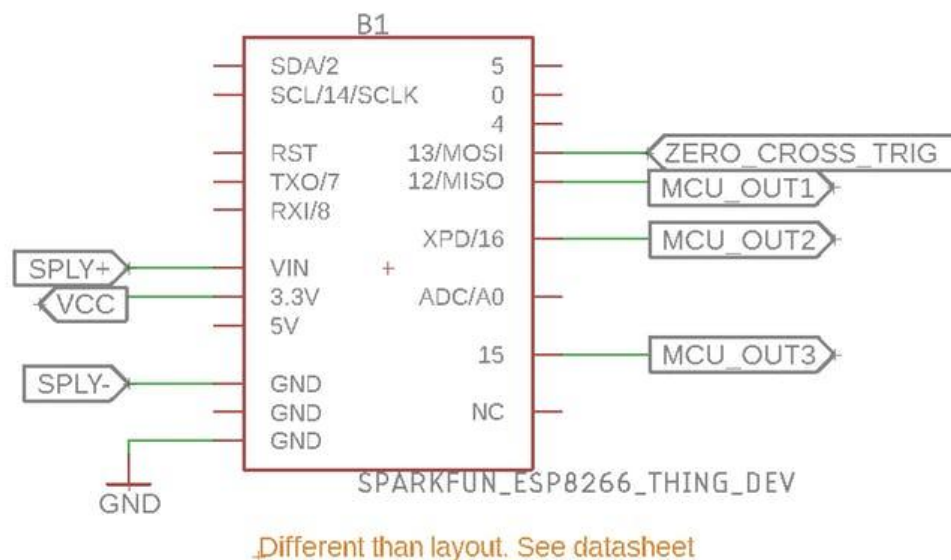


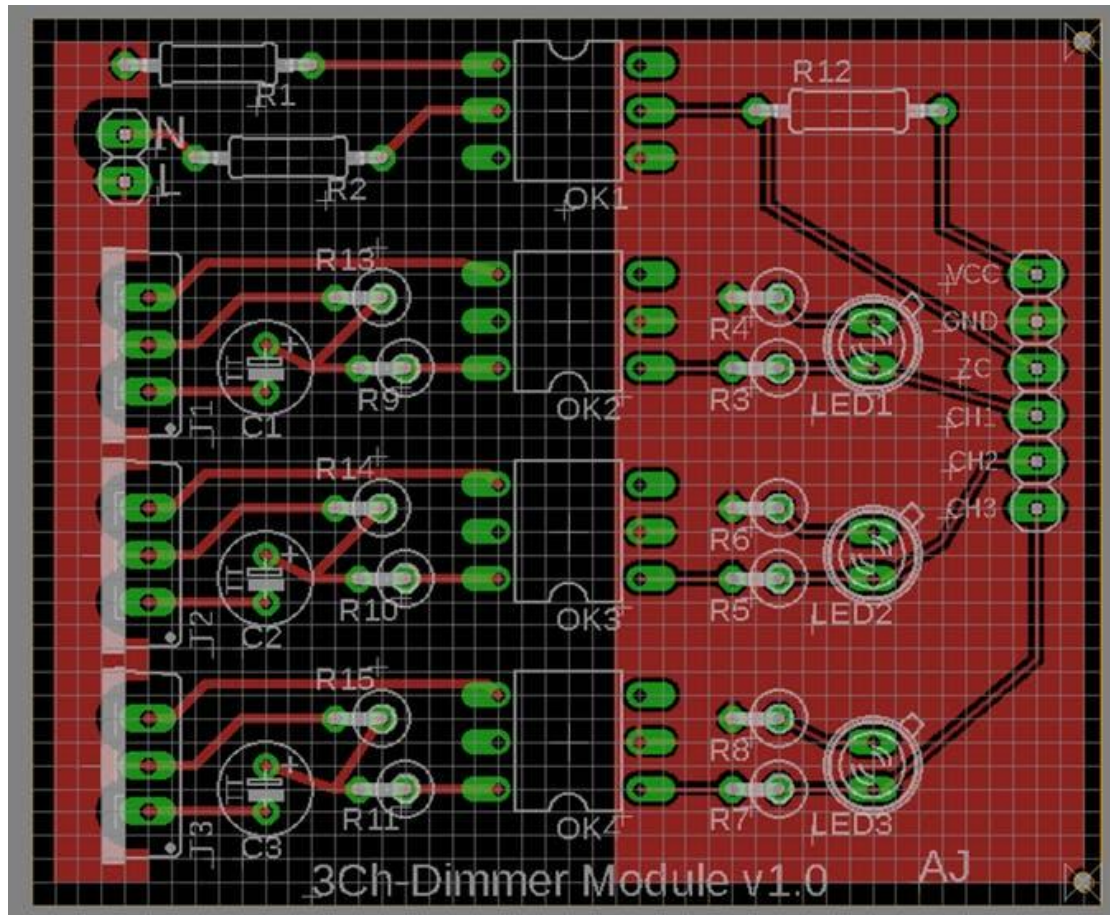
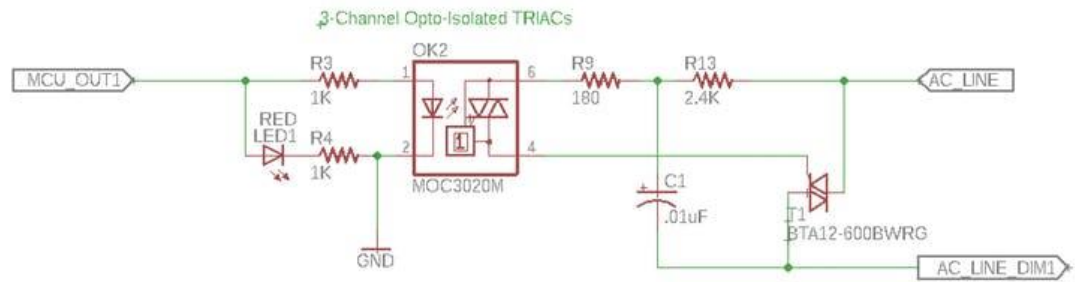
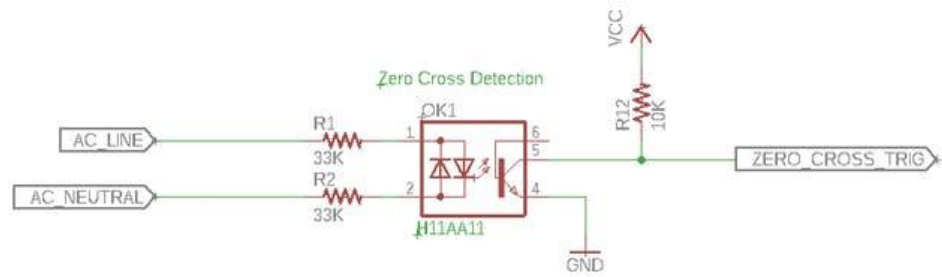
2.电子产品:

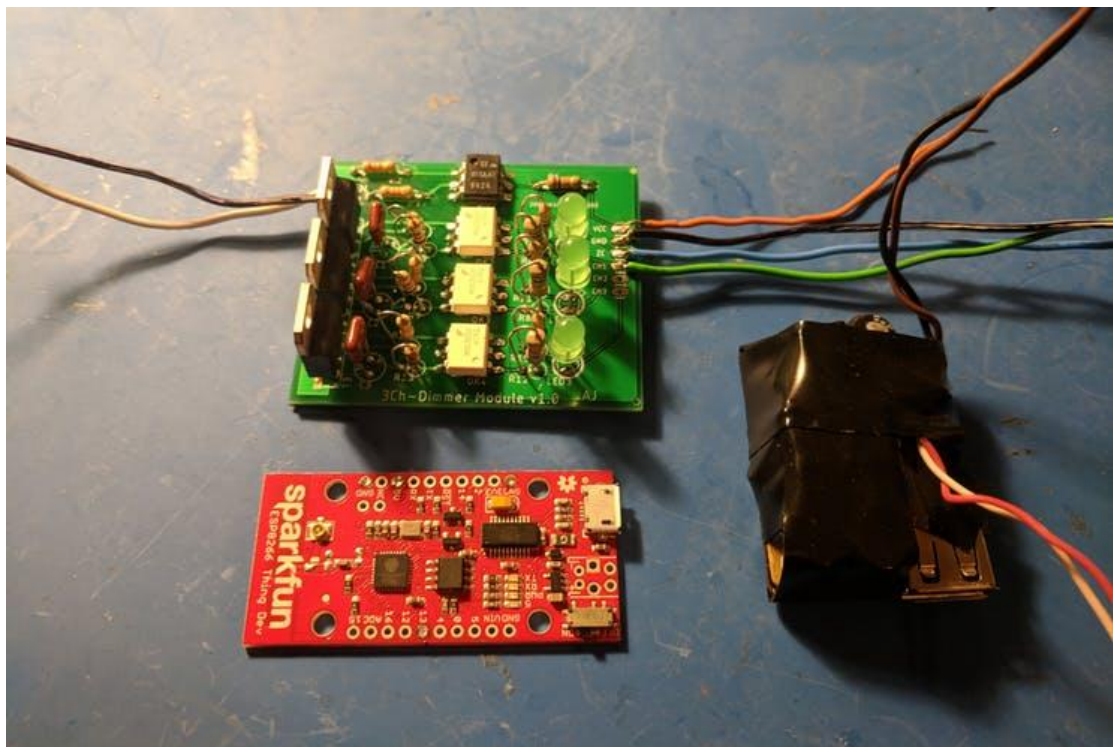
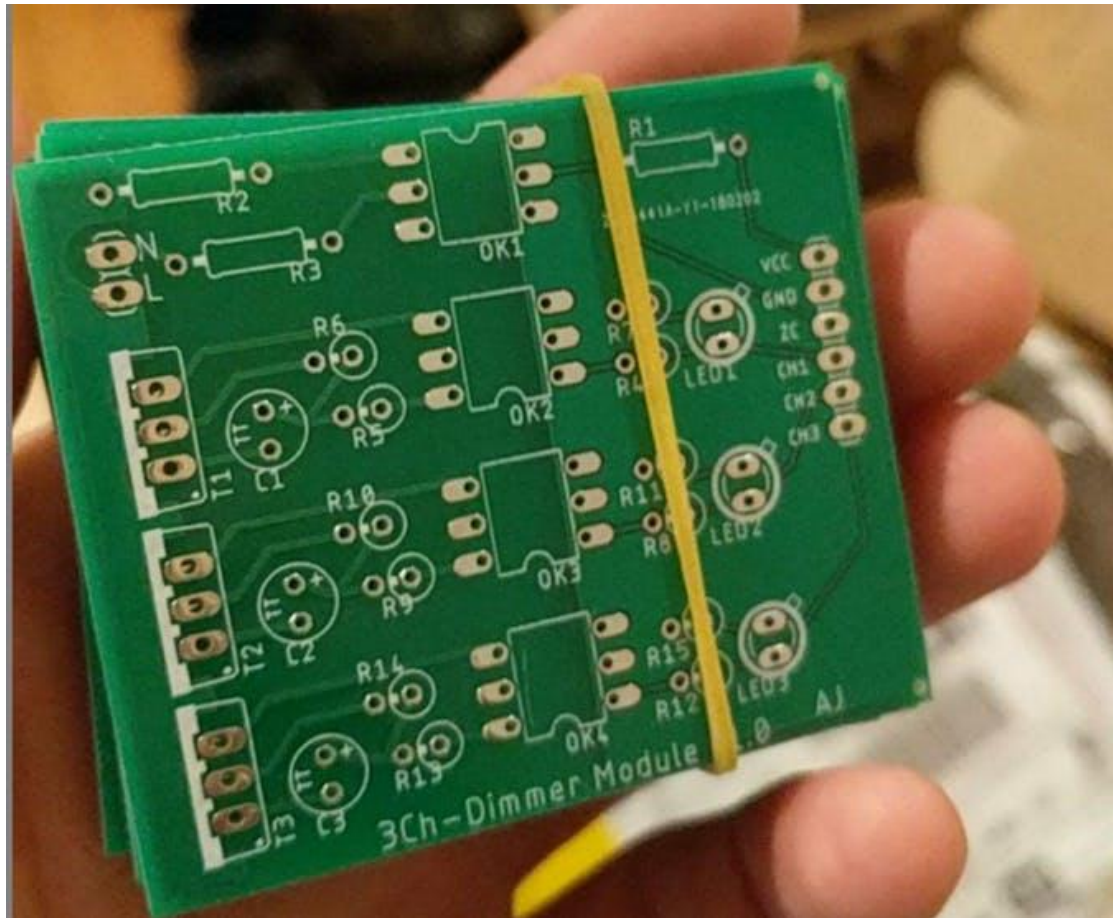
为了使用 MCU 进行交流调光，我们需要一个能够处理交流电压的电路和另一个电路，以便在交流波超过零时通知控制器。有关理论视图的详细信息，请参阅有用链接部分。

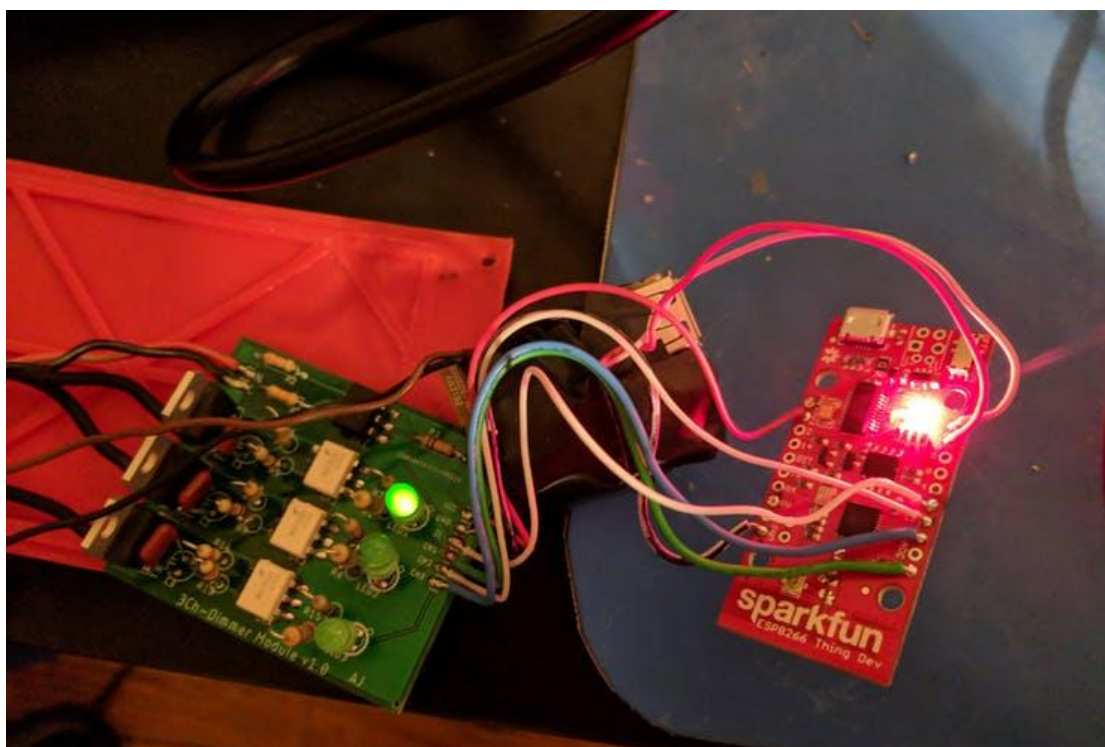
设计的电路检测交流波何时越过零点，并在光学隔离时向 ESP 设备发送信号。然后 ESP 设备计算已经过了多少毫秒并在适当的时间切换 GPIO。GPIO 输出到 MOC3020M，可将 MCU 器件与高压光电隔离。该信号关闭 TRIAC 的栅极，让电压流过，打开设备。PCB 原理图已创建，布局并发送给制造商，以最大限度地减少电子设备的占地面积。

PCB 组装并连接到 ESP8266 Thing Dev 和旧的 iPhone 充电器的 AC-DC 转换器。AC-DC 转换器连接到标准的 120VAC 电缆并为电子设备供电。





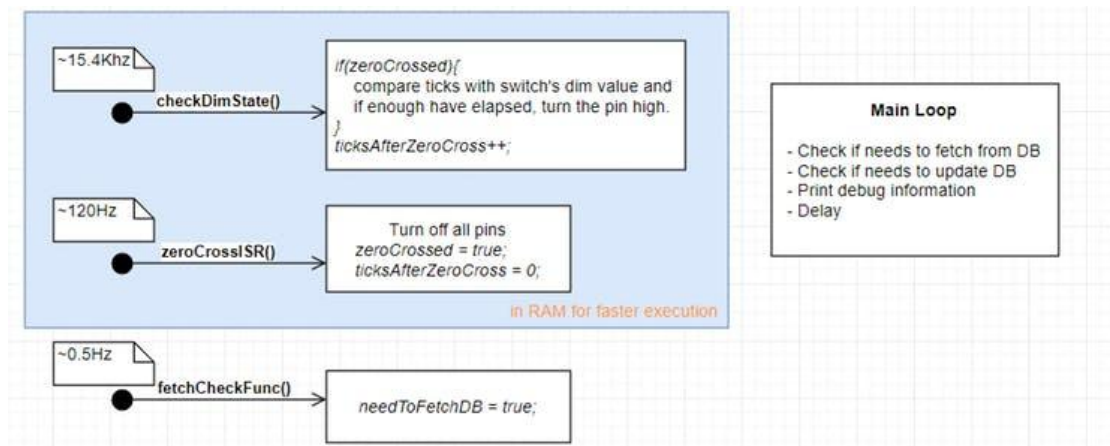




1/6 • ESP8266 引脚图

3.嵌入式代码

编写该程序的主要障碍是实时控制部分。ESP8266 不是专为实时控制而设计的，所以我们必须做一些杂技才能让它进行调光。ESP 的内部时钟运行在 80Mhz（默认情况下），我们可以关掉这个时钟来运行调光器中断。因为这个中断每隔几微秒就会运行一次，所以它必须很快。因此，我们必须将 ICACHE_RAM_ATTR 指令添加到我们的中断函数中，以将其置于 RAM 而不是 FLASH 中； 不是一个很好的文件指令。这个中断的准确性似乎是 $\pm 3\mu s$ ，这个应用程序的误差可以忽略不计。下面是代码结构的图表。



4. DynamoDB:

创建一个 DynamoDB 表, 其中 Partition 名称为 String, Sort 键为 Number, 然后为该表创建一个新项目。要遵循此 repo, 请使用 `devName(String): ESP8266, devID(Number): 1, onStatus(String): 1, 以及 brightLvl(String): 100`。

Create DynamoDB table

Tutorial ?

DynamoDB is a schema-less database that only requires a table name and primary key. The table's primary key is made up of one or two attributes that uniquely identify items, partition the data, and sort data within each partition.

Table name* ⓘ

Primary key* Partition key

String ⓘ

☒ Add sort key

Number ⓘ

导航到 *AWS 站点 > IAM* 并创建一个对 DynamoDB 具有完全访问权限的用户。这将用于为 Lambda 函数提供 ESP 设备的读写权限。现在我们可以处理嵌入式代码了

下载 ArduinoAWS 库并将其复制到 arduino 安装目录。在 Windows 机器 C:\ Program Files (x86) \ Arduino \ libraries 中。有关详细信息，请转到[此处](#)

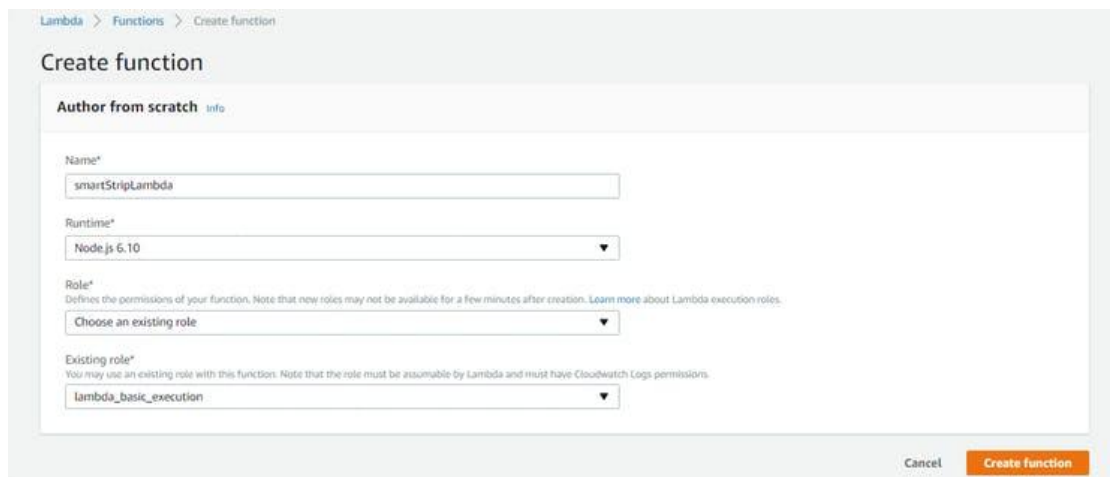
打开 smartPowerStrip \ code \ AlexaSkillsKit \ Embedded Code \ Esp8266DynamoDB 目录，并复制与在 IAM 上创建的用户对应的密钥。

编辑相关的代码； AWS_REGION, TABLE_NAME, pSSID 等，并刷新您的设备。如果您有兴趣了解代码的工作原理，可以转到嵌入代码部分。

ESP 设备现在应该能够获取并推送到云端！让我们添加 Alexa

5. Lambda 函数：

Lambda 函数将成为 DynamoDB 和 Amazon Echo 设备之间的中间人。导航到 AWS 站点 > Lambda > 功能 > 创建功能。为其分配先前创建的 IAM 角色



The screenshot shows the AWS Lambda 'Create function' page. The breadcrumb trail at the top is 'Lambda > Functions > Create function'. The main heading is 'Create function'. Below it, the 'Author from scratch' tab is active. The form contains the following fields:

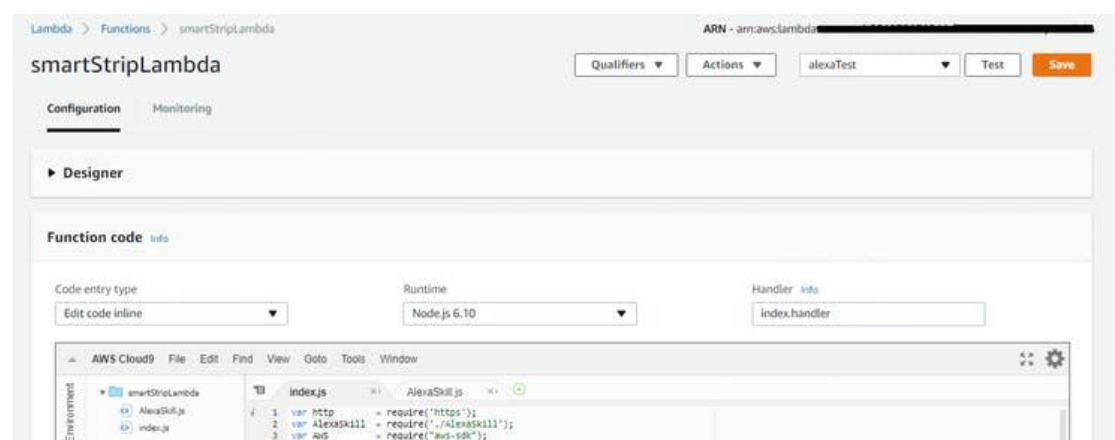
- Name***: A text input field containing 'smartStripLambda'.
- Runtime***: A dropdown menu with 'Node.js 6.10' selected.
- Role***: A dropdown menu with 'Choose an existing role' selected. A note below it says: 'Defines the permissions of your function. Note that new roles may not be available for a few minutes after creation. [Learn more](#) about Lambda execution roles.'
- Existing role***: A dropdown menu with 'lambda_basic_execution' selected. A note below it says: 'You may use an existing role with this function. Note that the role must be assumable by Lambda and must have Cloudwatch Logs permissions.'

At the bottom right, there are two buttons: 'Cancel' and 'Create function'.

打开 AlexaSkillsKit \ LambdaFunc \ index.js 并编辑任何相关参数； tableName, partitionKey 等等。完成后，将 index.js 和 AlexaSkill.js 复制到功能代码部分。

通过将 `testEvent.js` 复制到 Lambda 函数的测试事件部分来配置测试事件。默认情况下,测试事件模拟 Amazon Echo 设备被告知将灯泡变为 50% 亮度。

从下拉菜单中,选择创建的测试事件,保存代码并点击测试按钮。数据库将使用测试事件中的值进行更新。您可以查看执行结果以获取更多详细信息。



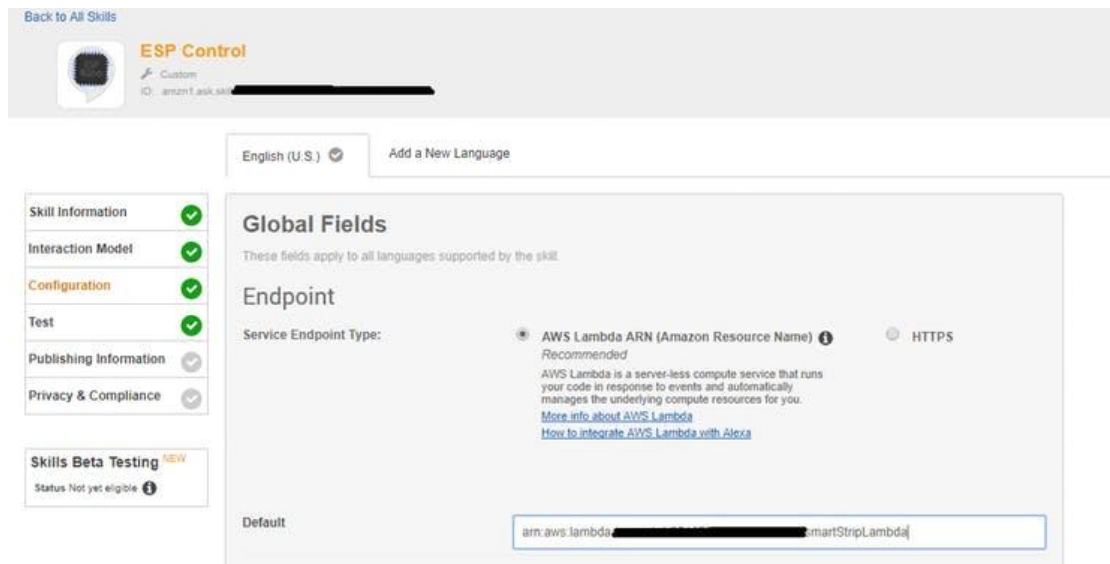
6. Alexa 技能套件:

现在我们可以添加 Alexa Skills Kit 作为 Lambda 函数的触发器。可以取消选中“禁用技能 ID”复选框。

在亚马逊开发者网站上,创建一个 Alexa 技能套件技能。我建议让调用名称容易记住,比如“众议院”。

根据 `smartPowerStrip | code | AlexaSkillsKit | speechAssets` 中的文件配置交互模型。请记住还要添加 `IntentSchema.json` 底部的自定义槽类型和文本文件中提供的 Sample Utterances。

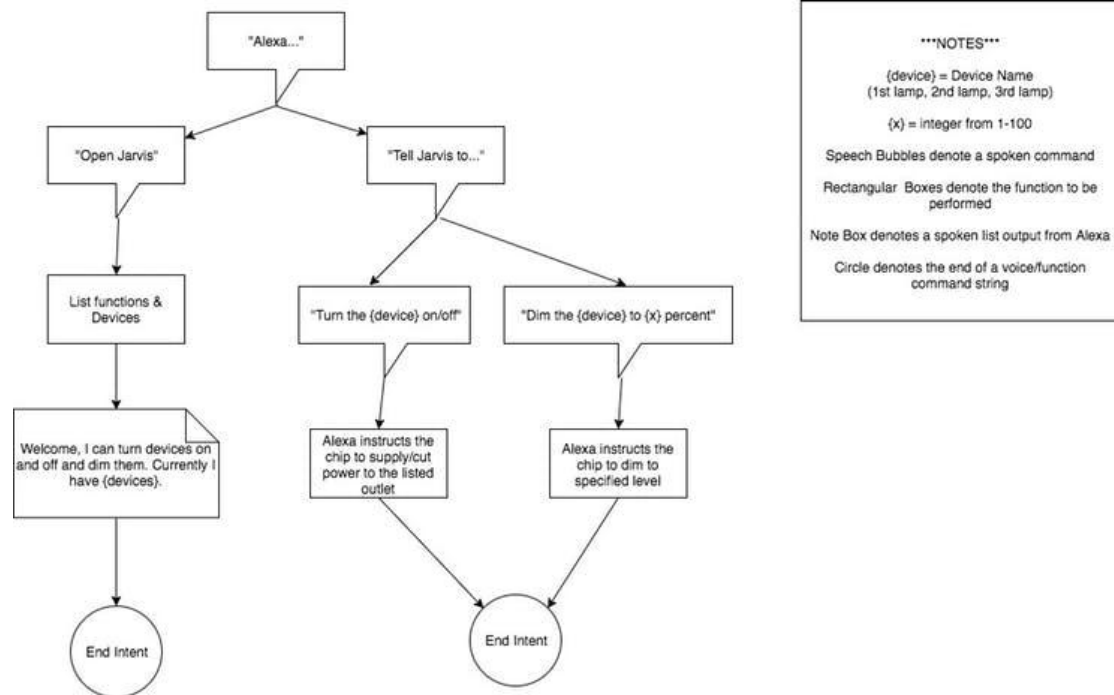
在 Configuration 选项卡上，选择 AWS Lambda ARN 作为端点并粘贴 Lambda 函数的 ARN。这会将您的自定义技能与创建的 Lambda 函数联系起来。



通过在服务模拟器中输入话语来测试功能，“将灯调暗至 50%”。
DynamoDB 将更新，ESP 设备将获取最新信息。现在你已经制作了自己的调光电源板！

7. 语音用户界面

此图详细说明了构建 Smart Strip 的不同命令和命令树。最终用户可以按照此图表激活项目所需的功能。



8.演示:

演示视频: <https://youtu.be/kUzPrfj1B4>

智能条带帮助菜单: <https://youtu.be/9Ed4aDO6zRE>

原文地址: <https://www.hackster.io/68994/smart-power-strip-5c1d20>