



让Arduino成为Android的“可穿戴设备”

谢作如 浙江省温州中学

Android手机虽然已经附带了多种传感器,如方向、重力、距离、加速度等。并且早在Android2.3(gingerbread)系统中,Google就提供了11种传感器供应层使用。但是,并非所有Android手机都把这些传感器配齐了,如温度、湿度、气压之类的传感器就不常见。为了使Android手机能支持更多的应用,越来越多的手机外设就涌现出来,可穿戴设备开发在短时间内就成为了硬件开发方面的“时尚”。其实,可穿戴设备的开发门槛并不高,通过Arduino,我们也可以初步体验Android的“可穿戴设备”的开发。

可穿戴设备指直接穿在身上,或是整合到用户的衣服或配件的一种便携式设备。手机可穿戴设备将会给我们的生活、感知带来很大的转变。但是,可穿戴设备中的传感器未必一定是手机中所没有的,如最近很流行的智能手环,其核心传感器就是加速度传感器和陀螺仪,都是智能手机标配的传感器。我们选择了温度和湿度传感器,通过Arduino将传感器数值发送给Android手机,以期通过这一项目来体验可穿戴设备和手机的信息互动。

● 项目描述

本项目设计了一个简单的通讯协

议,让Arduino能通过蓝牙将各种传感器(以温度和湿度传感器为例)的信息传输到Android手机,并显示出来。本项目的难度设计为中小学生都能完成,Arduino的程序我们选择用ArduBlock编写,而手机的程序由App Inventor编写。这个项目的流程如图1所示。



图1

● 项目分析

1. 协议设计

与本栏目上期文章《用Android控制Arduino小车》一样,首先需要为Android和Arduino设计通讯协议。这个协议规定了数据的格式以及数据代表的意义。针对ArduBlock教育版和App Inventor的特点,我设计了一个简单的通讯协议,各类数据的含义说明如表1。

表1 通讯协议表

数据(文本)	含义	
	传感器端口	传感器数值
010012	0	12
110123	1	123
211012	2	1012
310001	3	1
.....

因为App Inventor没有二进制的数据处理能力,这个通讯协议只能使用字符形式传送数值。虽然效率不高,但

也通俗易懂。之所以采用6个字符,是因为Arduino的A/D的转换分辨率为10位,即0~1023之间,需要留下4个字符,第一个字符是为了标识模拟传感器的引脚。而第二个字符永远是1,并没有实际作用,仅仅是因为在ArduBlock教育版可以使用“10000+传感器值”的简单表达

式来补足传感器数值不足4位时需要补上的“0”。

另外,ArduBlock教育版的串口打印会自动加上回车(这也是教育版为了简化积木模块而牺牲其更多功能的无奈之举),回车和换行刚好要占去2个字节,实际一次传输的是8个字符(字节)。

如果抛开这两款软件,我建议可以参照S4A的通讯协议,效率较高,2个字节就能完成一个传感器数据的传送。

2. 设备选择

虽然这是一个测试的项目,并非开发真正意义上的可穿戴设备产品,但我还是物色了一款体积较小的Arduino板子,做出“可穿戴”的感觉来。经过比较,DFRobot公司推出的Bluno Nano主控板成了我的选择。Bluno Nano集成了低功耗的蓝牙4.0芯片,而且只有拇指大小,非常适合用来开发做可穿戴设备的原型(如下页图2)。

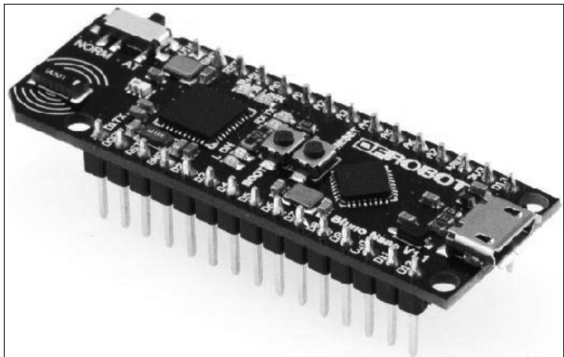


图2

除了Bluno Nano主控板外,只有硬币般大小的Microduino的产品也很适合做可穿戴设备的原型(如图3)。

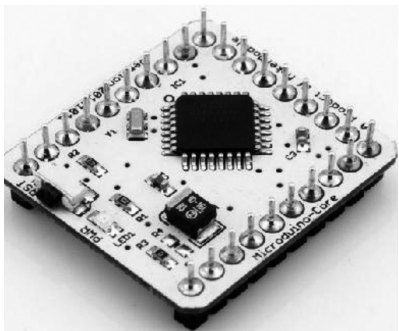


图3

当然,如果仅仅想测试Arduino和Android的连接,可以找任何一款蓝牙模块来实现,具体可以参考本栏目的上期文章。至于温度和湿度传感器,网上可以选择的店铺很多,价格从几元到十几元不等,不再一一说明。


● 编程实现

1.Arduino程序

我们使用ArduBlock教育版来写Arduino的代码。ArduBlock教育版是一款经过优化的版本,界面清爽、分类清晰且命名规范,适合在中小学的课堂上使用。其串口输出功能很简单,就一个“串口打印加回车”积木模块,这一积木模块和Arduino代码的对应关系如表2。

因为表1设计的协议,我编写了从串

表2 ArduBlock串口打印积木

ArduBlock代码	Arduino代码
	Serial.print("message"); Serial.println();

口输出模拟端口A0和A1值的代码,每隔300毫秒传输一次数据。其中用“10000+传感器值”的简单表达

式,补足了传感器数值不足4位时需要添加的“0”。这样加上回车换行,传输一个传感器的数值刚好需要8个字节(如图4)。

转成Arduino代码后为:

```
int _ABVAR_1_A0 = 0 ;  
int _ABVAR_2_A1 = 0 ;  
  
void setup()  
{  
    Serial.begin(9600);  
}  
  
void loop()  
{  
    _ABVAR_1_  
A0 = ( analogRead(0)  
+ 10000 ) ;  
    _ABVAR_2_  
A1 = ( analogRead(1)  
+ 10000 ) ;  
  
    Serial.  
print("0");  
  
    Serial.print(_  
ABVAR_1_A0);  
  
    Serial.println();  
  
    Serial.  
print("1");
```

```
Serial.print(_ABVAR_2_A1);  
Serial.println();  
  
delay( 300 );  
}  
  
注意: 如果出现下载失败的情况,  
请参照本栏目上期文章进行相应处理。
```

2.App Inventor程序

Android端App程序的主要功能是接收Arduino发送的数据,然后显示在屏幕上。代码并不复杂,除了常见的Label、TextBox、Button外,还需要ListPicker、BluetoothClient这几个必要的控件。因为要实时接收数据,我还加上了Clock这一时间控件。具体界面如图5所示。

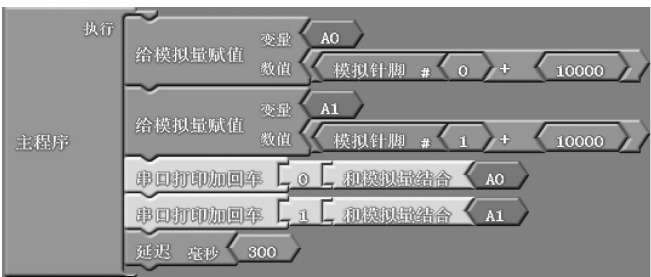


图4

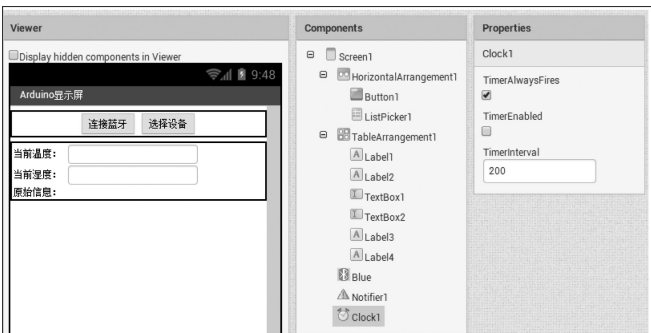


图5

App中所涉及控件的作用简要说明如表3。
给Button1控件编写Click事件,为了使程序更加完善,我添加了针对各种

表3 App各控件作用说明

控件类型	控件名称	控件作用
Label	Label1、Label2、Label3、Label4	显示文字
TextBox	TextBox1、TextBox2	显示传感器数值
Button	Button1	连接或者断开蓝牙
ListPicker	ListPicker1	选择设备列表
BluetoothClient	Blue	蓝牙设备(客户端形式)
Notifier	Notifier1	弹出提示框
Clock	Clock1	定时获取蓝牙串口数据

接下来,我们可以开始编写代码了,意外情况,会显示不同提示的代码(如图7)。
其中给ListPicker1控件写选择蓝牙设备的代码。需要注意的是,这个控件仅能显示手机已经配对过的蓝牙设备(如图6)。
Clock1控件的TimerEnabled属性默认值为false,当连接了蓝牙设备,TimerEnabled就修改为true。
Clock1控件Timer事件代码如图8所示。

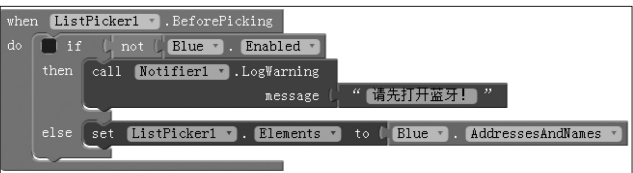


图6

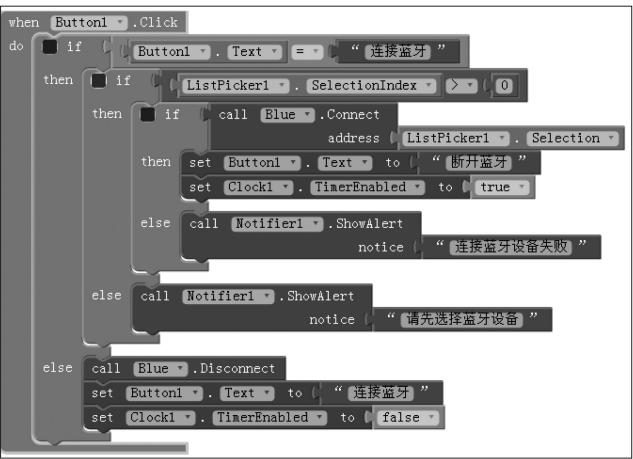


图7

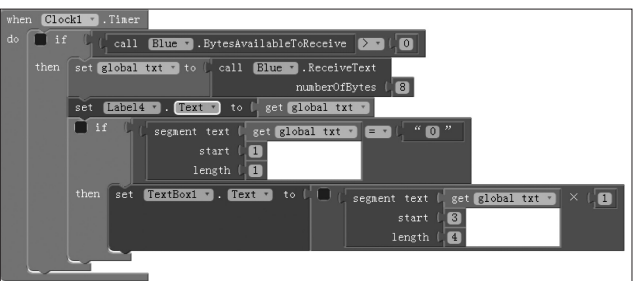


图8

来的数据了。



图9

● 拓展应用

在本项目中,Android仅仅将接收到的传感器数据显示出来,并没有做进一步的处理,也没有根据传感器信息,控制Arduino执行相应的动作。一般来说,手机还要对可穿戴设备的数据进行进一步处理,并存储在云服务器上,以便更加深入地进行分析。例如,智能手环可以统计用户每天的运动路径、消耗卡路里和摄入热量,也能根据事先的设定,通过振动马达来提醒用户应该运动或者休息了。结合本栏目上一期文章的内容,相信让中小学生实现这样的功能并不会很难。

● 总结

本项目还可以应用在某些特殊场合,如无线抄水电表,甚至也可以把Android作为Arduino的蓝牙显示屏来使用。蓝牙4.0的功耗很低,应用范围很广,Android手机结合Arduino后,功能将大大提高。从本质上看,可穿戴设备和智能家居、物联网并没有太大的区别,App Inventor编程的价值就在于其和硬件结合紧密,让学生们玩玩最新的技术吧。e

3.运行测试
编译为APK文件,然后安装。App的运行效果如图9所示。Bluno Nano主控板上的蓝牙默认配对码是1234,连接上后,就能看到传送过