NodeMCU 与 Micropython 打造门铃记录器

Phodal Huang

October 24, 2017

目录

原文链接：https://www.wandianshenme.com/play/esp8266-micropython-nodemcu-build-doorbell-l

这是一个由我的好奇心开始的小项目。有些时候，即使我们在家，我们的包裹也会交给邻居。因此，我想跟踪一下，邮递员是否真的使用我们的门铃？

在 Make 杂志（IoT Special）中，我发现了一篇“关于使用 Raspberry Pi 和声音传感器构建门铃闹钟”的文章。它给了我必要的驱动力，来实现我自己的项目。我发现在这样一个简单的项目上，使用一个 Raspberry Pi 有些过分。即使 Raspberry Pi 的零售价约为 12 欧元，它仍然需要一个 WiFi 适配器。这就是为什么我决定使用 NodeMCU 来实现。您可以在 Ebay 上用 6 欧元买到，并且内置 WiFi。此外，您还可以运行 Micropython，并且我想在短时间内尝试使用微控制器上的 Python 实现。

时间：1 小时所需要的技能：Python 语言的基本知识，Flask 和命令行费用：~15 欧元

步骤 1：硬件材料

我很惊讶要完成这个任务，只需要这么几个组件。

- 1 x NodeMCU IoT 平台 6 欧元
- 1 x 带数字输出的声音传感器 5 欧元
- 1 x USB 电缆
- 1 x USB 电源
- 3 x 连接线

您需要将声音传感器的输出连接到 D4 的 16 引脚。再将电源连接到 NodeMCU 上的对应引脚上。再用电位计更改声音传感器的灵敏度。

步骤 2：设置 NodeMCU

有一个非常全面的教程，用于在 NodeMCU 上安装 MicroPython（芯片为 ESP8266）。你可以简单地跟随这个教程。为此，您需要首先为 NodeMCU 串行芯片安装驱动程序。

在 NodeMCU 上安装串行芯片的驱动程序

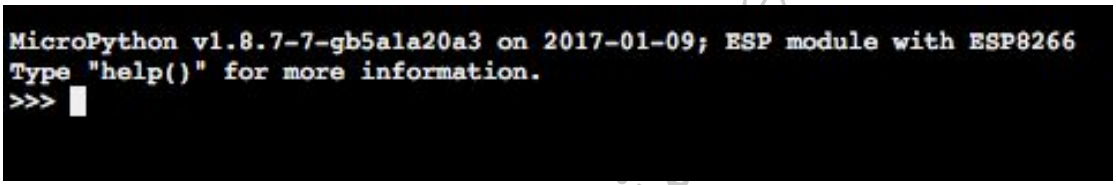
NodeMCU 上使用的串行/USB 芯片是 CH340G，一种中国产的低成本芯片，取代了 Arduino 板上使用的 FTDI 芯片。您将需要为您的操作系统安装配件驱动程序。我使用的是这个网站（CH340G Mac OS）上所给的驱动程序。你只需要按照说明操作即可。

安装 **Micropython**

如上所说，我们只需要按照这个指南：“Getting started with MicroPython on the ESP8266” 即可。

通过串行 **REPL** 连接到 **NodeMCU**

使用你的串行控制台连接到 NodeMCU。你可以从这篇 MicroPython 相关的文章《“Getting a MicroPython REPL prompt”》中了解到更多内容。最后，你会看到如下图所示的命令提示符。



现在您可以与 NodeMCU 进行交互。

下接来，我们要先连接到我们的 WiFi 网络。我们可以通过键入以下命令，来执行此操作。

```
>>> import network
>>> sta_if = network.WLAN(network.STA_IF)
>>> sta_if.active(True)
>>> sta_if.connect('<your ESSID>', '<your password>')
>>> sta_if.ifconfig()
('192.168.0.2', '255.255.255.0', '192.168.0.1', '8.8.8.8')
```

它将激活与您的网络的 WiFi 连接。用您的 SSHID 和密码替换上面的 ‘SSHID’ 和 ‘PASSWORD’。有关网络连接的详细说明，请参见 MicroPython 网络基础知识。

通过 **WiFi REPL** 建立连接

现在我们已经启动了 WiFi 支持，我们可以关闭串行控制台，并使用 Web 控制台与我们的 NodeMCU 通信并发送程序文件。在 http://micropython.org/webrepl/ 上打开 WebREPL，插入设备的 IP 并按下连接（connect）。现在您应该看到与串行控制台相同的提示。

现在我们准备好了，可以编写门铃记录器的程序了。

步骤 3：编写记录器客户端代码

我们的门铃记录器客户端的程序相当简单。将以下代码保存在名为 main.py 的文件中。

```
import machine
import time
from urequests import post

HOST_URI = 'http://yourserver/bell/api/add'
bell = machine.Pin(16, machine.Pin.IN)

def send():
    resp = post(HOST_URI)
    if resp.text == 'ok':
        led.high()
    time.sleep(10)
    led.low()

time.sleep(5)
while True:
    if bell.value() == 0:
       send()
```

这些代码实际上是，等待连接到声音传感器的 PIN16 上的一个信号。如果发生这种情况，NodeMCU 会向 HOST_URI 定义的 HOST 发送一个 POST 请求。在主机上运行一个小型 Flask 服务器，它会将事件记录在一个小型数据库中。

步骤 4：编写记录器服务端代码

对于编程日志服务器，我们将使用 Flask。Flask 是使用 Python 编写的微框架。服务器的代码可以在 Github 上找到：https://github.com/MrLeeh/bellwatcher 。克隆到本地计算机上。

现在，我们将使用以下命令创建并激活一个新的 Python 虚拟环境：

```
$ virtualenv venv
...
$ source venv/bin/activate
```

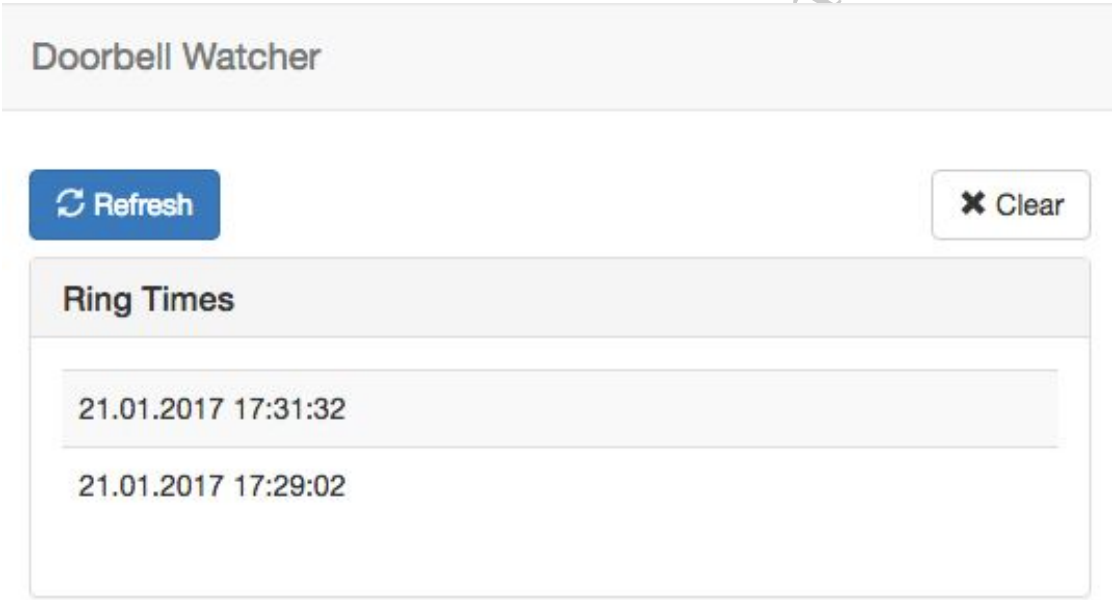
要安装所有必需的软件包，我们将使用 pip 软件包管理器。

```
$ pip install -r requirements.txt
```

现在，我们只需要通过调用启动服务器命令：

```
$ python app.py
Running on http://localhost:8081/ (Press CTRL+C to quit)
```

打开浏览器，并访问 http://localhost:8081/bell/ ，你应该看到如下的屏幕。



现在如果事件被触发，您将在列表中找到一个新的条目。您可以根据你的需要刷新，并删除列表。这实际上是一个非常简单的界面，你可以随心所欲地按你的习惯修改。

步骤 5：结论

所以就这样，该项目很小，功能还不是非常丰富。但现在我可以跟踪邮递员，并调查他是否真的使用了门铃。我正在考虑的一个增强功能是增加一个算法，将门铃与正常噪声分开（如，用力关闭门的声音）。

原文链接:http://www.instructables.com/id/Doorbell-Logger-With-NodeMCU-and-Micropython/

原文链接:https://www.wandianshenme.com/play/esp8266-micropython-nodemcu-build-doorbell-l