

让掌控板和Scratch无缝连接

谢作如 浙江省温州中学

李敦勇 浙江省温州市平阳新纪元学校水头校区

唐柳峰 深圳盛思科教文化有限公司

涉及学科：信息技术、物理

Scratch是一款由美国麻省理工学院(MIT)设计开发的少儿编程工具。因为其采用了积木式的图形化编程形式,只要会使用鼠标,就能编写出各式各样的作品,大大降低了编程的门槛。目前很多地区的小学生编程课,大都是基于Scratch进行教学的。Scratch不仅是一款利用积木式工具制作二维动画的软件,还是一个学习互动媒体技术的平台。Scratch可结合PicoBoard、乐高WeDo、Arduino、micro:bit等相关硬件,与外部世界进行互动,如读取传感器值,驱动电机等,从而设计出各种有趣的互动作品。

PicoBoard是Scratch开发团队设计的传感器板,通常称为“Scratch传感器板”,其功能已经内置在Scratch的各个版本中。早在2011年,笔者就尝试用Arduino自制了一款PicoBoard,但因为各种传感器不好固定,外形也不看,就没有真正在教学中使用。最近,笔者在拿到掌控板

的时候,不禁萌发一种想法:能不能把掌控板模拟为PicoBoard,让掌控板和Scratch无缝连接起来呢?

● 分析规划

PicoBoard上有按键、光线、声音、滑杆等传感器,另外还有四个普通的电阻。这些信息可以从Scratch的扩展功能模块中看出,以Scratch2.0为例,编程模块的界面如图1所示。

掌控板上集成了光线和声音传感器,同时有A、B两个按钮,分别对应了PicoBoard上的按键、光线、声音这三种传感器(如图2)。而滑杆传感器可以使用六个触摸按键来模拟,这样就基本上具备了模拟PicoBoard的条件。另外,掌控板的加速度传感器数据,可以用阻力(电阻)A、B、C来输出,这样学生就能做出更加有趣的交互作品了。

考虑到使用习惯,笔者将掌控

板的按键B,模拟为PicoBoard的按钮。加速度传感器Z轴信息,一般不太容易使用,于是舍弃,留出一个“阻力D”作为扩展接口。确定这些信息后,再对应PicoBoard的协议,分别进行了通道的编号(如下页表1)。

● 代码编写

Scratch开发团队定义了PicoBoard的协议。二者采用串口

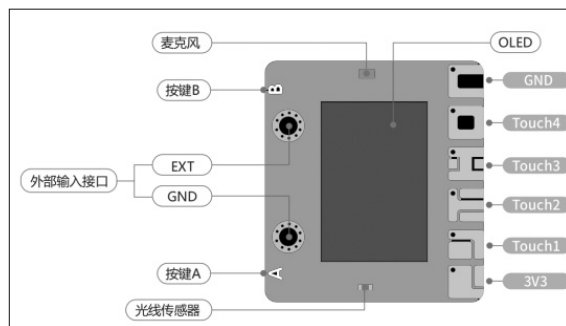


图1



图2

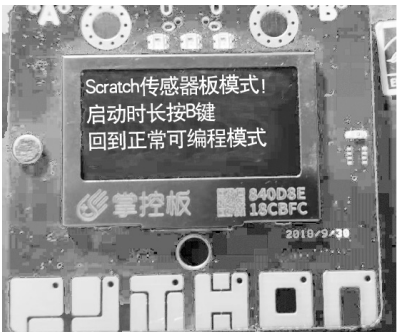


图3

表1

传感器通道	PicoBoard	掌控板
4	resistance-A	按键A
2	resistance-B	加速度传感器X轴
1	resistance-C	加速度传感器Y轴
0	resistance-D	预留外接温度电阻
3	button	按键B
5	light	光线传感器
6	sound	声音传感器
7	slider	触摸按键

表2

```
if request[0] == 0x01: #当接收到Scratch发来的0x01字节
    extValue=int(ext.read()/4)
    convert(0,extValue)
    reading = accelerometer.get_y()*1000 # 加速度的Y轴
    if reading >= 0:
        reading = int(reading / 2) + 512
        convert(1, reading)
    else:
        reading = 512 - abs(int(reading / 2))
        convert(1, reading)
    reading = accelerometer.get_x()*1000 #加速度的X轴
    if reading >= 0:
        reading = int(reading / 2) + 512
        convert(2, reading)
    else:
        reading = 512 - abs(int(reading / 2))
        convert(2, reading)
    if button_b.value()==0: # 按键B
        convert(3, 0)
    else:
        convert(3, 1023)
    if button_a.value()==0: # 按键A
        convert(4, 1023)
    else:
        convert(4, 0)
    convert(5, 1023-light.read()) # 光线传感器
    convert(6, sound.read()) # 声音传感器
    convert(7, ScanTouchpad()) # 触摸按键
```

通信,波特率38400bps。当Scratch发送0x01时,PicoBoard就发送八路的传感器数值。这些信息都是开源的,可以参考sparkfun分享在github的PicoBoard源码(开源地址: <https://github.com/sparkfun/PicoBoard>)。

其实有很多人基于各种硬件,编写了模拟PicoBoard的代码,如kos Vecsei的Scratchbit,能够将micro:bit模拟为PicoBoard,用MicroPython编写。研究这些代码,很快就能弄明白PicoBoard的通信机制。

(Scratchbit的开源地址: <https://github.com/vecsei/Scratchbit>)

我们采用MicroPython来编写PicoBoard的代码,核心代码如表2所示。

需要说明的是,为了让交互更加友好,当掌控板和Scratch进行通信时,板子上的RGB灯将会点亮。限于篇幅,这些没有在上述的核心代码中体现出来。完整的代码可以在掌控板的GitHub地址的examples中下载。(掌控板的开源地址: <https://github.com/labplus-cn/mPython/>)

● 功能测试

打开mPython软件,将上述代码刷入,并设置为默认启动。掌控板显示如图3所示的文字,表示启动成功。

1.在Scratch1.4测试

打开Scratch1.4,在“侦测”中找到PicoBoard的编程模块,编写一个读取传感器值的程序,并单击绿旗运行。等待10秒左右,当三个RGB灯都亮绿灯时,表示连接成功。这时,就能看到掌控板发来的各种传感器信息了。

2.在Scratch2.0中测试

打开Scratch2.0,在“更多积木”中添加扩展板PicoBoard。稍等片刻,软件界面上出现“PicoBoard”指令模块,当“PicoBoard”右边的黄色图标变成绿色时表示连接成功,同时掌控板上的三个RGB灯为绿色闪

表3

按钮	p	py	y	yt	on	n
取值	0	约10	约20	约30	约90	约100



图4



图5



图6

烁状态[注:可直接使用网页编程(<https://Scratch.mit.edu>),也可以使用PicoBoard,但要安装Web浏览器插件]。

● 应用拓展

下面以Scratch2.0为例,利用“滑杆”来做一个“黄山日出”的效果。

1. 前期准备

(1) 下载黄山图片,并利用图像编辑工具将图片中的“山”抠出来。

(2) 掌控板滑杆数值取值的测试。

掌控板对应的滑杆为6个触摸按键(p、y、t、h、o、n六个字母),

滑杆的初始值约为50.0。按p键是0,同时按住py是10,以此类推,刚好把100分为10档(如表3)。

2. 制作过程

(1) 硬件连接。
将掌控板设置为Scratch传感器板模式,并与Scratch成功连接。

(2) 角色。将黄山图片设为背景,黄山抠图作为角色1,太阳为角色2。

(3) 脚本。

背景:因滑杆的初始值约为50,所以将滑杆传感器值进行四舍五入,如果其等于50的话一直等待。当触摸其他按键时,返回值将发生变化,不等于50,进入下面的“重复执行”,然后亮度随着滑杆传感器值的变化而变化(如图4)。

角色1(黄山抠图):开始移至最上层,这样可以挡住太阳,让日出效果更佳。同样在滑杆传感器值没有发生变化时一直等待,然后亮度随着滑杆传感器值的变化而变化(如图5)。

角色2(太阳):开始先将太阳固定在(-20,-20)的坐标位置,同样在

滑杆传感器值没有发生变化时一直等待。为了实现太阳斜向上的效果,将X、Y坐标都随滑杆传感器值的变化而变化(如图6)。这样当单击绿旗运行时,用手指从掌控板的p键划向n键,这时太阳会慢慢从山的后面升起,天色也慢慢变亮。

当掌控板处在PicoBoard模式时是无法读取文件和刷入程序的。如果想进入读取文件或刷入程序的编程模式,得让掌控板进入REPL模式,具体方法为按下按键B不放,然后按下REST键,直到OLED显示屏上显示“replMode”字样才松开按键B。这时,掌控板又可以恢复到自由编程模式了。

掌控板的价格比市场上通用的Scratch传感器板便宜不少,还可以随时恢复到可编程模式,可谓一板多用,价美物廉。相信很快会有更多的教育创客开发出新的玩法,如利用掌控板的无线通信功能,做网络版本的Scratch传感器板等,从而让更多的孩子爱上Scratch,爱上编程。e

如果对相关内容感兴趣,请关注主持人博客。

