

# 用掌控板做智能家居控制器

邱奕盛 上海人工智能实验室

谢作如 浙江省温州中学

涉及学科：物理、技术、艺术

为了让学生体会新技术的发展,笔者在温州中学人工智能实验中配置了一系列的智能家居设备,如智能灯泡、智能窗帘、智能插座和智能开关等。通过一个智能音箱设备,即可使用语音管理所有智能设备的工作状态。但是,用语音控制虽然看起来很方便,在某些特定的场合并不适合,如有噪音,识别效果并不好,再如上课时喊一声“小爱同学”,也有点尴尬。

在本栏目上一期的文章中,笔者用开源物联网网关Home Assistant(简称 HASS)实现了个性化控制各种设备,于是继续在HASS的帮助下,准备将开源硬件改造为智能家居的控制器。而自带屏幕的掌控板,肯定成为最佳选择之一。

## ● 掌控板做智能家居控制器的原理分析

Home Assistant有很多扩展插件,除了可以方便地连接各品牌智能设备,使得各品牌智能家居设备可以统一管理和控制外,还支持MQTT协议。掌控板等开源硬件虽

然不能直接接入HASS,但是经过MQTT服务器做中转后,即可实现对智能设备的控制,也就是说利用掌控板能够制作一个与众不同的智能家居控制器。其工作原理如图1所示。

从图1中还可以看出,只要将掌控板与MQTT服务器连接,消息传递就不再受限于局域网,可以在任何有网络的地方实现控制实验室的智能设备。

## ● 准备工作：基础环境的搭建和设置

### 1. 搭建MQTT服务器

MQTT服务器很多,HASS也

支持绝大多数的服务器。SIoT是一款适合中小学教师和学生学习的跨平台MQTT软件,下面以它为例介绍MQTT服务器的搭建。

在gitee.com/vvlink/SIoT上可以下载对应版本的SIoT软件。软件不用安装注册,直接运行即可。用户名为siot,密码为dfrobot。

SIoT服务运行在1883端口,Web管理端口为8080,可以通过浏览器查看(如图2)。

如果希望实现因特网的远程控制,那么可以将SIoT搭建在公网服务器上,或者直接注册企业的MQTT服务,如EasyIoT、OneNet等。

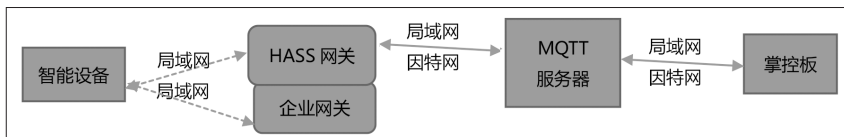


图1 利用掌控板控制智能家居的工作流程图



图2 SIoT的Web管理界面

2.HASS的安装与设置

上一期文章已经介绍了如何搭建HASS并接入智能家电设备。接下来,就需要配置MQTT平台地址。

首先,通过浏览器访问HASS的网页管理界面,地址是{HASS的IP}:8123。点击配置—设备与服务—添加集成—输入“MQTT”搜索。其次,如图3所示,填入MQTT服务器相关信息,其中服务器这一栏填写MQTT的服务域名或IP。如果使用的是企业MQTT服务,一般还需要SSL/CA身份认证,具体操作可以参考www.home-assistant.io/docs/mqtt/certificate/。

● 用掌控板控制智能插座

以SIoT控制智能插座的开关为例,需要配置一个“场景自动化”的脚本,具体操作如下:

打开HASS的网页管理界面,点击配置—场景自动化—添加自动化—从空的自动化开始,进入自动化配置。

在触发条件中选择触发条



图3 配置MQTT服务器信息

件类型为“MQTT”

(如图4),主题填写chazuo/kaiguan,这样,HASS会监听这个主题收到的

MQTT消息,一旦收到,就会触发“动作”(如图5)。在动作中,选择动作类型为“调用服务”,服务选择“开关:Toggle”表示切换开关状态。目标选择设备为智能插座。点击保存。

接着,添加一个自动化,实现设



图4 触发条件类型



图5 配置自动化

以YAML编辑

```
1 service: mqtt.publish
2 data:
3   payload_template: '{{ states("switch.cha_zuo_1") }}'
4   topic: chazuo/zhuangtai
```

图6 YAML编辑效果

备状态的查询,这一功能属于高级功能,需要通过YAML语言编写,具体流程如下:

在触发条件中选择触发条件类型为“MQTT”,主题填写chazuo/chaxun。在动作中,选择动作类型为“调用服务”,服务选择“MQTT: Publish”表示发送MQTT消息,Topic填写chazuo/zhuangtai,然后点击这个模块的三个点,切换到YAML编辑模式,添加一行payload\_template: '{{ states("switch.cha\_zuo\_1") }}',这里的states()代表获取实体switch.cha\_zuo\_1的状态,花括号表示是变量的引用。最后的效果如图6所示。

保存配置后,接下来就可以通过mPython给掌控板编程了。预设的功能是:按A键切换智能插座的开关,B键查询智能插座的状态。模块化编程代码如下页图7所示,运行效果如下页图8所示。

● 用掌控板调节智能灯泡的色彩

智能家电的控制不只有开和关的工作状态,还有很多其他的功能可以被控制,如窗帘的幅度、灯光的色彩等。我们可以通过以json格式的MQTT消息发送复杂指令控制



图7 掌控板代码



图9 YAML编辑效果

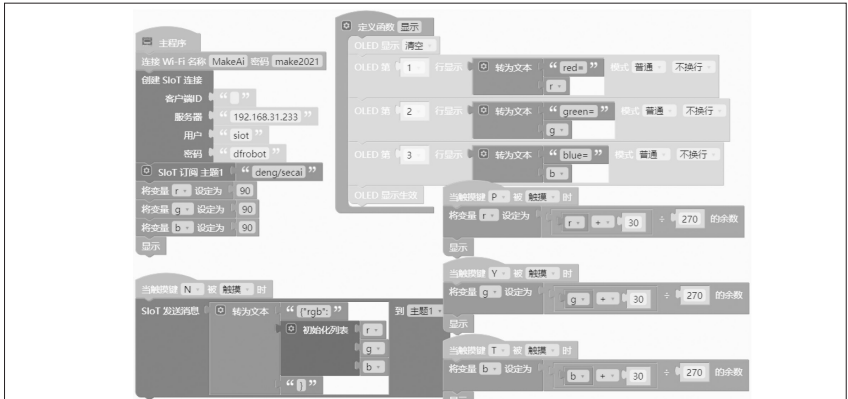


图10 掌控板代码

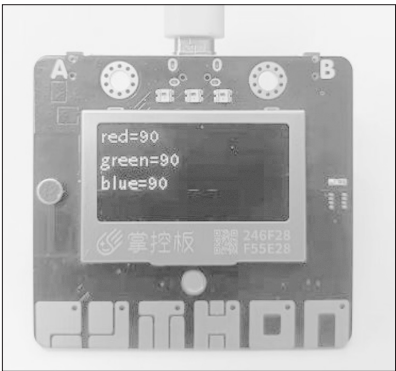


图11 掌控板运行效果

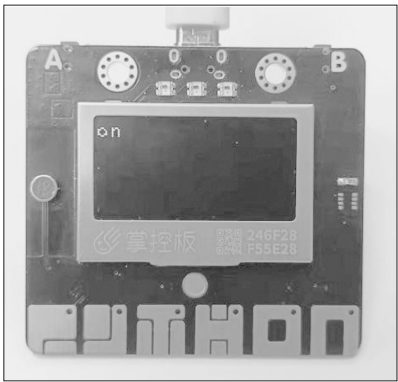


图8 掌控板运行效果

过的设备,如Yeelight彩灯,勾选“Brightness”并设置一个数值,然后切换至以YAML编辑,在data中添加rgb\_color: '{{ trigger.payload\_json.rgb }}',表示灯光颜色设置根据触发条件中接收到的数据。效果如图9所示。

接着,通过mPython给掌控板编程,模块化编程代码如图10所示,运行效果如图11所示。

### ● 小结

借助HASS和MQTT,笔者成功地实现了用掌控板来控制智能家居设备。实际上,除了掌控板,任何一款支持MQTT的软硬件都能控制智能家居设备,如行空板、虚谷号、树莓派等。MQTT的编程门槛很低,小学生用Mind+和mPython,也能设计一个与众不同的控制面板。只要创意无限,用户和智能家居的交互方式就会越来越灵活有趣。e

家电,这里介绍最常见的智能灯的色彩调节。

先创建一个“自动化”脚本,在触发条件中选择触发条件类型为“MQTT”,主题填写deng/secai。在动作中,选择动作类型为“调用服务”,服务选择“灯光: Turn on”表示开灯,目标选择已经添加