

# 用掌控板做课堂答题器

谢作如 浙江省温州中学  
李琦 浙江省诸暨海亮实验中学

涉及学科：信息技术、数学

在上课过程中,要了解学生的学习掌握情况,教师通常用提问的方式,让学生举手反馈。但是,用举手的方式来统计结果显然不够精确,而且只能用于判断题。在一些智慧课堂中,教师常常会借助可以在线统计结果的答题系统,设计以选择题为主的课堂练习,学生则使用平板电脑、手机之类的智能终端,及时反馈练习的结果。

因为课堂管理的问题,学生人手一台平板电脑或者手机,对许多学校来说还很难普及,比较常见的做法是购买手持式课堂答题器。其实,利用创客空间中的开源硬件,可以自主开发一个简单的课堂答题器,用低成本的方式实现课堂练习的实时反馈。

## ● 技术分析

课堂答题器是一个典型的信息系统。从原理上看,课堂答题器一般由答题终端、服务器和结果统计模块三大部分组成。答题终端接受学生输入的答案,并将身份标识信息和答案发送给服务器;服务器负责接收数据,处理后存入数据

库;统计结果模块以图示的形式呈现答题的统计结果,供教师访问查看。系统工作流程如图1所示。

笔者在编写浙江教育出版社出版的教材《开源硬件项目设计》时,曾经用micro:bit设计了一个类似的课堂答题器。因为micro:bit不支持Wi-Fi,所以只能利用无线广播技术,在服务器上增加一块micro:bit作为信息接收机,再通过串口采集数据到数据库,过程有点复杂。在掌控板发布后,笔者修改了代码,直接在服务器上运行Web服务,掌控板通过Wi-Fi提交答题结果,少了一个环节,操作就简洁多了。

用Python的Flask框架编写

一个Web服务器并不复杂。在数据库方面,自然选择了Python内置的Sqlite。而这个数据库仅仅用于存储数据,一个表(STU\_ANS)、两个字段就可以了,如下页图2所示。其中STU用来存储学生标识信息,ANS存储答案。

## ● 硬件搭建

因为micro:bit的输入按钮只有A和B,作为课堂终端总感觉捉襟见肘。为了能够正确输入,笔者借鉴了空调遥控器的控制方式,即用一个按钮来循环切换答案,另一个按钮表示确定。

当更换为掌控板后,这个问题就不复存在了:掌控板有6个触摸键,可以组合出很复杂的操作。答

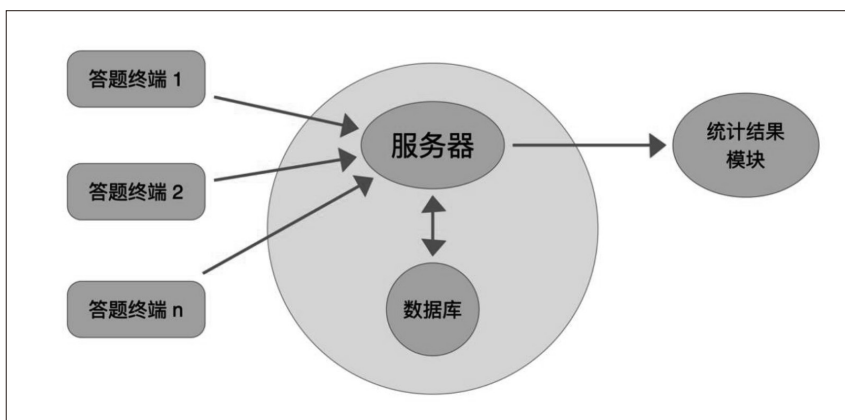


图1

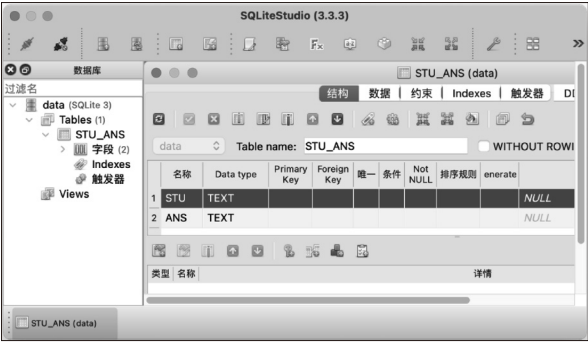


图2



图3

题终端只需一块掌控板和供电底板，不需要外接任何电子模块。笔者在网上找了一下，发现盛思设计的掌控宝、桃李科教设计的百灵鸽等扩展板，采用的是堆叠式设计，方便手持操作和随身携带，很适合作为答题终端。如图3所示，就是加了硅胶外套的掌控宝，看起来很有产品的“质感”。

● 代码编写

在课堂答题器项目中，软件设计是重点，大部分功能都需要软件来支撑。要考虑多终端的区分，还要用友好的界面呈现答题统计结果等。

1. 多终端区分的功能设计

“多终端区分”显然是一个重要的功能需求，因为要记录每一位学生的答题情况。实现用户身份标识的方式很多。例如，获取某

一设备的序列号，并将序列号作为唯一标识；给设备进行编号，保证编号是不重复的，便可以作为标识；使用设备的IP地址，用IP地址作为唯一标识等。

如果使用的终端很多，采用IP地址作为唯一标识的方式比较合适，因为这样做不需要给每一个掌控板烧录不一样的固件，只需要在路由器

上给掌控板的MAC地址分配固定IP。而如果只有十来个终端，直接编号再刷写固件也很方便。

2. Web服务器代码编写

Web服务器需要提供多个接口，每一个接口都是一个唯一的URL，如接收答题数据、返回统计结果等。返回数据采用标准的Json格式，具体规划如下表所示。

以统计结果为例，返回的Json数据样例如图4所示。统计结果的算法比较简单，只要不断累加即可，参考代码如下页图5所示。

因为代码较长，这里就不再提

URL	作用	输入数据	输出数据形式
/	默认主页，即呈现统计结果	无	网页
/test_connect	连接测试	无	Json
/post_ans	接收答题数据	答题结果	Json
/get_stu_ans	返回全部答题记录	无	Json
/get_ans_count	返回统计结果	无	Json

```
{"status": "success", "total": 5, "data": {"A": 2, "B": 2, "C": 1, "D": 0}}
```

图4

供具体的Python代码。如有需要，请访问GitHub: <https://github.com/xiezuoru/xzrbook>。

3. 掌控板的代码编写

掌控板的代码并不长，核心功能是连上Wi-Fi，然后在OLED上显示操作提示。通过触摸键更新答案，按下B键则提交答案到服务器。不得不说，使用MicroPython的urequests库实现HTTP协议的网页访问操作非常简单。核心代码如下页图6所示。

● 产品测试

将掌控板的代码下载到掌控板上后，开始测试这个答题系统。首先要运行Web服务，界面如下页图7所示。

打开掌控板的电源开关，等待数秒后，掌控板连上Wi-Fi，其OLED屏幕上会显示操作信息。选择答案后提交，显示信息如下页图8所示。

打开浏览器，访问 <http://127.0.0.1:8080>，即可看到一个简单的结果统计网页，如下页图9所示。

为了方便教师上课管理，笔者还特意设计了一个简单的网页，用浏览器打开后，就能设置正确答案，查看答题情况，或者清除当前

```
db=sqlite3.connect(db_file)
cur=db.cursor()
cur.execute("select ans from stu_ans")
result=cur.fetchall()
results={}
results["A"]=0
results["B"]=0
results["C"]=0
results["D"]=0
for r in result:
    if r[0]=="A":
        results["A"]=results["A"]+1
    elif r[0]=="B":
        results["B"]=results["B"]+1
    elif r[0]=="C":
        results["C"]=results["C"]+1
    elif r[0]=="D":
        results["D"]=results["D"]+1
```

图5

```
url = 'http://192.168.3.221:8080'
stu="STU001"
rsp = urequests.get(url + '/test_connect')
d = rsp.json()
oled.fill(0)
if d["status"]=="ok":
    oled.DispChar("服务器连接成功..",0,0,)
    oled.DispChar("学生: "+stu,0,16,)
    oled.DispChar("触摸选择答案, B 键提交..",0,32,)
    ans="A"
    while 1:
        if touchPad_P.read()<400:
            ans="A"
        elif touchPad_Y.read()<400 or touchPad_T.read()<400:
            ans="B"
        elif touchPad_H.read()<400 or touchPad_O.read()<400:
            ans="C"
        elif touchPad_N.read()<400:
            ans="D"
        oled.DispChar(ans,0,48,)
        if button_b.value==0:
            rsp = urequests.get(url + '/post_ans?stu=' + stu + '&ans=' + ans)
            d = rsp.json()
            if d["status"]=="success":
                oled.DispChar("答案提交成功!",12,48,)
            oled.show()
```

图6

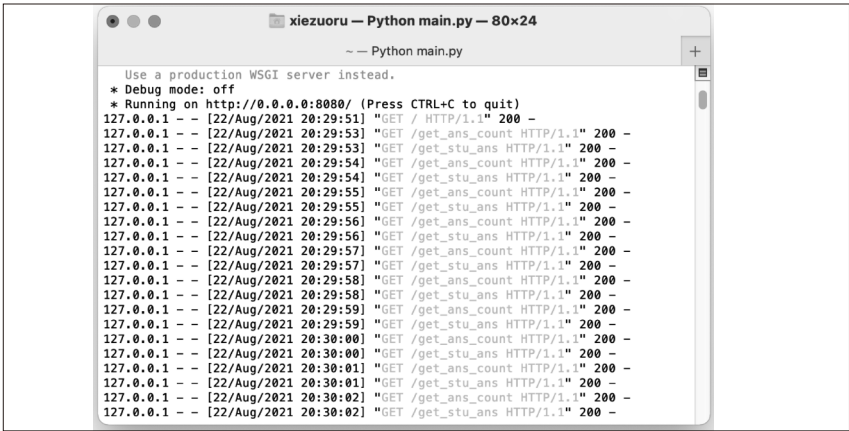


图7

答题记录,如图10所示。

笔者曾经在一堂小规模

开课中,使用过这个课堂答题器。

无论是从性能的稳定还是使用的



图8

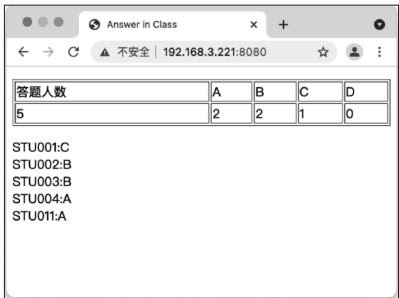


图9



图10

便捷方面去比较,它和购买的课堂答题器产品差别并不大。部分学生了解了运行原理后,还提出一些很不错的想法,如将题目内容直接发送到掌控板上通过OLED显示屏显示出来,答题结束后告诉学生是否正确。当学生掌握了开源硬件的基本操作后,就多了一个能实现创意的工具。智慧校园和智慧课堂,都将成为学生发挥创意的技术实践场,这正是我们努力的方向。e