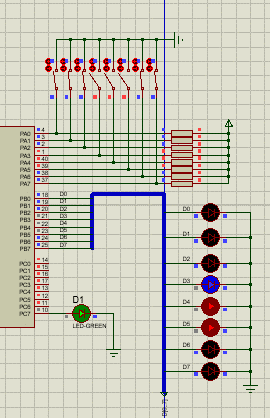
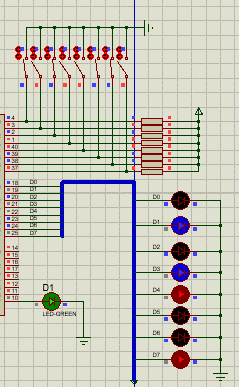
# 设计过程

## 基础实验1

该实验采用查询方式实现对指示灯的控制，因此只需使用基本的输入输出方式（方式0）即可。根据电路图可知，8255的A口输入，B口输出，C口没有用到则置为输入。该电路可以即时反应开关的开合情况，PAi口对应的开关打开，则PBi口对应的灯亮，开关状态会立即用一组LED灯显示出来。

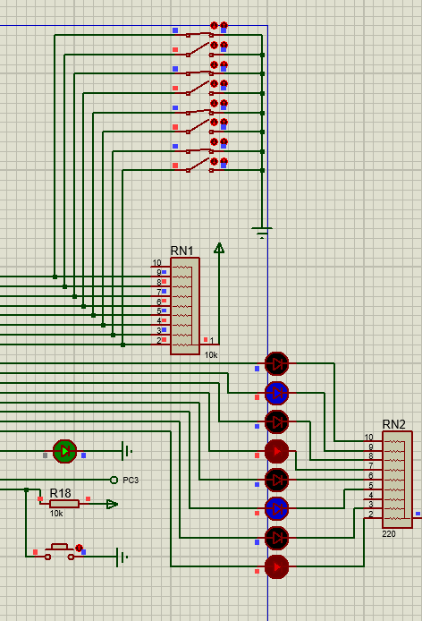
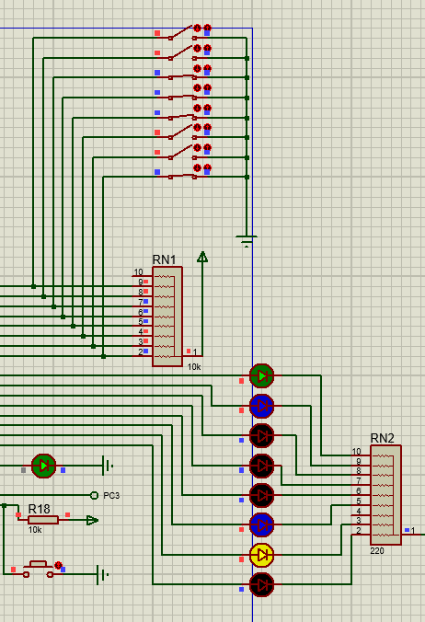
如下图所示，A口的3，4，5端口连接的开关打开，则B口对应的3，4，5端口灯亮，任意改变开关状态，LED灯会马上反应最终的开关状态：

## 基础实验2

该实验要求用方式1以中断方式采集开关状态并用一组LED灯显示出来。首先根据电路图对8255进行初始化，写控制字使A口用方式1输入，B口用方式0输出，C口置为输入。由于采用中断方式，因此可以外接一个按钮，使得每按下一次按钮，8255读取一次开关的状态，并将结果用LED灯显示出来。此外，还需要写端口C控制字，选择PC4置1，并将按钮外接在PC4。

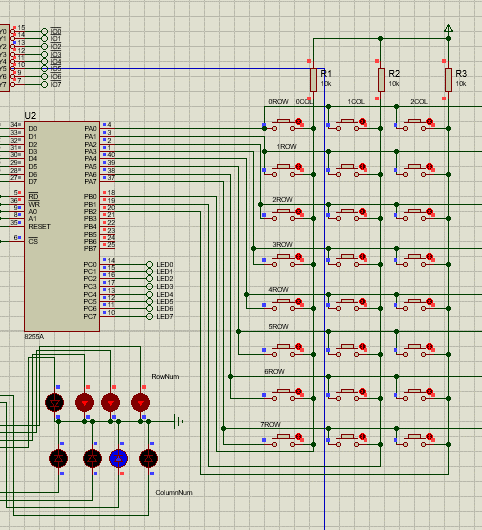
给出一个开关的初始状态，按下按钮后发出响应信号，选通PC4，则LED显示出当前开关状态，任意改变开关状态，再次按下按钮后，LED状态随之更新，一次实验的结果如下图所示：

## 拓展实验1

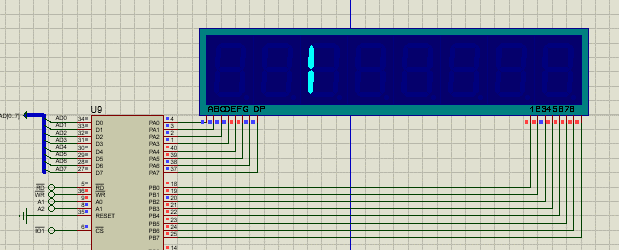
该实验要求用8255实现一个8×3键盘的按键识别，为此采用一组LED灯用于显示按下按键所在的行号和列号。由于行列数较少，因此只需8个LED灯即可完成所有状态的显示。使用一片74LS373用于输出，用高4位表示行号，低4位表示列号，行号的范围从上往下分别是0-7，列号的范围从左往右分别是0-2。按下某一按键，其所在的行列数便以LED的形式显示出来。

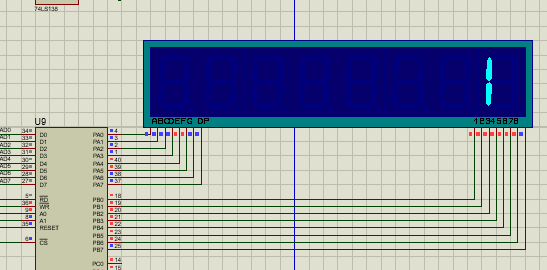
例如，按下最后一行第二列按钮，则LED灯显示0111 0010，即7行2列，结果如下图所示：



## 拓展实验2

实验要求在八个数码管上显示12345678，但最终未能成功实现，仅仅做到了在8个数码管上循环显示某一个数，若想要改变显示的数字，则需要在程序中更改编码表中想要显示的数字的值，实验使用共阴极的编码表，尝试传送数字1的结果如下：





# 实验总结

## 心得体会

本次8255的一系列实验让我对并行接口的实现计数和8255芯片的工作原理更加熟悉。通过编写不同的程序，我对于方式0方式1的适用场合更加明确，通过写控制字的情况我也对A、B口何时输入何时输出有了更清晰的认识。实验过程中，我对于Proteus软件本身也更为熟练，在老师的指导下，发现需要连接多个电阻时可以选择一个排阻，排阻相当于内置了多个电阻，并且便于修改阻值，在需要用到大量且阻值相同的电阻时，使用排阻无疑大大提高了电路的效率。

在对8255初始化的过程中，需要注意，由于Proteus软件自身的原因，初始化时需要先写C口置1/0的控制字，再写输入输出控制字，否则无法成功完成初始化。

## 实验问题汇总

1. **必做实验2中尽管开关打开，对应LED灯并没有亮起**：选择的阻值过大，Proteus选择普通电阻时默认电阻为10K，需要适当调小阻值（如200Ω），但也不可直接将电阻去除，否则本应显示高电平的一段将会显示高阻态。
2. **选做实验1按键和LED灯不对应**：起初连线时采用两组LED灯（8个一组）作为输出显示端，编写程序时用BX寄存器的高、低8位分别存储对应的行、列号，但在编写输出程序时对BH进行逻辑左移的过程中覆写了其中原有的值，输出结果也就与实际不符。出于简化程序的目的，由于实验按键仅为一8\*3矩阵，只需8位数码管即可显示全部状态，因此在程序中只需使用程序中的低8位，写入行号后若需要写入列好，只需事先将低4位左移4位再和列号相加即可。
3. **选做实验2没有实现指定功能**：尽管已经在程序中事先定义了编码表（包括1-8对应的共阴极编码表数值），编程过程中却始终只能读取编码表中的第一个，编写循环程序使得每显示完一个数字便更改地址指向下一元素，也没有达到预期效果。本想在执行换码指令XLAT时使用段超越，但编译时却无法通过，最终没能成功更改正确。