

在 **Pycom** 使用 **Python+
Micropython + MQTT** 进行物联网
编程

Phodal Huang

September 8, 2017

目录

步骤 1: 准备	3
什么是 MQTT	3
你需要什么?	3
步骤 2: 开始编程	3
Main.py	3
步骤 3: 设置 Adafruit IO	5
步骤 4: Finished!	5

玩点什么: <https://www.wandianshenme.com>

原文链接:<https://www.wandianshenme.com/play/mqtt-pycom-micropython-build-iot-with-python>

本教程将引导您使用 MicroPython 通过 MQTT 通过 subscribe/publish (发布/订阅) 方法进行通信。

这是使用 WiPy、LoPy、SiPy、或其他 MicroPython 板开始发送和接收数据的简单方法。我们将使用 io.adafruit.com 作为 MQTT 代理来测试和显示我们的数据。

步骤 1: 准备

什么是 MQTT

MQTT 代表了 MQ Telemetry Transport。它是针对于受限设备和低带宽, 高延迟或不可靠网络设计的发布/订阅, 非常简单和轻便的消息传递协议。其设计原则, 尽量减少网络带宽和设备资源需求, 同时还要确保可靠性和一定程度的交付保证。这些原则, 还可以使协议成为连接设备新兴的“机器对机器”(M2M) 或“物联网”世界的理想选择, 以及带宽和电池电量非常高的移动应用。

你需要什么?

为了遵循本玩法, 您需要以下内容:

- 一个支持 MicroPython 的设备 (WiPy, LoPy, SiPy 等)
- Adafruit IO 账户
- MQTT 库: [PyCom MQTT](#)

步骤 2: 开始编程

假设您已经安装了 MicroPython 设备 (本教程中我们使用 WiPy 2.0) 并更新了最新的固件, 然后就能开始了。如果你不知道 Pycom, 你需要访问 docs.pycom.io 了解 Pycom 设备的更多信息。你也可以从你的制造商的网站, 了解到更多的详细信息。

连接到您的设备, 并访问其文件存储 (FTP 或 Pymakr 同步)。然后, 找到 lib 文件夹 (/flash/lib), 并复制到本教程末尾引用的 mqtt.py 中。这个是我们将会使用的 MQTT 协议库。

Main.py

现在我们可以开始编写我们的 main.py 文件, 来连接、发布和订阅 MQTT。

打开一个新的文件，或者在 `flash` 下编辑 `main.py`。我们想在这个脚本中做一系列的事情：

- 连接到 **WiFi**
- 连接到我们的 **MQTT** 代理
- 然后发布我们选择的主题的消息

我们将编写脚本并填写空白，稍后我们将在 `io.adafruit.com` 中进行配置。

```
1 from network import WLAN
2 from mqtt import MQTTClient
3 import machine
4 import time
5
6 def sub_cb(topic, msg):
7     print(msg)
8
9 wlan = WLAN(mode=WLAN.STA)
10 wlan.connect("yourwifinetwork", auth=(WLAN.WPA2, "wifipassword"),
11 timeout=5000)
12
13 while not wlan.isconnected():
14     machine.idle()
15 print("Connected to Wifi\n")
16
17 client = MQTTClient("device_id", "io.adafruit.com",user="your_username",
18 password="your_api_key", port=1883)
19 client.set_callback(sub_cb)
20 client.connect()
21 client.subscribe(topic="youraccount/feeds/lights")
22 while True:
23     print("Sending ON")
24     client.publish(topic="youraccount/feeds/lights", msg="ON")
25     time.sleep(1)
26     print("Sending OFF")
27     client.publish(topic="youraccount/feeds/lights", msg="OFF")
28
```

```
29 time.sleep(1)
```

我们刚写的 `main.py`，它将向我们的 **MQTT** 代理发布一个开/关灯的消息。虽然我们并没有将灯开，关连接到我们的设备，但我们可以模拟其行为。我们期待每秒钟会在我们的 `topic/feed` 上显示一条消息，其将在“ON”和“OFF”之间交替显示。

步骤 3: 设置 **Adafruit IO**

Adafruit IO 代理是想要尝试 **MQTT** 和订阅/发布消息的任何人的好地方！打开您的网络浏览器，并导航到 `io.adafruit.com`，注册一个帐户然后登录。

现在点击 `feed` 选项卡，并创建一个新的 `feed`。你可以命名任何你喜欢的东西，但是在本教程中，我们称之为“lights”，与我们在 `main.py` 文件中使用的是一样的。

这是我们将监听的消息的主题。我们现在需要一些东西来显示我们的传入数据。**Adafruit IO** 有一个很好的功能，称为“仪表盘 (Dashboards)”，可用于显示 `feed` 上发生的任何事情。我们将使用一个简单的开/关开关来显示 **WiPy** 的变化输入。

创建一个新的仪表盘，然后点击它的名称打开它。从这里，我们可以添加能与我们的数据交互的各种显示，切换和视图。选择一个切换块，然后选择“lights”源作为数据源。

你可以思考一下，我们已经建立的发布、订阅方式。我们的 **WiPy** 正在发布关于灯条件 (ON 或 OFF) 的数据，**Adafruit IO** 作为我们的中间件 (或代理) 和作为订阅者的仪表盘。

您现在应该使用 **Adafruit IO** 用户名和密码——在 **Adafruit IO** 设置选项卡下生成并称为 **AI O** 密钥，来替换 `main.py` 文件中的如下代码：

```
1 client = MQTTClient(client_id="example_client", server="io.adafruit.com",  
    user="Adafruit IO Username", password="Adafruit IO Key", port=1883)
```

假设一切都正确设置，那么您应该每秒钟看到“ON”和“OFF”之间的开关切换！

步骤 4: **Finished!**

就是这么简单！

现在，你拥有一个能工作的 **MQTT** 发布者、代理和订阅者，让您可以利用这种低带宽消息。您可以以任何您想要的方式，扩展 `main.py` 脚本：比如你想监控一个房间的温度或跟踪你的门打开的次数！

MQTT 是允许设备节省电力、数据, 以及支持大量流量的好方法。要知道 Facebook 甚至在他们的 Messenger 平台使用 MQTT!

原文链接: <https://www.hackster.io/bucknalla/mqtt-micropython-044e77>

原文链接: <https://www.wandianshenme.com/play/mqtt-pycom-micropython-build-iot-with-python>

玩点什么: <https://www.wandianshenme.com>