**STEM公共课教学材料代码说明**

一般说明

* 所有连接方式都在代码中，本文件主要写的是一些前提条件、如何操作、注意事项之类的
* 单个数字表示连接所用的为数字端口，即D1~D13。模拟端口会写为A0~A5
* 有些例程要通过Serial Monitor看结果，所有的例程的Monitor比特率统一都是9600
* 杜邦线使用的一般原则：
  + 红色接电源正极 VCC 5V 3.3V
  + 黑色接电源负极 GND
  + 一个传感器所用到的线要使用不同颜色的线，便于区分

Arduino

**BT\_Test**

* 适应很多蓝牙，HC-05、HC-06都可以
* 需要在手机上装一个能给连接成功的蓝牙发信息的App，安卓手机可以直接安装“BlueSPP\_v7.0.apk”。苹果手机可以安装“Bluetooth Terminal”
* 蓝牙一般都是BT-05这样的，直接连接就行，不用配对码。连接上蓝牙之后，给连接的蓝牙发"A"，就能收到"Hello World"的回复

**DS1302\_Test**

* 这个时钟模块比较旧了，现在成本相对没那么高精度又相对高的模块是DS3231模块。不过原理都大同小异，无非就是设置时间，以及读取时间。还有一个处于中间的模块是DS1307。
* 目前的代码里边，需要自行把要设置的时间写进DS1302，但这样肯定会有人问能不能自动把PC的当前时间写入DS1302，而不用自己写个时间这么傻。这个是有解决办法的，需要通过tmElements\_t来获取当前PC的时间之后，一个一个解析为year、month、day等。如果大家有兴趣的话可以自己写个函数来转换。
* 例如DS3231这个模块，是有专门的一个library封装了很多函数，直接把tmElements\_t对象丢进去set函数就完事了，很方便

**Servo\_Test**

* 这个功能比较简单，就两个：一个是从0到180度往返循环；一个是0 - 90 -180 - 90 - 0
* 开关是#define SMOOTHMOVE，没注释就是第一种情况，注释掉就是第二种
* 这么写主要是顺便想让学生明白这是Arduino的一个比较常用的"开关"的写法，这个例程的代码足够简单，容易理解

**Ultrasonic\_Test**

* 较简单，大部分同学都在课堂上完成了，需要在Serial Monitor里边看结果，注意比特率要设置为9600。

**TFTLcdST7735\_Test**

* 芯片型号是ST7735，但这个例程能适应adafruit ST77XX的TFT LCD
* 要先安装Adafruit-GFX-Library https://github.com/adafruit/Adafruit-GFX-Library
* 要先安装Adafruit-ST7735-Library https://github.com/adafruit/Adafruit-ST7735-Library
* 安装Library使用IDE自动的库管理功能比较方便
* 需要连接5个接口：CLK/SDA/RS/RST/CS，具体对应关系在代码开头
* 我们这个LCD是默认背光的，所以没有一个专门的LED接口
* 如果要显示自己的东西，看代码就行，把相应的函数里边的内容换成自己想要显示的内容

**ULN2003andStepper\_Test**

* 代码很简单，主要是得外接5V电源

**Nrf24l01\_Test**

* 先安装RF24库 https://github.com/nRF24/RF24

**ColorSensor\_Test**

* 代码上传成功后，打开串口监视器，手动校正颜色步骤
* 用一张白色的纸放在传感器上面，查看读数范围，例如在38~40之间跳动，则记录40
* 用一张红色的卡片纸放在传感器上面，查看读数范围，例如在349~352之间跳动，则记录350
* 设置代码中的值，示例程序中对应语句为：

frequency = map(frequency, 40,350,255,0);

修改为实际取值

1. 同理用绿色，蓝色纸重复 1~3 步骤，修改代码（以下为示例程序中的源代码）

frequency = map(frequency, 50,420,255,0); //绿色

frequency = map(frequency, 36,325,255,0); //蓝色

**无伤电路连接工具介绍**

* 链接：https://www.tinkercad.com
* 使用邮箱注册后点击Start Tinkering即进入官方三维建模教程，有兴趣的同学可自行尝试
* 重新进入主页选择Circuits模块即可进入官方教程，跟着做一遍之后就会用了

