掌控板结合APP Inventor2玩转物联网

谢作如　浙江省温州中学
金从军　17coding.net
张晴　浙江省教育技术中心

涉及学科：信息技术、物理

通俗地讲，物联网（Internet of Things）就是“物物相连的因特网”，其目标是让万物沟通对话。比如在电视机上装传感器，可以用手机通过网络控制电视的使用；在空调、电灯上装传感器，计算机可以精确调控、开关，实现有效节能；在窗户上装传感器，你就可以坐在办公室里通过计算机打开家里的窗户透气；等等。

物联网是创客空间中学生最喜欢挑战的新技术之一，如用Arduino来做一个基于物联网的自动浇花系统，远程获取传感器信息等。相对来说，用Arduino或者micro:bit设计物联网作品的难度还是有点高，需要借助于特定的物联网模块才能实现。而随着掌控板的出现，开源硬件连接网络变得容易，在TingWebIO库的支持下，开发物联网作品也越来越简单了。

● 技术分析

掌控板是创客教育专家委员会提出和设计的国产开源硬件。因其采用的ESP32芯片，是乐鑫最新的WIFI+蓝牙低功耗物联网芯片，性能十分强大，所以能够直接作为一个网络服务器运行。而APP Inventor是一个基于云端的，以图形化形式编程的安卓手机应用程序开发环境。它能将枯燥的代码编程方式转变为积木式的图形化编程，即使不懂得编程语言的人，也可以开发出属于自己的手机应用程序。

从技术上看，只要掌控板运行一个Web服务器，APP Inventor通过Web浏览框或者Web客户端组件，就能访问掌控板的资源。无论是感知还是控制，只要双方设定一个协议即可。但是TingWebIO的作者张路老师认为，完全可以用更加简单的方式，如将掌控板模拟为一个TingWebDB（微型网络数据库）服务器，那么用户只要记住关键词，就能用写入数据库的方式实现“控制”，用读取数据库的方式实现“感知”。

如图1所示，当客户端发出保存数据请求时，请求信息中会携带两个参数——标记和数据，服务器会将“标记”解释为掌控板上的输出资源，如当标记为“buzz”时，输出资源为蜂鸣器，并将“数值”解释为具体的输出值，如蜂鸣器的鸣响频率。同样，当客户端发出读取数据请求时，会携带一个“标记”参数，服务器会将参数解释为掌控板上的某个资源，并将该资源的状

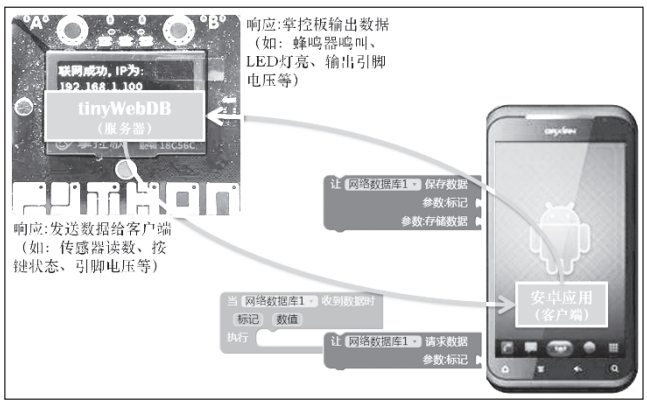

图1

表1

资源标记	资源名称	AppInventor操作类型	数据含义与取值范围
buttona	按键 A	请求数据	松开为 1，按下为 0
buttonb	按键 B	请求数据	松开为 1，按下为 0
touchpadp	触摸按键 P	请求数据	断开值>600，按下值<100
touchpady	触摸按键 Y	请求数据	断开值>600，按下值<100
touchpadt	触摸按键 T	请求数据	断开值>600，按下值<100
touchpadh	触摸按键 H	请求数据	断开值>600，按下值<100
touchpado	触摸按键 O	请求数据	断开值>600，按下值<100
touchpadn	触摸按键 N	请求数据	断开值>600，按下值<100
light	光线传感器	请求数据	0~4095
sound	麦克风	请求数据	0~4095
accelerometer	加速度传感器	请求数据	xyz 三轴加速度值：-2g~+2g
rgb<n> n 为 0,1,2	LED 灯珠	保存数据	逗点分隔红绿蓝亮度值，如 255,0,0
display 或 oled	OLED 显示屏	保存数据	显示文本：show:<文本>:<x>:<y>， 清空屏幕：fill:1 或 fill:0
buzz	蜂鸣器	保存数据	播放：on 或 on:<频率值> 停止播放：off
music	音乐	保存数据	播放乐谱： 1. 自编乐谱：音高:音长,音高:音长,... 2. 乐谱名： 如 birthday 3. 非音符：pitch:<频率>,<时长>
pind<n>	数字 IO 引脚	请求数据	高电位为 1，低电位为 0
		保存数据	断开：0 接通：1
pina<n>	模拟 IO 引脚	请求数据	返回：0~1023，对应于 0~3.3V 电压
		保存数据	0~1023，对应于 0.3.3V 输出电压

图2

态返回给客户端，如当标记为“buttona”时，掌控板将返回按键A的状态（1为断开，0为接通）。

如果想在安卓APP中获取掌控板的状态数据，或控制掌控板的输出，就必须清楚地知道掌控板上各项资源的标记、名称，以及接收数据的规格。表1中列出的相关信息，可以帮助开发者实现安卓应用与掌控板之间的协作。

注：表1中出现的<n>为相应资源编号，编写时需替换为具体数值（0、1、2等）。例如，模拟引脚0写作“pina0”，三个LED灯珠分别写作“rgb0”“rgb1”“rgb2”等。

功能测试

使用TinyWebIO功能比较简单，只要让掌控板运行这个程序即可。大致步骤如下：

（1）下载原生开发工具mpython2及驱动程序，用户可根据自己的操作系统版本选择下载不同的文件（下载网址：https://mpython.readthedocs.io/zh/latest/board/software.html）。

（2）下载TinyWebIO库文件及启动文件。其中tinywebio.py为库文件，main.py为启动文件样例（下载网址：https://gitee.com/roadlabs/TinyWebIO）。

（3）安装并启动mpython2，将tinywebio.py和main.py（需要修改代码中的SSID名称和密码）两个文件分别写入掌控板。

虽然很多版本都可以给掌控板升级固件或者刷写程序，但推荐使用mpythonX软件。mpythonX软件中的固件已经整合了tinywebio.py，升级固件后，只要编写连接WIFI代码，然后在oled上显示IP地址，再启动

appserver服务即可。mian.py的参考代码如表2所示。

安装TinyWebIO库中包含的“测试.apk”文件，设置好掌控板显示屏上的IP地址，就可以测试功能了。如上页图2所示，读取“light”即可获得光线值，写入“buzz”和“on”，掌控板上的蜂鸣器就会响起。

图3

应用拓展

在APP开发中，只要对服务器地址是掌控板IP的TinyWebDB进行操作，即可实现手机和掌控板的互动。其实，TinyWebIO作用并不局限于APP Inventor2，还有很多有趣的玩法。

表2

```
from mpython import *
from tinywebio import appserver
mywifi=wifi()
mywifi.connectWiFi('xzr','12345678')  #这里要修改为正确的SSID名称和密码
oled.fill(0)
oled.DispChar('联网成功,IP为：',0,0)
oled.DispChar(str(mywifi.sta.ifconfig()[0]),0,16)
oled.show()
appserver.start()
```

表3

```
import requests
host = '192.168.1.100'
port = '8888'
addr = 'http://%s:%s' % (host, port)
def submit(path, payload):
  server_addr = '%s%s' % (addr,path)
  try:
    r = requests.post(server_addr, data=payload)
  except:
    return None
  else:
    return r.text
def getvalue(tag):
  path = '/getvalue'
  payload = {'tag':tag}
  return submit(path, payload)
def storeavalue(tag, value):
  path = '/storeavalue'
  payload = {'tag':tag, 'value': value}
  return submit(path, payload)

# print(getvalue('light'))
# print(storeavalue('buzz', 'on'))
```

(1)用浏览器(手机、电脑)和掌控板互动。在浏览器中输入掌控板的IP地址，端口8888，就能看到一个简单的功能引导界面(如图3)。

点击“/storeavalue”可以控制掌控板，点击“/getvalue”可以读取掌控板的传感器信息，具体的使用方法和APP Inventor一致。这样，手机即使不安装APP，也能够和掌控板互动。

(2)PC端编程和掌控板互动。TinyWebIO实质上就是一个标准的Web服务器，可以使用任何一种编程语言，发送标准的HTTP请求，即可实现和掌控板的互动。参考Python代码如表3所示。

相对于其他开源硬件，掌握板的优势在于其天然支持WIFI，因为ESP32原本就是工业级的物联网芯片。借助TinyWebIO库，掌握板不仅和APP Inventor2无缝连接，还能用各种编程语言编写各种应用程序，远程获取各种传感器的信息，或者控制LED或继电器，让中小学生也能轻易完成一个“联通万物”的物联网作品，相信老师们开发物联网课程又有了新的选择。

如果对相关内容感兴趣，请关注主持人博客。