

国家课程、地方课程和校本课程构成了我国三级课程体系,对于物联网技术的普及而言,我们仍需要通过发展的眼光看待这个问题。谢作如老师身处信息技术需要参与高考的浙江省,其主导的这样一组课程,从国家课程的改造、初中课程的衔接和高中典型项目的延展的不同角度,给出了物联网课程进一步走向纵深的不同的可能性。

# 与时俱进,用更合适的方式 学习物联网技术

## ——对浙教版高中信息技术教材内置项目的“升级”分析

谢作如 浙江省温州中学

浙江教育出版社的高中信息技术教材(下面简称“教材”)的必修2模块中,“内置”了一个搭建信息系统的学习项目,即第四章《信息系统的搭建》中的“室内环境实时监测系统”。这是一个基于物联网技术的项目,涉及了软件和硬件。教材的编写工作在2016年左右启动,那段时间物联网技术还没有“飞入寻常百姓家”,但随着物联网技术的普及和发展,各种开源、免费的物联网服务器软件也陆续发布,搭建物联网应用有了更多选择,无论从价格成本、技术难度还是功能去衡量,教材中的方案都可以进行“升级”。

### ● 教材内置项目方案简介

高中信息技术必修2模块的名称为“信息系统与社会”,模块的学习目标是让学生认识信息系统在社会中的作用,并能合理使

用信息系统解决生活、学习中的问题。通过动手搭建“室内环境实时监测系统”这一项目,能够让学生从中真正理解物联网系统背后的原理,知道信息系统与外部世界的连接方式,了解常见的传感与控制机制。

系统功能需求主要包括:①利用智能终端结合相关的传感器,如温度传感器、湿度传感器、粉尘传感器、烟雾传感器等,实时检测室内环境的各种指标,并上传到服务器保存在数据库,供后期分析。②通过Web页面实时显示各种传感器上传的数据,能根据各种需求直接生成在线实时图表。③使用者可以对监测系统进行相关的设置,限定温度、湿度、烟雾等环境因素的临界值。一旦某项指标超过所限定的临界值,系统将自动触发声光警报,并发送警示信息通知

用户。④能够通过网络发送指令,控制相关执行器(如家用电器)自动调节室内环境。

考虑到各地的教学条件和师资的技术水平,教材中的监测系统仅采集了光线、温度和湿度信息,执行器方面也仅做了声光警报,并没有和家用电器进行关联。

### ● 教材内置项目方案分析

“室内环境实时监测系统”项目的硬件(智能终端)是micro:bit,物联网协议采用HTTP协议,编程语言是Python+MicroPython。综合各地的教师培训活动和教材实施情况看,一线教师们普遍反映教材的技术难度较高。究其原因,在于智能终端和物联网协议的选择并不合理。因为要搭建教材内置的物联网应用项目,编写组不得不增加了一些前置的支撑技术,无形中

拔高了整本教材的技术难度。

### 1. 对智能终端的分析

最初选择的智能终端是Arduino。考虑到Arduino只能支持C/C++语言编写,编写组选择了支持MicroPython的micro:bit。其实这是一种无奈的选择。首先,micro:bit自身不带Wi-Fi功能。虽然DFRobot公司开发了物联网模块(OBLOQ),用串口转发的方式来访问网络,但是要使用物联网模块,学生要先将micro:bit接上扩展板,然后通过杜邦线连接。模块和micro:bit的接线中,RX和TX引脚要交叉,这一点特别容易出错。其次,物联网模块上没有显示屏,出现故障后只能根据板载LED灯的颜色来判断是否正常连上Wi-Fi。这一联网过程对学生来说是个“黑盒”,有时很难分析是哪一步出错了,编程和调试的难度也提高了。另外,micro:bit的内存太小了,如果URL返回的HTML代码较多,板子就会系统崩溃而重启。

### 2. 对物联网协议的分析

物联网通信协议很多,如AMQP、JMS、REST、HTTP/HTTPS、COAP、MQTT等。2017年前,MQTT还算不上是主流协议。因而,编写组选择了HTTP协议。

既然采用了HTTP协议,就必须要先搭建Web服务器,既然要搭建Web服务器,则只好加入了Flask(一个极简的Python的Web框架)和Sqlite(Python内置的数据库)

的学习内容。牵一发而动全身,这导致了这一模块涉及的技术很多,门槛就偏高了。尤其是智能终端和Web服务器的连接,常常会因为电脑防火墙干扰而失败。

如果学生希望将这一系统部署到互联网上,即能够通过公网的Web服务器来存储、中转数据,技术门槛则会更高。因为,用Flask开发Web应用,需要同时了解前端和后端的网站开发技术,如HTML、JavaScript,技术难度太大了。

### ● 智能终端的升级思路

2018年是国产开源硬件的元年,各种为教育而设计的开源硬件如雨后春笋般涌现。综合价格、性能、编程语言和技术难度,笔者提出了两种替代micro:bit的智能终端升级方案。

#### 升级方案1: 采用内置Wi-Fi模块的智能终端

ESP8266、ESP32等国产芯片是为物联网而生,基于这类芯片开发的开源硬件是物联网终端的最好选择,如掌控板、FireBeetle和光环板等。教材中采用的micro:bit编程软件BXY,同时支持掌控板的编程,示例代码不需要做任何修改,就能直接运行在掌控板上。省去接

线的过程,项目搭建的技术门槛一下子就降下来了。

掌控板自带了常见的传感器和执行器,只要一块主板,就能完成教材中的基本传感和控制实验。用掌控板来连接网络,过程和手机、电脑联网几乎一致。如图1所示,代码仅仅6行,就实现了联网和显示IP地址的功能。

此外,掌控板支持的几乎是完整的MicroPython,而micro:bit支持仅仅是简化版的MicroPython。如掌控板可以用urequest、json等Python上很常见的库,实现较复杂的功能,而micro:bit是做不到的。

#### 升级方案2: 采用运行Linux系统的智能终端

在2017年前,能够运行Linux系统的智能终端并不多,2018年板载Arduino的虚谷号发布,随后虚谷物联项目启动,SIoT软件和siot库相继推出,用Python开发物联网应用变得非常方便。2020年pinpong库发布,让Python连接开源硬件变得非常简单,只要数行代码,就能实现读取传感器,控制执行器。

pinpong是基于通用串口通信协议Firmata开发的Python库。当

```
from mpython import *
import network
my_wifi = wifi()
my_wifi.connectWiFi('jif', 'pkq88888')
oled.DispChar(my_wifi.sta.ifconfig()[0], 0, 0, 1)
oled.show()
```

图1

Python访问开源硬件时,电脑会通过串口连接的开源硬件烧录一个特定的固件,然后借用Firmata实现对硬件的读写工作。pinpong兼容各种常见的开源硬件,实现了“一次编写处处运行”。具体用法请参考本刊2020年12月专题《用Python驱动开源硬件》。

使用pinpong的最大优势在于,搭建物联网项目不需要在Python和MicroPython之间来回切换。学生仅仅使用Python代码,就能完成与智能终端的交互:传感和控制。在课堂教学中,师生不用再去关心硬件的类型,只需要掌握最基本的引脚电平控制和读取的语句,利用普通电脑结合任何一款开源硬件,都可以完成引脚电平控制和读取的实验,或者搭建出物联网应用原型。如果想制作一个演示作品,那么把代码“部署”(复制)到一个卡片电脑上即可。

### ● 物联网协议的升级思路

前文已经提及,正是因为选择了HTTP协议,教材中才不得不加上Flask、SQLite等比较专业的技术。如果教材将HTTP更换为MQTT协议,那么搭建物联网项目不仅变得简单,而且功能更加强大。

MQTT(Message Queuing Telemetry Transport,消息队列遥测传输协议)是一种基于发布/订阅(Publish/Subscribe)模式的“轻量级”通信协议。MQTT最大的优点在于以极少的代码和有限的

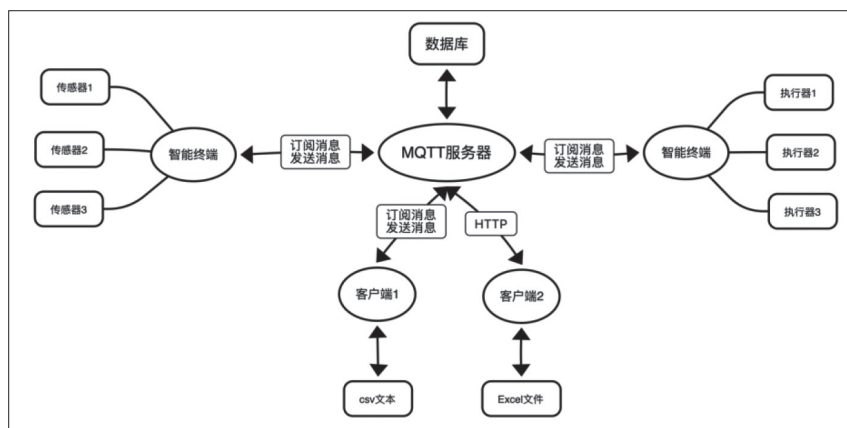


图2

带宽,为连接远程设备提供实时可靠的消息服务。目前,MQTT是物联网协议中应用最广泛的一种。

采用MQTT协议搭建教材项目,需要借助MQTT服务器(MQTT Broker),系统架构如图2所示。虽然和借助Web服务器搭建的系统架构并没有太大的区别,但MQTT服务器是现成的,不需要编写代码来搭建,数据库技术更不需要了解,学生只需要理解MQTT协议的消息发送和订阅机制,给智能终端编写相应的代码即可完成“物物相连”。至于想监视数据或者控制智能终端,可以用Python编写代码,保存数据则用CSV文本文件即可。如果还想降低门槛,直接用浏览器访问MQTT服务器的Web管理页面,也可以用现成的MQTT客户端工具,如MQTTX。绝大多数的MQTT服务器都自带了Web管理功能,有些还提供了WebAPI功能,兼容HTTP协议。

在教学中,一般需要在局域网内搭建服务器,所以一键部署的

SIoT是最好的选择。如果想部署到公网上,则可以直接借助EasyIoT或者阿里云等免费的MQTT服务器。掌控板、虚谷号、树莓派等都支持MQTT协议。虚谷号和树莓派自身还可以运行SIoT软件,将自己模拟为一个MQTT服务器。至于编程,教学中完全可以使用名为“siot”的MQTT库,同时支持MicroPython和Python,代码规范是统一的。(SIoT的下载地址:<https://gitee.com/vvlink/SIoT>)

当然,选择MQTT协议最重要的原因并不是为了降低技术门槛,而在于MQTT是当前主流的物联网开发协议。目前,除了Python和MicroPython外,主流的编程语言如C、C++、C#、GoLang、JavaScript都支持MQTT协议,甚至连Mind+之类的图形化编程语言都有MQTT扩展库。借助MQTT协议,用很简洁的代码就能搭建出一个物联网应用。学生不仅可以用MQTT来理解物联网的核心原理,

(下转第21页)

后记:最近的人工智能技术的发展仰赖于移动互联网技术普及之后人们大量贡献的数据,通过深度学习算法,使得机器可以更好地学习和理解人的各种物理特征,如我们的形象对应着人脸识别技术、我们的声音对应着语音识别技术、我们发表的文字配合的表情包对应着自然语言理解技术。但是机器目前理解的还只是人,对人和人之间的相互作用构成的人类社会还不够了解。因此,我们要获得更有用的人工智能训练成果,需要提供更复杂的人类社会的数据,这需要有更多的自动设备的参与,这些设备很多是物联网设备。相信物联网设备的普及应用,能够更好地让人类社会的行为“数字化”,创造出更加高效和智慧的社会治理形式。在这个过程中,反垄断、网络安全和国家安全的各种担忧始终是这一领域顶头上的“一朵乌云”,而国产芯片和自研物联网平台的“自主可控”、开源文化和社区的“创意分享”以及国家教材结合地方教材的“技术民主化”,是其中的重要基石。e

## (上接第14页)

还能用物联网技术真实解决一些生活中的难题。

### ● 建议和结论

综上所述,若在智能终端和物联网协议方面进行“升级”,教材中的物联网项目就会更接地气,更容易实施。抛开教材本身,对照课标和物联网技术的发展趋势,我们也可以得出如下三点建议:①物联网技术完全有必要进入学生课堂。②物联网协议应该从HTTP转向MQTT。③智能终端应该内置Wi-Fi功能。

物联网协议的选择和硬件无

关,对教材编写者和教师来说,可能更关心如何在五花八门的硬件中选择一个合适的智能终端。在笔者看来,如果从当前的硬件和技术来选择,掌控板、乾坤板之类显然是最合适的,因为软硬件成熟,案例丰富。但如果考虑到未来几年的发展,虚谷号、树莓派和香蕉派之类的能够运行Linux系统的卡片电脑可能更适合高中,也就是说用pinpong库实现物联网中的传感和控制功能,因为掌控板的教育定位本来就在小学和初中。

信息技术的发展日新月异,纸

质教材无论如何快速迭代,始终会比现实发展“慢一拍”。但是从HTTP到MQTT,恰好处于物联网技术发展的分水岭,无论从哪个角度看,都有“升级”的必要。本文虽然针对浙教版教材做分析,但是提出的几种物联网实现方案,对其他地区、其他版本的中小学信息技术教材都有参考价值。与时俱进,让学生动手玩转真正的物联网技术,拥抱万物相连的时代,是每一位信息技术教育工作者努力的方向。e