

用Home Assistant 搭建个性化智能家居系统

谢作如 浙江省温州中学
邱奕盛 上海人工智能实验室

涉及学科：信息技术、科学、艺术

智能家居在我们的生活中已经随处可见,买一个智能家居设备,轻点手机屏幕,就能完成设备的连接与控制,甚至可以设定一系列的自动化功能。例如,房间的灯泡可以根据人体红外传感器检测房间的空闲状态,来自动决定开关,或者设定每天晚上7点准时开灯,又或者通过语音指令来实现智能控制。总之,智能家居的普及极大地便利了我们的居家生活。

然而,尽管“米家”“天猫精灵”等平台提供了一定的智能化管理方式,但只能使用其提供的APP进行控制。想实现自己写一段Python代码控制设备,或者把开源硬件改造为智能家居设备控制器,都有些困难。于是,笔者把目光投向了一款智能家居的管理平台——Home Assistant(简称HASS),这是一个能够将不同品牌智能家居设备管理起来的开源物联网网关软件。

● HASS的工作原理分析

手机APP之所以能够远程控

制智能家居设备,是依赖于与设备相连的企业网关或者服务器。以智能灯泡为例,首先是智能灯泡和网关相连,手机APP发送命令到网关,网关再发送命令到智能灯泡,从而实现控制。但是,绝大多数的智能家居网关并没有开放编程接口,用户无法对网关进行编程。HASS相当于是一个额外搭建的智能家居网关,因为其开放了API接口,那么我们就可以编程来控制局域网内的智能家居设备,达到个性化控制的目的。

如图1所示,HASS和原来的企业网关并行工作,智能灯泡既能接受企业网关的控制,也可以接受

HASS的控制,这样就能确保原来的智能家居系统也能正常工作,不受破坏。

● 基于HASS的智能家居系统实现

HASS是一个开源项目,可以运行在很多硬件上。笔者选择用树莓派4来运行HASS平台,并且用来管理学校人工智能实验室中的智能设备。

1.HASS的安装

HASS提供了好多种安装方案,经过比较,采用Docker方案安装最为方便。

(1) 给树莓派安装Docker

树莓派连上Wi-Fi后,在

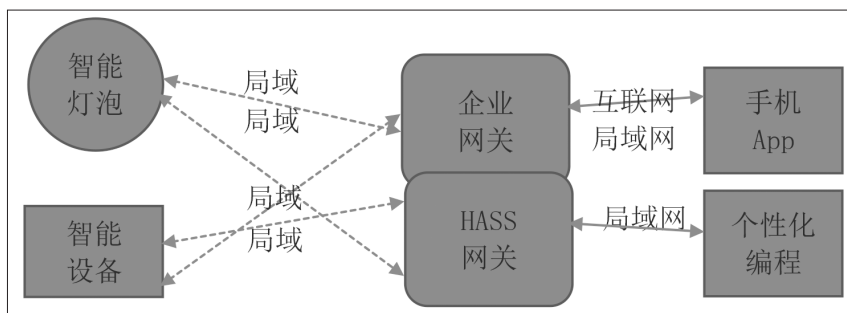


图1



图2

树莓派操作系统内使用命令行安装Docker。笔者强烈建议先安装Docker图形化工具portainer,然后通过浏览器来安装HASS。

(2) 在Docker安装HASS

在命令行输入代码完成安装。

2.HASS平台的设置

启动Docker容器后,在浏览器输入{树莓派的IP}:8123(默认端口号),即可访问HASS前端页面,第一次进入需要等待一会儿以完成联网更新,然后完成用户初始设定

即可。

进入系统后,需要申请一个长期访问令牌,用来做编程连接的身份认证。在网页左侧侧边栏选择最下面一栏的用户,然后点击右侧页面最下方的创建令牌,创建完成之后请务必保存好令牌的内容。由于令牌内容各不相同,下文以**your token**代替。

3.智能家居设备的设置

HASS支持的智能家居设备种类广泛,兼容各大平台,如小米、

天猫精灵等。这里以Yeelight智能灯泡和米家智能插座Wi-Fi基础版为例,来说明设备的接入和设置。这两个设备都可以在“米家APP”进行管理。

4.智能设备接入HASS

在浏览器输入{树莓派的IP}:8123

访问HASS平台,点击左侧侧边栏上的“配置”,然后在右侧页面选择“设备与服务”,在跳转的页面点击右下角的添加集成,来完成设备的接入。

(1) Yeelight智能灯泡接入

点击添加“集成”后,在搜索框输入Yeelight,选择该集成后会自动在局域网内查找符合条件的设备,并且自动完成接入。

(2) 米家智能插座接入

插座的接入要稍微麻烦一点,在搜索框输入Xiaomi Mii,选择该集成后还需要在网页上登录米家账号,然后选择智能插座设备,HASS平台会自动在局域网内查找选择的智能插座设备,并且自动完成接入(如图2)。

(3) 在HASS页面控制

点击左侧侧边栏的“概览”,可以看到刚才添加的设备已经显示在右侧页面中了。这时就可以通过网页上的开关进行控制设备了,如图3所示。

5.控制程序的编写和测试

HASS提供了WebAPI,通过网络即可实现对设备的控制。笔者选择了Python为编程语言,结合requests库进行编程测试。

(1) IDLE连接情况测试

输入下页图4所示的代码。

屏幕会打印出在HASS中的所有设备的状态。返回的内容是json格式,由于内容太多,IDLE会压缩内容,仅输出response=Squeezed



图3

```
import requests
url = "http://{树莓派IP}:8123/api/states"
headers = {
    'Content-Type': 'application/json;charset=UTF-8',
    'Authorization': 'Bearer **your token**',
}
response = requests.request('GET',url, headers=headers)
print('response=',response.text)
```

图4

```
url = "http://{树莓派IP}:8123/api/states/{entity_id}"
```

图5

```
import requests
url = "http://{树莓派IP}:8123/api/services/light/toggle"
headers = {
    'Content-Type': 'application/json;charset=UTF-8',
    'Authorization': 'Bearer **your token**',
}
payload = '{"entity_id": "{entity_id}"}'
response = requests.request('POST',url, headers=headers, data=payload)
print('response=', response.text)
```

图6

```
payload = '{"entity_id": "{entity_id}", \
    "brightness":100, \
    "rgb_color":[255,255,0]}'
```

图7

text(218 lines),可以双击展开,也可以将输出结果复制到json格式化工具网站上做格式化,以便更好地查看返回结果。

在输出的结果中搜索yeelight,可以找到智能灯泡的信息,留意其中的entity_id这一项。

(2) 设备信息查询测试

将url修改为下面的形式,其中{entity_id}是刚才查询的结果,如图5所示。

这样,返回结果就是智能灯泡的状态信息了。可以看到,“state”的值是“on”。类似地,也可以查看智

能插座的状态信息。

(3) 使用Web API控制智能灯泡

输入如图6所示的代码,代码中调用了HASS的service服务中的toggle,它的功能是切换灯的开关状态。service中还有更多可用的服务,如turn_on(打开)、turn_off(关闭)等,可以自行探索。

如果想对灯点亮后的亮度、色彩进行进一步设置,可以在payload中添加相应信息,如图7所示,这样,可以设置灯的亮度为100,颜色为黄色。

● 基于HASS的个性化控制

想法

一旦实现了智能家居设备的可编程控制,各种个性化的创意就涌现出来。

创意1:能够“吹”的灯泡。

用Python代码定时获取当前的天气预报,然后显示不同的灯泡色彩,这样的创意笔者已经做过,不够好玩。但是,如果在声音传感器上吹口气,就吹出灯光摇曳的效果呢?

创意2:魔棒控制灯光。

在哈利波特的魔法世界里,挥动魔棒就可以实现意念控制,进而可以利用加速度传感器感知魔棒的运动方向,让灯光呈现魔幻的效果。

创意3:根据主人情绪改变灯光。

可以通过摄像头捕捉人脸,经过算法分析人脸表情,根据情绪的高涨与低落,将灯光氛围与情绪匹配。

常见的创客作品往往局限于LED、舵机、直流电机之类的电子模块,我们一直希望创客作品能够直接和家用电器形成交互。现在有了HASS的支持,学校的人工智能实验就变成了一个新的研究智能家居和物联网技术的实践场,下一步笔者还会通过人工智能技术,用各种有趣的方式控制灯光、开关、窗帘,还有门锁等。e