

# 用Python写一个基于Web的物联网应用程序

谢作如 浙江省温州中学

寒假期间,忙里偷闲设计了一个名叫“sfarm”的开源智慧农场方案。sfarm采用了应用最广泛的MQTT协议,多节点设备汇集到一个服务器上,不仅能自动灌溉,还实现了远程控制,定时采集各种数据。方案中涉及的开源硬件、传感器、电磁阀和其他灌溉设备,在淘宝上都能买到,因而得到了很多朋友的关注。其中福州连江五中的卢老师就在学校的大花坛上部署了一套,运行效果很好。

虽然系统运行正常,但很快就产生了新的需求:当有嘉宾或者学生来参观智慧农场系统的时候,如何快速展示这一系统的“物联”功能?毕竟打开SIoT(MQTT服务器)的Web管理页面,再进行演示总不够便捷,也难以体现个性化。这就需要写一个物联网应用程序,对接入的各个节点设备进行直观的管理。当然,我们期望这个物联网应用程序的开发难度不能太高,最好学生也能够开发。这样,智慧农场就和学生的物联网学习、编程学习密切结合起来了。

## ● 用Web方式开发物联网应用程序的优势分析

物联网应用程序的开发有很多种方式可以选择,比较常见的是用VB、C#之类的语言来编写,但是开发的应用程序要运行在电脑上,无论是演示还是平时管理都不够灵活。用App inventor编写一个手机App也是不错的选择,技术门槛也低,中学生大都能参与编写。但美中不足的是,这个App要事先安装,而且只能运行在安卓手机上,不够灵活且不具备普适性。

用类Scratch的软件来编写应用程序,也是可行的一种选择,如Mind+(一款图形化编程软件)内置了MQTT模块,可以编写物联网应用程序。Mind+的门槛很低,小学生也能够编写出一个非常个性化的物联网应用。但遗憾的是,Mind+的代码不能脱离Mind+环境运行,也不能编译为一款独立的软件。

毫无疑问,最灵活的方式肯定是采用B/S架构,编写一个单独的管理网页,即WebApp。在服务器上运行该程序,任何手机的浏览器都

能打开这个特定的网页,看到各种采集的数据,还可以通过网页按钮来控制“浇水”。这样的应用场景显然很“智能”很“贴心”,但遗憾的是,编写WebApp一般需要掌握一定的网页前端开发技术,而目前学生普遍缺少这方面的编程基础。

## ● 用Python开发WebApp

能不能用Python来编写WebApp?现在的中学信息技术教材几乎都采用Python语言了,如果能够用Python来编写网页,学生就不需要学习HTML、JS和CSS等网页开发技术。当然,答案是肯定的,一个名叫“Remi”的Python库,就是用来开发WebApp的。

### 1. Remi库简介

Remi是一个用于Python应用程序的GUI库,它将应用程序的界面转换为HTML,以便在Web浏览器中呈现。严格地说,我们不能用Remi库来编写传统的网站,而只将他当成Web形式的Tkinter库(Python最经典的图形界面库)来使用。如果要做网站,还是要老老实实学点前端知识,然后结合Python

```

from remi import start, App, gui
class MyApp(App):
    def init(self, args):
        super(MyApp, self).init(args)
    def main(self):
        container = gui.VBox(width=300, height=200, style={'margin': '5px auto'})
        self.lbl = gui.Label('你好, 我是一个WebApp!')
        self.bt = gui.Button('请点击这里')
        self.bt.onclick.do(self.on_button_pressed)
        container.append(self.lbl)
        container.append(self.bt)
        return container
    def on_button_pressed(self, widget):
        self.lbl.set_text('你点击了按钮!')

start(MyApp)

```

图1

的Flask框架来开发。

## 2. Remi库的安装

Remi可以采用pip命令安装, 具体如下:

```
pip install remi
```

## 3. Remi库的代码示例 (如图1)

运行这段代码后, 浏览器会自动打开一个本地的网址, 出现如图2



图2



图3

所示的界面。将“127.0.0.1”换成IP地址, 就能通过其他电脑、手机的浏览器来访问了。

点击“请点击这里”按钮, 界面会发生变化, 如图3所示。不用写复杂的JS代码, 在Remi的支持下, 网页交互就变得这么简单。

如果需要了解更多关于Remi库的资源, 可以访问github或者官方文档。

github地址: <https://github.com/dddomodossola/remi>

文档地址: <https://remi.readthedocs.io/en/latest/>

## ● 基于Remi编写基于Web的物联网应用程序

既然是编写物联网应用程序, 那么肯定还需要安装siot库。这也是“虚谷物联”团队开发的Python库, 因为MQTT的官方Python

库(paho-mqtt)编写出来的代码冗长, 不好理解, 于是委托上海蘑菇云团队在paho-mqtt的基础上进行了新的封装。siot库可以通过pip命令来安装, 命令如下:

```
pip install siot
```

## 1. 物联网应用的工作流程

因为“sfarm”每一个节点都有采集土壤、光线信息的功能, 都具备单独的电磁阀, 用物联网消息来控制。那么, 这个应用程序既需要显示消息, 又要能够发送消息(控制指令), 工作流程如图4所示。其实, 绝大多数的物联网应用程序, 都需要同时具备订阅和发送消息功能。

## 2. 核心代码1: 物联网消息的订阅

因为要同时订阅多个主题的消息, 代码中采用了“定点存放定时处理”的方式, 即先将接收到的消息放在一个字典(topic\_msg\_map)中, 然后定时在字典中查找是否有新消息, 读取新消息后就删除这一消息, 避免重复读取。参考代码如下页图5所示。

## 3. 核心代码2: 物联网消息的显示

显示消息需要不断刷新网页

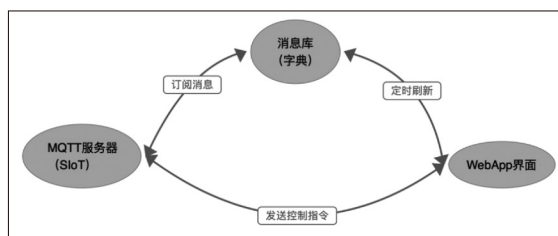


图4

```
import siot,time
topic_msg_map={}
iot_server = '192.168.3.136' # mqtt服务器地址
iot_user = 'scope'
iot_pwd = 'scope'
projectid = 'sf88' # 项目编号
def on_topic_subscribe(client,userdata,msg):
    global topic_msg_map
    topic_msg_map[str(msg.topic)]=str(msg.payload.decode())
def on_topic_read(topic):
    global topic_msg_map
    result=topic_msg_map.get(topic,None)
    if result:
        del topic_msg_map[topic]
        return str(result)

siot.init(' ',iot_server,user=iot_user,password=iot_pwd)
siot.connect()
siot.subscribe(projectid + '/light',on_topic_subscribe)
siot.subscribe(projectid + '/soil',on_topic_subscribe)
siot.loop()
```

图5

```
def showmqtt(self):
    while True:
        getcmd = on_topic_read(projectid + '/light')
        if getcmd:
            self.lbl_1.set_text('环境光照值:' + getcmd)
        getcmd = on_topic_read(projectid + '/soil')
        if getcmd:
            self.lbl_2.set_text('环境土壤湿度:' + getcmd)

        (target=self.showmqtt)

    t.start()
```

图6

```
def on_button_pressed(self, widget):
    global i,projectid
    i=i+1
    siot.publish(projectid + '/relay','1')
    self.lbl_9.set_text('成功发送浇水指令!次数:' + str(i))
```

图7

组件中的文字,这需要启动一个新的进程来处理,然后在这个进程中不断循环读取字典(topic\_msg\_map)中的消息。参考代码

如图6所示。  
**4.核心代码3: 物联网消息的发送**  
消息发送通过“浇水”按钮来

实现。参考代码如图7所示。  
**5.核心代码4: 物联网设备的切换**

sfarm是一个多节点工作的物联网系统,因而需要加上设备切换功能,比较简单的做法是用一个列表(project\_list)记录需要呈现的设备名称,然后在切换时修改全局变量(id)。参考代码如下页图8所示。

● 运行效果和反馈

Python的神奇,就在于其拥有无穷无尽的“库”,能够用来快速解决问题。从找到Remi库到写出代码,我仅仅花了一个晚上的时间,这种编程的感觉特别好。最终的代码运行效果如下页图9所示。

福州连江五中的卢老师将代码运行在sfarm系统的物联网服务器上,然后将访问地址做成二维码,用手机扫码(手机要能够访问服务器)的方式来打开这个网页,效果很不错,如下页图10所示。从难度上看,只要有Python编程基础,就能编出类似的程序。他正在将这方面的内容开发成一个物联网课程,让学生真正参与到sfarm的开发和维护中。

完成了这个物联网应用后,最大的收获并不是解决了sfarm系统的演示问题,而在于找到了一种设计Python程序界面的新方法。例如,可以用Remi库写一个基于Web的校园气象站、摄像头远程控制程序等。相对来说,用