第十六课，呼吸门垫

1.呼吸灯

LED灯大家都已经熟悉了，他就是一个发光二极管，电流从正极流入驱动二极管发光然后从负极流出。其次我们看到了门垫这个词，是不是很熟悉？每次我们这次又有“回家”了。我今天要做一个非常有意思的灯，呼吸灯。当我们回到“家中”的时候我们的呼吸灯就会亮起来。对于呼吸灯大家可能还不太明白，什么是呼吸灯？就是LED按照由暗到亮，再由亮到暗这样慢慢变化的发光。这样就是呼吸灯了。我们可能见过一些鼠标或者键盘忽悠呼吸灯的效果。先来想想他是怎么工作的吧。呼吸灯：发光方式是由暗到亮，然后由亮到暗。那么说明他的电流在变化着。从大到小，在从小变大。来回循环，就形成了呼吸灯的形式了。那么我们今天就来学习一下这个控制电流的方法

2. PWM

# PWM，中文是脉宽调制，是指用微处理器的数字输出对模拟电路进行控制。是一种对模拟信号电平进行数字编码的方法。那么我们今天会用pwm对电路的电平控制 ，从而改变LED的电流，从而达到我们需要的呼吸灯的功能。

# 电路连接方式和上节课一样。

# 接下来我们看一下程序。让我们一起在程序中了解PWM的使用

3.16\_1WelcomeLED

打开程序，首先我们看到我们导入的包文件中，并没有一个PWM的文件，那是因为PWM在GPIO里。是RPi提供给我们的。所以并不需要单独导入了。首先我们用create（）方法来连接游戏，然后再一个我们确定好的位置创建一个羊毛块，（怎么找位置坐标，我不说了），

然后我们要设置一下LED的针脚号，（22号，大家请随便。不是固定的22.）然后我们设置了针脚编号规范。设置22号为输出。然后我们开始应用PWM了。创建对象pwm初始化。设置针脚为我们用的22号，频率为50（就是电流变化的速度）.这个频率不要超过100，因为频率太高，CPU承受不住压力。pwm.start(0)启动pwm，参数为空占比，范围是0-100。

接下来就是flash（）函数了。我们之间一部分。

xrange：我们已经用过range了。range（start，stop，step）start和stop是列表开始和结束。step是步长，默认值为1（如果不需要更改可以不写step）。

例如range（0，10）那么输出0，1，2，3，4，5，6，7，8，9我们可以在shell中做一下实验，for a in range(8,10):print a，

例如range（5，10）那么输出5，6，7，8，9

例如range（0，10，3）那么输出0，3，6，9

那么xrange和range的使用方法是完全一样的。不同的是，range生成的是对象。xrange生成的是一个对象。所以一般我们用xrange就行了。当然，用range也是一样的。

ChangeDutyCycle()用这个函数就可以改变电流了。那么我看这个循环，在time中参数是.02也就是0.02的意思。在循环中，每0.02秒循环一次。每次i加5，循环100次。每次电流都会增加。然后紧接着在进入下一个循环。从100每次递减5，直到0

接下来就简单了。在try下运行当我们玩家坐标和羊毛坐标一样时，调用flash（），else不然就关闭pwm并且清理针脚设置。

4.总结

今天主要就是学习Pwm这个脉宽调制，这个函数，用处很多。可以更改电流，那么就可以改变LED的亮度。电机的转速等等。

同时，我们也学习了一个新的方法，xrange，使我们的代码变的更加灵活多变，同样也更加方便我们以后的使用。

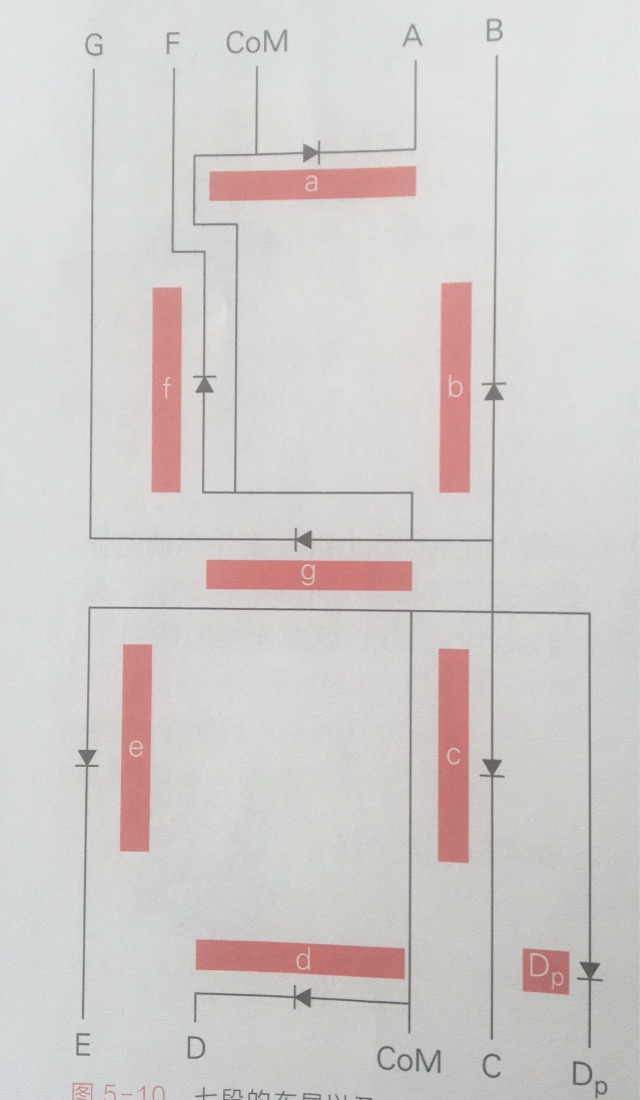
第十七课，七段数码管

1，数码管

数码管，就是将8个二极管组合成一个电子元件。其中有一个是表示小数点的，另七个可以组成各种团。这是单位二极管，把2个或者多个单位数码管组到一起，就成了多位数码管了。我们今天学习的是单位数码管。由8个二级管组成。其中，分为共阴极数码管，和共阳极数码管，也就是共同用一个正极或者负极。共阴阳极对我们变成影响不大，影响的使我们的插线。那么我们先来分辨一下数码管的阴阳极。

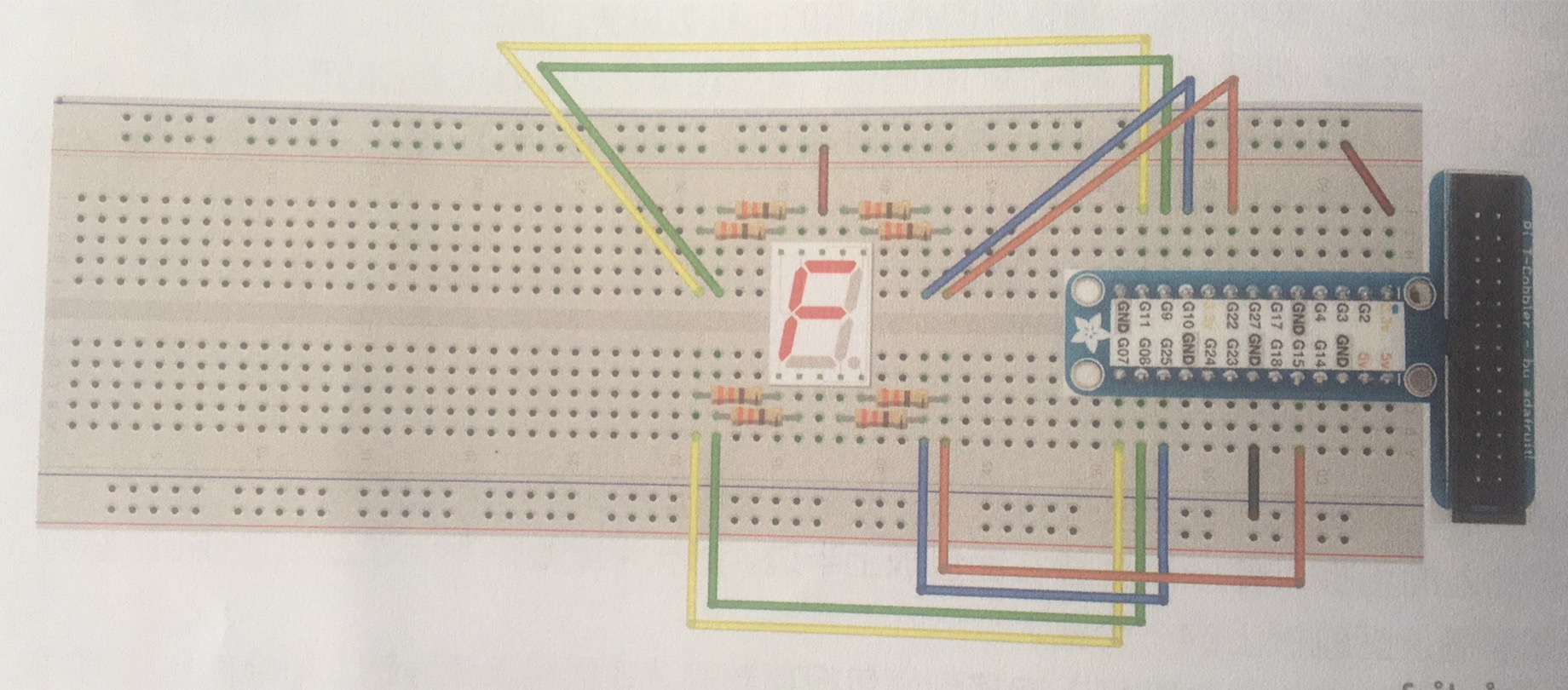
2，共极

我们先老看一下数码管的电路图，三角号表示的是电流方向



图中我们可以看出来电流是从Com线流入的。也就是共阳极的数码管。在程序中我们将给他设置成False，那么共阴极的我们将设置成True。（一会在程序中会有体现的）。一个10根线，2根共极，其他事8个二极管的线。图中是2根正极线，8个二极管的负极线。编号为ABDCEFG（Dp）Dp为小数点的二极管。多讲一点，我们能看出来每个二极管都是并联的，也就是说8个二极管中有一个坏了，其他也能正常工作。

3，电路



如图连线，（这个是共阳极的数码管）每个二极管对应一个针脚，中间的连接了3v针脚。注意。注意，注意：每个二极管在连接的时候。必须串联一个330欧的电阻。否则二极管将被电流击穿（烧毁），不要用10k电阻，太大了。二极管智能微微亮，用330就行。

4，17\_1Display

我们直接讲一下新知识，其他的，各位老师带孩子们复习的方式慢慢讲吧。况且每句代码都带有注释。希望各位老师认真一些。不会出问题的。

在程序中我们将8个我们用到的针脚都存入了一个list中LED\_PINS,注意，这个顺序是非常重要的，要对应着ABCDEFG（Dp）的顺序填写，因为我在做这个课的时候用的是共阴极数码管，所以ON变量存的是True。然后我们用变量，遍历LED\_PINS设置这些针脚为输出。然后我们有创建了一个list，把我们要亮起来的二极管存入True，不需要亮的存入False，pattern = [True,True,True,True,True,True,True,False]这个顺序说明，小数点不亮，其他的都亮。我们接着往下看。进入循环，循环中，判断的是pattern，然后打开（设置成高电平）相应的针脚号的二极管。那么从这里我们能看出来patter中的True，False并不是用来控制二极管的，而是方便我们来做判断的。一个循环8次，每次提取一个LED\_PINS中的针脚号，设置成高电平。最后因为判断是False，所以运行else，提取最后一个针脚号，设置成低电平，这样就完成数码管的点亮了，最后是七个亮，小数点那个不亮

raw\_input("finished?")这个方法我们以前用过，只不过跟上次不一样的是，我们这次只要点击一下回车就可以直接往下运行程序了。当我们程序运行到raw\_input("finished?")这个函数的时候程序会暂停。当我们点击回车的时候，程序继续运行。运行了GPIO.cleanup()清理了所有的针脚设置。

5,总结

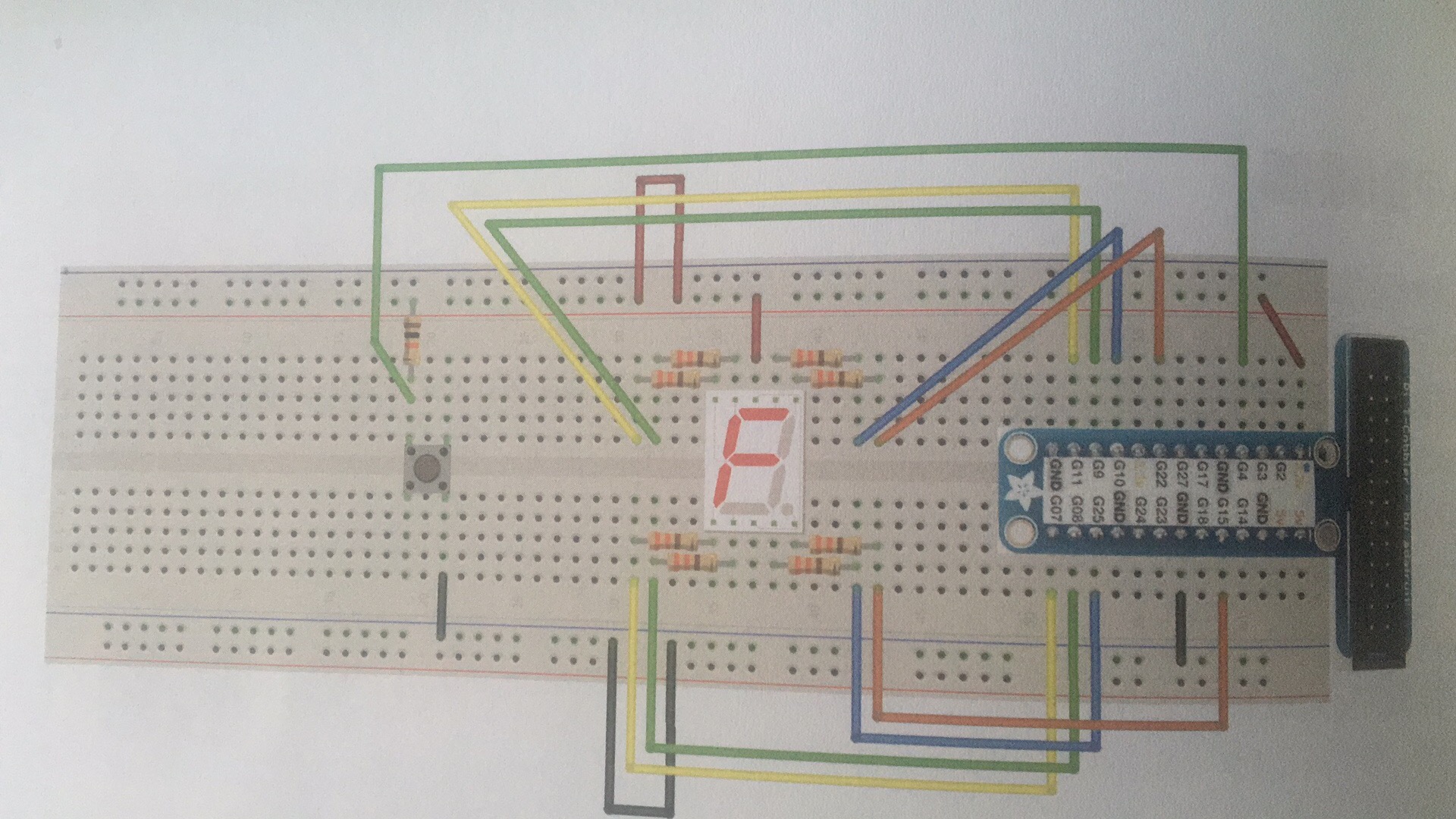
总结一下，其实这几节课讲的list比较多。这也是在教我们灵活的运用list可以实现很多功能。其次，我们要注意的是，二极管非常脆弱，所以我们在用二极管的时候，一定要加一个电阻，作为保护。最好计算一下电路中的电流，千万不要太大，否则二极管必烧。然后还需要注意的是，在我们填写数码管8个针脚到list中的时候，一定要看好针脚号，按照ABCDEFG(Dp）的顺序。

第十八课，引爆器

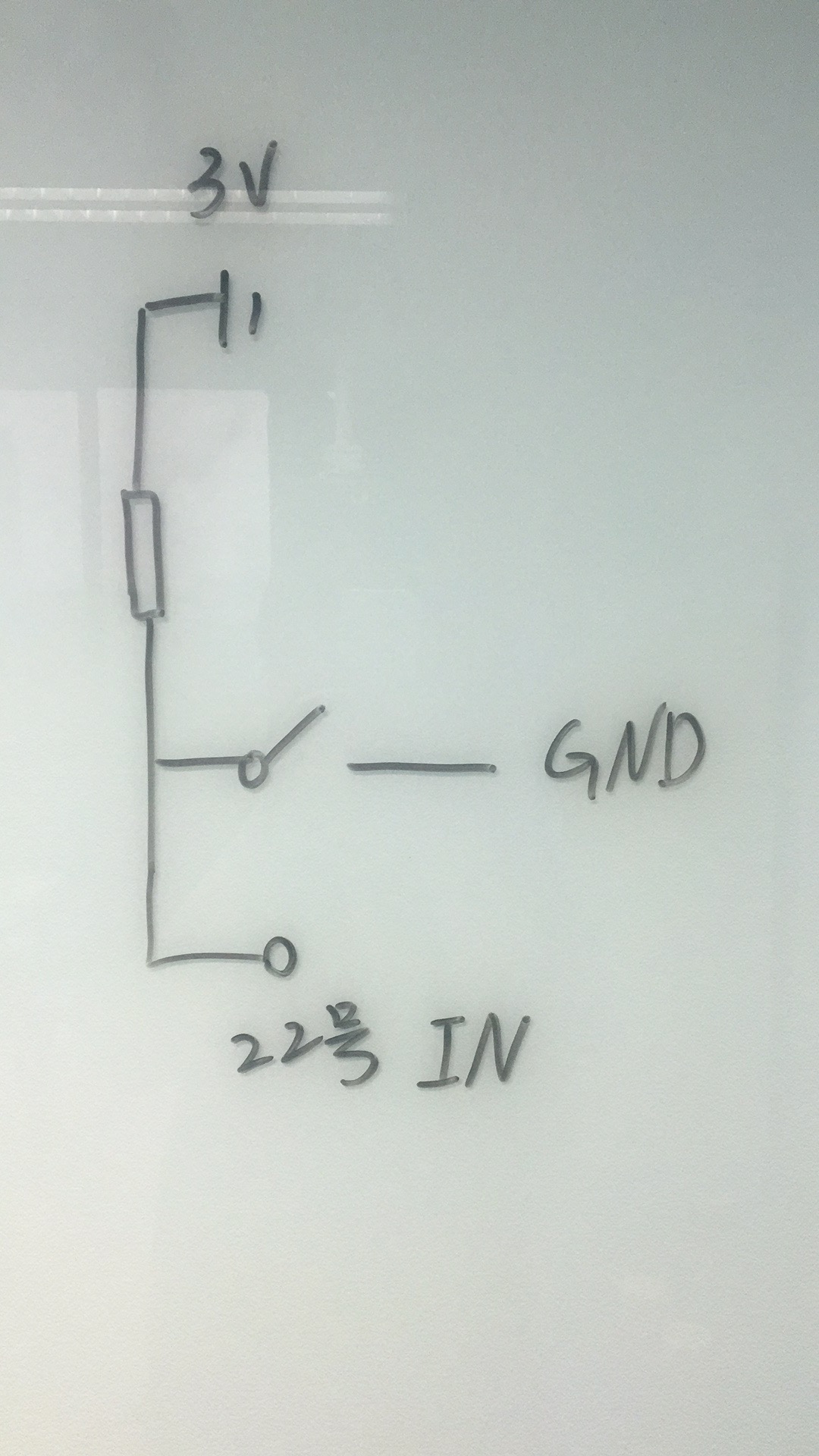
1，引爆器

提起引爆器，我们第一印象应该是遥控，按钮，爆炸等词。其实我们今天要学习的就是要用一个按钮来控制游戏中的TNT，（等你试验一下就知道，TNT并没有爆炸。使我们用AIR把方块都替换了。哈哈），接下来看一下电路。

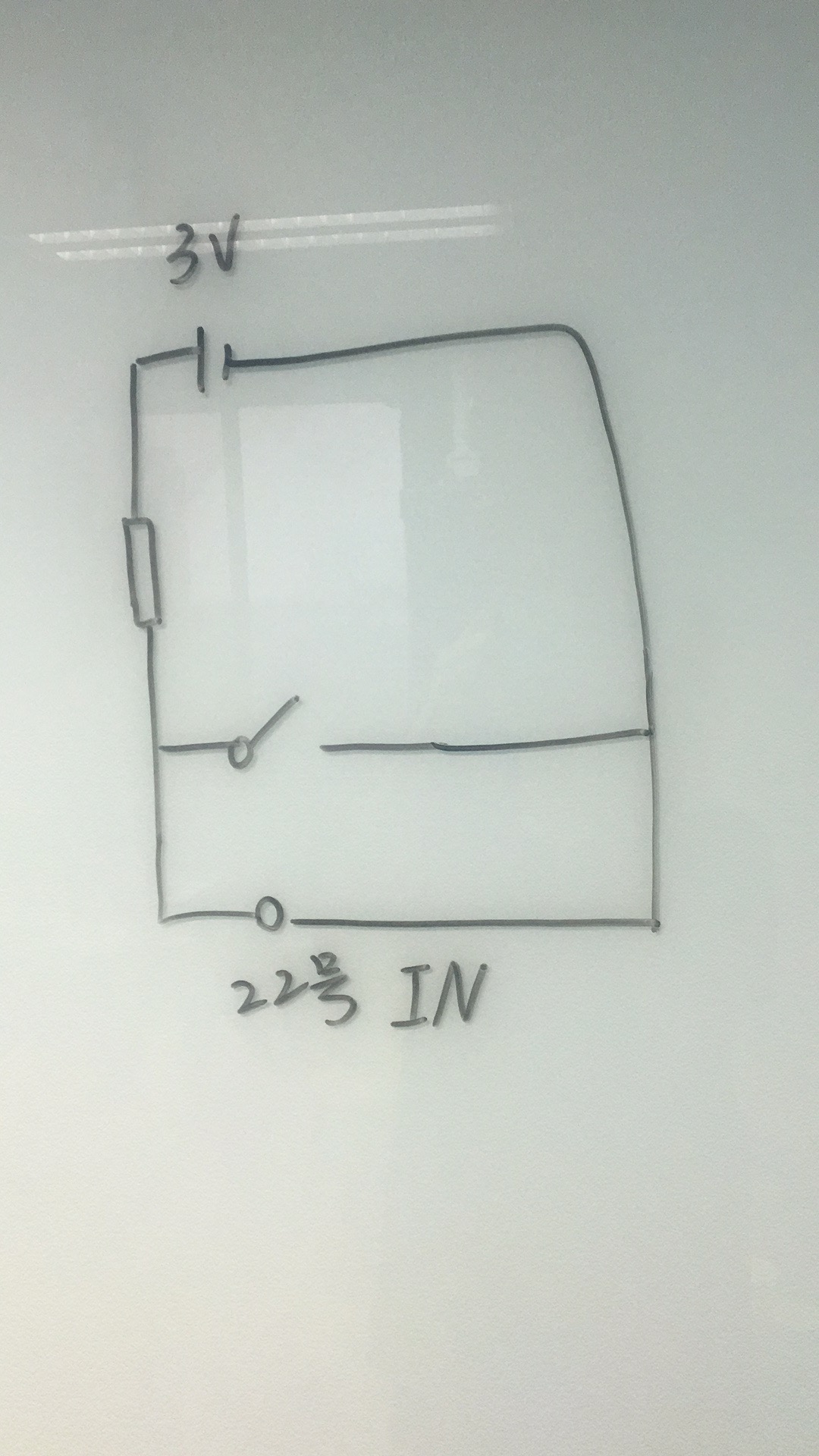
2，电路



和上节课的电路差不多。我们将按键（不分正负极）和一10k的电阻串联在一起另一端连接3V电压，另一端连接GND，那么这是一个断路还是一个回路？其实在我们没按按钮时，应该是断路，但，其实这是个回路，接下来我们看图分析一下



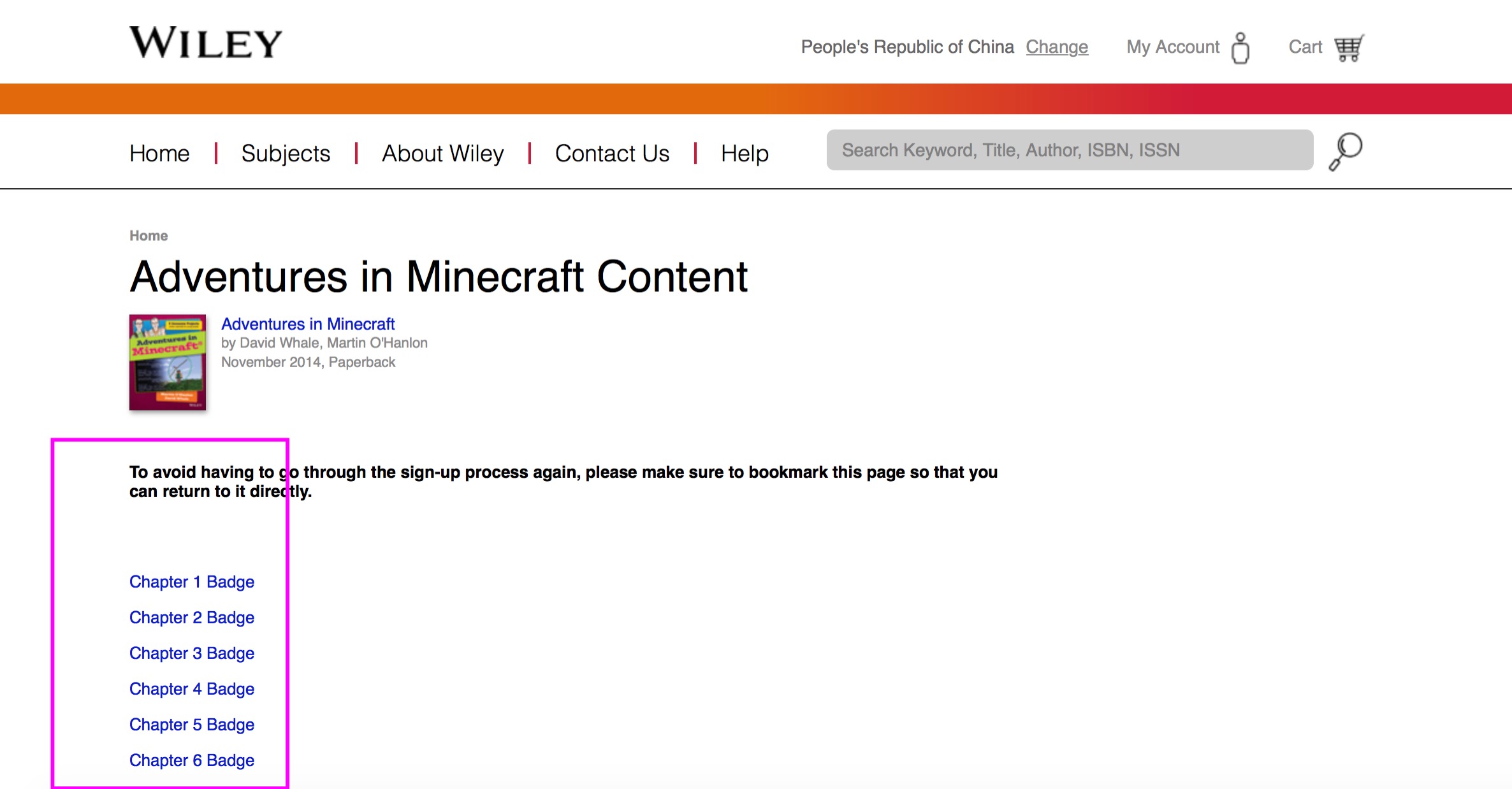
这个是我们直观的图，这个图看着可能不是很好看，那我们看下一个。



这个图就很清晰了。如果我们不按按钮，那么22号针脚将会一直有电压，如果我们按开关，那么22号针脚就被中间这个导线给短路了。也就是说，电路中只存在一个电阻元件，22针脚就没有电压了。那么我们判断的也是这个有无电压，也就是说，当我们按按钮时，22号针脚没电压。

3,seg7

这是一个包文件。我们可以用它将数码管很规律的做出一些团，比如1，2，3，4，5，6，7，E,F等等的图案，那么在我们树莓派上是没有这个文件的。我们需要自己下载一个。首先打开树莓派带的谷歌浏览器，负值这个网站：http://as.wiley.com/WileyCDA/Section/id-823690.html



进入这个网站之后，往下翻，点击这个选项[Starter Kit Pi](http://media.wiley.com/assets/7266/46/AIMStarterKitPi.tar.gz)



点击之后就会进行下载了。然后解压就行了，文件中有两个文件一个mcpi，一个anyio，因为我们在树莓派上编程，所以我们只需要anyio就可以了（mcpi可以删掉，因为我们的树莓派自带这个）我们要用的seg7就在这个anyio里了。

4，18\_1detonator

首先我们来看一下这些导入的包文件，接下来我们看一下bomb（x，y，z）函数，set了一个TNT，然受循环6次，这个是为了让数码管从5开始，这是倒计时，54321，循环完事，发送聊天“bang”

并且设置一个区域为空气造成爆炸结果。

现在看一下try：循环，每0.1秒进行一次判断，当4号针脚没有电压的时候，说明按钮被按了。调用bomb，

5,总结

这节课最难得就是这个按钮连接的电路了。因为我们这个开关并没有输入信号的功能，所以我们用了按钮短路针脚的方式，来获取按按钮的动作，从而来实现我们的功能。还有就是seg7这个文件。我们可以打开它，看看是怎么实现让数码管显示12345的。